Australian
National
University

Non-colliding Gaussian Process Regressions

Wayne Wang

Supervised by

Dr. Dale Roberts

June 2020

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Finance, Economics and Statistics with Honours in Statistics at the Australian National University.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University, and, to the best of my knowledge and belief, contains no material published or written by another person, except where due reference is made in the thesis.

Wayne Wang
June 15, 2020

Acknowledgements

First of all, I would like to thank my supervisor Dr Dale Roberts for his patience, encouragement and support throughout the honours year. Whenever I had difficulties understanding some proofs or concepts, he always patiently gave me a detailed explanation which helped me a lot during my research. He also pointed me to many useful resources that helped me learn about topics in Gaussian processes. Thank you for introducing me to and sparking my interest in Gaussian process regression. This thesis would not be possible without you. I would also like to thank my closest friends Victor, Tatsunori and Lizzie for their care and support over the year. Without their friendship, the honours year would have been very difficult. Finally, I would like to thank my parents who have worked really hard to provide me with the opportunity to study overseas. I would not have had such a wonderful experience studying at ANU if not for them.

Abstract

Imposing constraints on models has been a way to incorporate prior information to develop more realistic models and/or compensate for the lack of data. In this thesis, we propose a way to impose a “non-colliding constraint” on Gaussian process regression when modelling multiple (unknown) functions at the same time. The non-colliding constraint prevents the situation where the predictions from different regressions intersect with each other. This is a desirable property when the physical process that we are trying to model exhibits a multi-layered structure such as in stratigraphy or when the underlying functions should not intersect, for example the highest, and lowest temperature of a given time period. We show that the non-colliding problem can be reformulated to modelling a sequence of Gaussian process regressions with inequality constraints. We then use a piecewise linear approximation approach proposed by López-Lopera et al. (2018) to achieve this. Through an extensive simulation study, we show that our method is able to produce more realistic models that reflect the prior information of no collisions, as well as smaller errors with less variability than the standard Gaussian process regression especially when the training set is small.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Many statistical and machine learning algorithms follow a purely data-driven methodology, learning exclusively from the evidence presented in the training data. The more data there is, the better the performance. Unfortunately, there are cases where training data is limited or expensive to obtain. In these situations, sometimes incorporating some ancillary knowledge may be able to compensate for the shortage of data and one such way is to impose certain constraints in the modelling process.

Over the last years, a number of research papers have appeared that impose constraints into various models with boundedness, monotonicity, and convexity constraints being the most common forms. Many models with constraints have been considered such as gradient boosting trees (Israeli et al., 2019), random forest (González et al., 2015), Bayesian linear regression (Migliorati et al., 2018) and local polynomial regression (Aıt-Sahalia and Duarte, 2003). Constrained models have been applied to a variety of problems across diverse fields, for example natural language processing (Chang et al., 2008), image processing (Acton and Bovik, 1998), option pricing (Aıt-Sahalia and Duarte, 2003) and estimating dose-response curves (Lee, 1996). Constrained models allow users to incorporate prior information into the model which help guide the model in the learning process, usually resulting in more realistic and better performing models than non-constrained models when data is scarce.

In this thesis, we consider Gaussian processes and Gaussian process regression. Gaussian processes can be viewed as an infinite-dimensional generalisation of multivariate Gaussian distribution in that just as a multivariate Gaussian distribution describes random variables that are vectors, Gaussian process describes random variables that are functions. Gaussian process regression is a non-parametric Bayesian method which conditions Gaussian process on data allowing us to simulate and predict where we don't have data. It is highly flexible and is able to describe complex structures through covariance functions as well as having the ability to control various properties of the resulting regression function such as smoothness, differentiability, and stationarity. The novelty in this thesis is that we consider a model with $k \geq 2$ Gaussian processes and look at imposing a "non-colliding constraint" between them.

Our motivation for such a constraint comes from the fact that when modelling multiple (unknown) functions at the same time we may want the resulting predictions to not intersect with each other. This could be due to the physical process that we are trying to model exhibiting a multi-layered structure such as in stratigraphy or having prior knowledge that the underlying functions should not intersect, for example predicting the highest, and lowest temperature of a given time period. The non-colliding constraint serves this purpose as it restricts the predicted process paths from colliding with each other.

1.2 Outline and Contributions of the Thesis

The main contribution of this thesis is developing a methodology to impose the non-colliding constraint on Gaussian process regression and to also give an asymptotic result for non-colliding Ornstein-Uhlenbeck processes.

This thesis is structured as follows. The rest of Chapter 1 gives a literature review on the some of the related topics. In Chapter 2, we introduce Gaussian process and Gaussian process regression. Some key aspects in modelling with Gaussian process regression will be discussed, providing a tutorial on how to build a variety of models with various structures. Then in

Chapter 3, we present some existing theory on non-colliding Gaussian processes given by Karlin and McGregor (1959) and Grabiner (1999), also proving a new asymptotic result for non-colliding Ornstein-Uhlenbeck processes. In Chapter 4, we propose a way to impose the non-colliding constraint on Gaussian process regression. Algorithms will be given to show how to implement the non-colliding model. Some related issues are discussed as well such as parameters estimation, local non-colliding constraint and noisy observations. We will show in Chapter 4 that simulating the truncated multivariate Gaussian distribution is required in the implementation of non-colliding Gaussian process regressions so in Chapter 5 we look at some ways to simulate the distribution and compare their efficiency. In Chapter 6, we conduct a simulation study on the non-colliding model we proposed. A number of different examples are provided and the performance of the non-colliding model is compared with the non-constrained Gaussian process regression model. Additionally, some limitations of the non-colliding model as well as when it is the appropriate choice over the non-constrained model will be discussed.

1.3 Literature Review

To the best of our knowledge, the incorporation of a "non-colliding constraint" has never appeared in the machine learning or statistics literature. We believe that this is mainly due to the fact that jointly modelling k vector-valued outputs (often called the multi-task problem in machine learning) has not received much attention. However, once embarking on this project we found a number of interesting papers in the theoretical literature on non-colliding Gaussian processes and also some recent work on Gaussian processes with inequality constraints which are both close to our topic. Therefore it seems appropriate to briefly review these topics.

1.3.1 Non-colliding Gaussian Processes

Although the topic is slightly niche, non-colliding Gaussian processes have been explored in the literature due to their applications in random matrix theory and combinatorics. The first key result in the area is a paper put forward by Karlin and McGregor giving some first results about the dynamics of non-colliding stochastic processes. In Karlin and McGregor (1959) they derive a formula which gives the transition probability of independent strong Markov processes that have not collided during a given time. However, a more widespread interest in non-colliding Gaussian processes was not developed until Dyson (1962) proposed a model to describe the eigenvalues of a Hermitian matrix whose elements execute independent Brownian motions. The model describes a system of independent Brownian motions conditioned to never collide with each other. Due to the connection with random matrices, Dyson (1962) sparked an interest in studying non-colliding Gaussian processes in the random matrix theory community. Further, since the Karlin and McGregor formula can find the non-colliding transition probability for any group of (identically distributed) strong Markov processes, it has been used to study the non-colliding version of many other stochastic processes such as random walks (König et al., 2002), the corner-growth model (Johansson, 2002), Poisson processes (O'Connell, 2002), Yule processes (Doumerc, 2005), and squared Bessel processes (König et al., 2001). The idea has also been carried over to other state spaces such as non-colliding Brownian motions on a circle (Hobson and Werner, 1996).

The first asymptotic result about non-colliding Gaussian processes appeared in Grabiner (1999) where an asymptotic result for non-colliding Brownian motions using the formula from Karlin and McGregor (1959) was derived. We will prove a similar asymptotic result in Chapter 3 for Ornstein-Uhlenbeck processes.

1.3.2 Gaussian Process Regressions with Inequality Constraints

In Chapter 4, we propose a methodology for tackling non-colliding Gaussian process regression. Although this is a topic that has never been considered before we show that it can be approached by reformulating the problem in terms of a sequence of Gaussian process regressions with inequality constraints. This is not unexpected as it is common in mathematics that some problems can be mapped onto others and one can draw an analogy between how many optimisation problems (e.g., machine learning models) can be mapped onto solving a convex optimisation problem with inequality constraints.

In the same manner, there are some quite natural ways that Gaussian processes could be constrained, for example, by ensuring that paths are bounded. Another example could be that paths are monotonic. Interestingly both these problems can be mapped to a Gaussian process with inequality constraints.

As a Gaussian process is in infinite-dimensional object (i.e., function-valued object) it needs to be approximated by something finite-dimensional for it be implemented on a computer. In the literature about Gaussian process regression with inequality constraints there are two types of numerical methods that have been considered.

The first approach discretises the input space into a set of virtual observation locations and simulates a conditional Gaussian process which satisfies the inequality constraints at these locations (see for example Abrahamsen and Benth 2001, Da Veiga and Marrel 2012, Golchi et al. 2015, Riihimäki and Vehtari 2010, Wang and Berger 2016, Agrell 2019). Only a finite number of input locations satisfies the inequality constraints under this first approach. One attractive property of Gaussian process that is important in modelling monotonicity and convexity constraints under this approach is the fact that the partial derivative processes of a Gaussian process are also Gaussian processes, since differentiation is a linear operator (Cramér and Leadbetter, 2013). These Gaussian processes have covariance functions that can be derived from the original Gaussian process (see for example Rasmussen and Williams 2006). This allows for the inclusion of derivative observations in a Gaussian process model and the ability to predict derivatives. For example, Riihimäki and Vehtari (2010) introduced monotonicity information to a Gaussian process model by including virtual derivatives at some pre-specified locations. By doing so, the derivative process is required to be positive at these locations. In later work, ways to find an optimal set of virtual observation locations such that the constraints can be satisfied at any input location with sufficiently high probability has been explored (Golchi et al. 2015, Wang and Berger 2016, Agrell 2019).

The second approach uses a finite-dimensional approximation of Gaussian process which allows the constraints to hold in the entire domain. This approach was first introduced by Maatouk and Bay (2017), and later followed up by López-Lopera et al. (2018). Maatouk and Bay's piecewise linear approximation of Gaussian process allowed them to incorporate the advantage of inequality constrained splines, that is the ability to have inequality constraints satisfied in the entire domain. A lot of research has been done on splines with inequality constraints as well as applications (see for example Fritsch and Carlson 1980, Wright et al. 1980, Villalobos and Wahba 1987, Micchelli and Utreras 1988, Ramsay 1998, Dole 1999, Wolberg and Alfy 2002).

Since we want the non-colliding constraint to hold in the entire domain, we will make use of the approach described in López-Lopera et al. (2018) to develop a methodology for non-colliding Gaussian process regressions in Chapter 4.

Chapter 2

Gaussian Processes and GP Regression

In this chapter, we give a brief introduction to Gaussian processes and Gaussian process regression. In Section 2.1, we define Gaussian processes and give two well-known examples: the Brownian motion and the Ornstein-Uhlenbeck process. In Section 2.2, we present Gaussian process regression, an application of Gaussian processes to regression problems. Through covariance functions, which will be discussed in Section 2.2.2, Gaussian process regression offers great flexibility and is able to describe a variety of structures, as well as being able to control various properties of the resulting regression function such as smoothness, differentiability and stationarity.

2.1 Gaussian Processes

Gaussian processes arise from an infinite-dimensional generalisation of multivariate Gaussian distribution. A multivariate Gaussian distribution describes random variables that are vectors, whereas Gaussian process describes random variables that are functions. We now give a formal definition for Gaussian processes.

Definition 2.1. A stochastic process, $\{X(t); t \in T\}$, is a *Gaussian process* if for every finite subset of indices $t_1, t_2, \ldots, t_k$ in the index set T, $(X(t_1), X(t_2), \ldots, X(t_k))$ is a multivariate Gaussian random variable.

Just as a multivariate Gaussian distributions is characterised by a mean vector and a covariance matrix, a Gaussian process is completely determined by a mean function, $m(t) = \mathbb{E}[X(t)]$, and a covariance function, $k(s,t) = \text{Cov}(X(s), X(t))$. Two well-known stochastic processes: Brownian motion and Ornstein-Uhlenbeck process, are in fact Gaussian processes. Both processes have been studied and applied extensively in fields such as physical sciences, mathematics, and finance. We now show that these two processes are indeed Gaussian processes that can be defined by a mean and covariance function.

Example 2.1. A Brownian motion (see Definition A.1) is an example of a Gaussian process with $m(t) = 0$ and $k(s,t) = \min\{s,t\}$ (see Proposition A.1 for further details).

Example 2.2. Consider the Ornstein-Uhlenbeck process $\{X(t); t \geq 0\}$ given by

$$X(t) = X(0)e^{-\theta t} + \int_0^t e^{-\theta(t-s)} \sigma dW(s).$$

It is easy to argue that this process is Gaussian as it is given by the addition of a deterministic part and a Wiener integral (which is Gaussian). Further, we have that

$$\mathbb{E}[X(t)] = X(0)e^{-\theta t},$$

and for $0 \leq s \leq t$, we have

$$\begin{aligned}
\text{Cov}[X(s), X(t)] =& \mathbb{E}[(X(s) - \mathbb{E}[X(s)])(X(t) - \mathbb{E}[X(t)])] \\
=& \mathbb{E}\left[\int_0^s e^{-\theta(s-u)} \sigma dW(u) \int_0^t e^{-\theta(t-v)} \sigma dW(v)\right] \\
=& \mathbb{E}\left[\int_0^s e^{-\theta(s-u)} \sigma dW(u) \left(\int_0^s e^{-\theta(t-v)} \sigma dW(v) + \int_s^t e^{-\theta(t-v)} \sigma dW(v)\right)\right]
\end{aligned}$$

By the independent increments property of Brownian motion,

$$\begin{aligned}
=& \mathbb{E}\left[\int_0^s e^{-\theta(s-u)} \sigma dW(u) \int_0^s e^{-\theta(t-v)} \sigma dW(v)\right] \\
&+ \mathbb{E}\left[\int_0^s e^{-\theta(t-u)} \sigma dW(u)\right] \mathbb{E}\left[\int_s^t e^{-\theta(t-v)} \sigma dW(v)\right] \\
=& \sigma^2 e^{-\theta(s+t)} \mathbb{E}\left[\int_0^s e^{\theta u} dW(u) \int_0^s e^{\theta v} dW(v)\right]
\end{aligned}$$

By Itô isometry,

$$\begin{aligned}
=&\sigma^2 e^{-\theta(s+t)}\left[\int_0^s e^{2\theta a}da\right]\\
=&\sigma^2 e^{-\theta(s+t)}\mathbb{E}\left(\frac{1}{2\theta}e^{2\theta s}-\frac{1}{2\theta}\right)\\
=&\frac{\sigma^2}{2\theta}(e^{-\theta(t-s)}-e^{-\theta(s+t)})
\end{aligned}$$

If $0 \leq t \leq s$, $\mathrm{Cov}[X(s),X(t)] = \frac{\sigma^2}{2\theta}(e^{-\theta(s-t)} - e^{-\theta(s+t)})$. Therefore, the covariance function of the Ornstein-Uhlenbeck process is

$$k(s,t) = \frac{\sigma^2}{2\theta}(e^{-\theta|t-s|} - e^{-\theta(s+t)}).$$

2.2 Gaussian Process Regression

Gaussian process regression is a non-parametric Bayesian method. By conditioning a Gaussian process prior on data, the posterior distribution can be found which is then used to make predictions. Earlier, when we introduced Gaussian processes, T was used as the index set. In Gaussian process regression, we use the independent variables as the indices and the expected value of the Gaussian process at each observation as the regression output. We will see shortly that not only do we receive an output from our Gaussian process regression model, we also obtain the probability distribution of the output, thus allowing us to calculate the confidence interval for our predictions as well.

A regression problem can be formulated as:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$$

where $\mathbf{x}_i$ is the ith observation of the independent variables and y_i is the corresponding ith observation of the dependent variable. ϵ_i is the Gaussian noise term which is independent and identically distributed for all i and follows a normal distribution with mean 0 and constant variance σ_n^2.

The goal of the regression problem is to make inference about f so that given any $\mathbf{x}$ we can predict the corresponding y value. Let $X =: [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_1}]^T$ denotes the $n_1 \times p$ matrix containing the observed independent variables (the "training set"), where n_1 is the number of observations and p is the number of independent variables, i.e. $\mathbf{x}_i \in \mathbb{R}^p$. Now, let $\mathbf{y}$ denote the vector of the dependent variable used for training and $\widetilde{X}$ be the the $n_2 \times p$ test set. To simplify notation, we write $\widetilde{\mathbf{f}} := f(\widetilde{X})$.

Then, making use of the nice property of a Gaussian process (Definition 2.1), the prior distribution is given by

$$\widetilde{\mathbf{f}} | \widetilde{X} \sim \mathcal{N}(m(\widetilde{X}), K(\widetilde{X}, \widetilde{X}))$$

This is due to $\widetilde{X} := [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_{n_2}]^T$ being countable and finite. $\widetilde{X}$ is used as the index set for the Gaussian process f. Then by the definition of Gaussian process, $\widetilde{\mathbf{f}} = [f(\widetilde{\mathbf{x}}_1), f(\widetilde{\mathbf{x}}_2), \ldots, f(\widetilde{\mathbf{x}}_{n_2})]^T$ has a multivariate normal distribution with mean defined by the mean function, $m(\widetilde{X})$, and covariance matrix defined by evaluating the covariance between all pairs of observations using the covariance function $k(\cdot, \cdot)$, giving

$$K(\widetilde{X}, \widetilde{X}) := \begin{bmatrix} k(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_1) & k(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2) & \ldots & k(\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_{n_2}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\widetilde{\mathbf{x}}_{n_2}, \widetilde{\mathbf{x}}_1) & k(\widetilde{\mathbf{x}}_{n_2}, \widetilde{\mathbf{x}}_2) & \ldots & k(\widetilde{\mathbf{x}}_{n_2}, \widetilde{\mathbf{x}}_{n_2}) \end{bmatrix}.$$

It follows that the joint distribution of $\widetilde{\mathbf{f}}$ and $\mathbf{y}$ under the prior is

$$\begin{bmatrix} \widetilde{\mathbf{f}} \\ \mathbf{y} \end{bmatrix} \Bigg| \widetilde{X}, X \sim \mathcal{N}\left(\begin{bmatrix} m(\widetilde{X}) \\ m(X) \end{bmatrix}, \begin{bmatrix} K(\widetilde{X}, \widetilde{X}) & K(\widetilde{X}, X) \\ K(X, \widetilde{X}) & K(X, X) + \sigma_n^2 I \end{bmatrix} \right)$$

The posterior distribution $\widetilde{\mathbf{f}} | X, \mathbf{y}, \widetilde{X}$ can then be found using a fact about the conditional distributions of a multivariate Gaussian distribution given by Theorem 2.1 (proof is given in Appendix).

Theorem 2.1 (Conditional multivariate normal distribution). *Let* $\mathbf{y}$ *be a* N *dimensional random vector which follows a multivariate normal distribution.*

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

If we partition $\mathbf{y}$ *into* $\mathbf{y}_1$ *(*$p \times 1$*) and* $\mathbf{y}_2$ *(*$q \times 1$*), and* $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$ *accordingly (such that* $\mathbf{y}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_{11})$ *and* $\mathbf{y}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_{22})$*), then the conditional distribution of* $\mathbf{y}_1$ *given* $\mathbf{y}_2$ *is* $\mathbf{y}_1|\mathbf{y}_2 \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})$ *where*

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2),$$
$$\widetilde{\boldsymbol{\Sigma}} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Theorem 2.1 allows us to obtain the posterior distribution

$$\widetilde{\mathbf{f}}|X, \mathbf{y}, \widetilde{X} \sim \mathcal{N}(\bar{\mathbf{f}}, \mathrm{Cov}(\widetilde{\mathbf{f}}))$$

where

$$\bar{\mathbf{f}} := \mathbb{E}[\widetilde{\mathbf{f}}|X, \mathbf{y}, \widetilde{X}] = m(\widetilde{X}) + K(\widetilde{X}, X)[K(X,X) + \sigma_n^2 I]^{-1}(\mathbf{y} - m(X))$$
$$\mathrm{Cov}(\widetilde{\mathbf{f}}) = K(\widetilde{X}, \widetilde{X}) - K(\widetilde{X}, X)[K(X,X) + \sigma_n^2 I]^{-1}K(X, \widetilde{X})$$

Finally, to make a prediction using the Gaussian process model, simply compute the posterior mean at the test input locations, and if one is interested in the uncertainty of the predictions, compute the posterior covariance which can be used to calculate the confidence interval for each point of prediction. Figure 2.1 illustrates one example of Gaussian process regression.

2.2.1 Incorporating Explicit Basis Functions

In practice, we often set the mean function to be 0 due to a lack of prior information on it. However, there are times when one might wish to specify a mean function for reasons such as interpretability and inclusion of prior information. A fixed mean function can be easily applied using the equations given earlier. However, specifying a single mean function can be challenging in practice and one might find it easier to specify a number of basis functions instead.

Consider the model

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{b}(\mathbf{x})^T\boldsymbol{\beta}$$

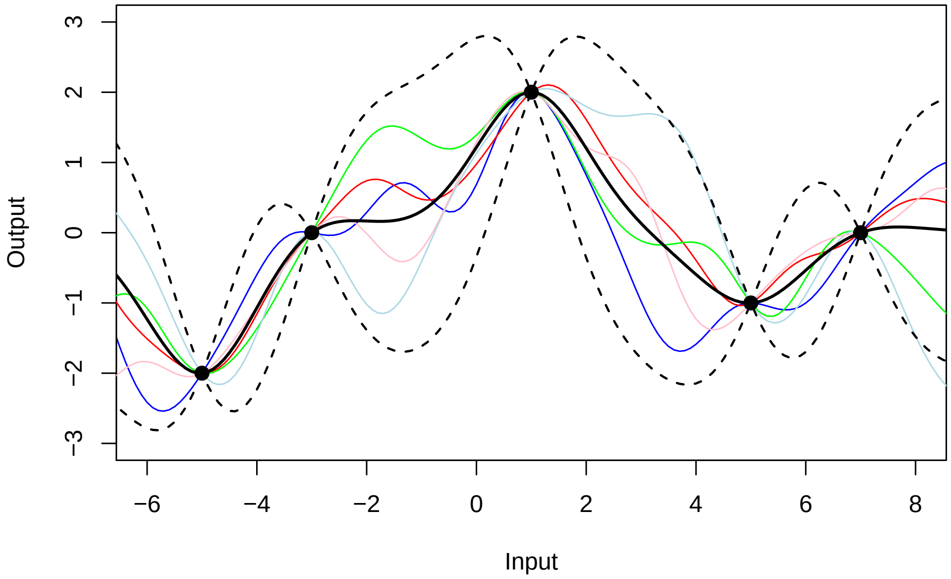


Figure 2.1: Five random functions (i.e., sample paths) drawn from the posterior distribution (coloured solid lines). The solid black line indicates the posterior mean. The black dots are observations that are assumed to be noise free i.e. $\sigma_n^2 = 0$. The dotted lines represent 95% confidence interval which equals to the posterior mean plus and minus 1.96 times the posterior standard deviation.

where $f(\mathbf{x}) \sim \mathcal{GP}(0, k)$, $\mathbf{b}(\mathbf{x})$ is a set of basis functions (e.g. $\mathbf{b}(x) = (1, x, x^2, x^3)$), and the vector $\boldsymbol{\beta}$ gives the weights of the basis functions. This model can be interpreted as that the data is close to a linear model given by $\mathbf{b}(\mathbf{x})^T\boldsymbol{\beta}$ and the residuals are modelled by a Gaussian process. The weights $\boldsymbol{\beta}$ can be estimated with the parameters of the covariance function, which will be discussed in 2.2.3.

If we choose the prior on $\boldsymbol{\beta}$ to be Gaussian, then $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and

$$g(\mathbf{x}) \sim \mathcal{GP}(\mathbf{b}(\mathbf{x})^T\boldsymbol{\mu}, k + \mathbf{b}(\mathbf{x})^T\Sigma\mathbf{b}(\mathbf{x})).$$

Again, we can apply the mean and covariance functions on the training and test data, and use Theorem 2.1 to find the posterior distribution as before, giving

$$\widetilde{\mathbf{f}}|X, \mathbf{y}, \widetilde{X} \sim \mathcal{N}(\bar{\mathbf{f}}, \mathrm{Cov}(\widetilde{\mathbf{f}}))$$

where

$$\bar{\mathbf{g}} = \widetilde{B}^T\bar{\boldsymbol{\beta}} + K(\widetilde{X}, X)[K(X,X) + \sigma_n^2 I]^{-1}(\mathbf{y} - B^T\bar{\boldsymbol{\beta}})$$

$$\mathrm{Cov}(\widetilde{\mathbf{g}}) = \mathrm{Cov}(\widetilde{\mathbf{f}}) + R^T[\Sigma^{-1} + B[K(X,X) + \sigma_n^2 I]^{-1}B^T]^{-1}R.$$

Here, B and $\widetilde{B}$ are the matrices given by evaluating $\mathbf{b(x)}$ on the training and test data respectively, $\bar{\boldsymbol{\beta}} = [\Sigma^{-1} + B[K(X,X) + \sigma_n^2 I]^{-1}B^T]^{-1}[B[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} + \Sigma^{-1}\boldsymbol{\mu}]$, and $R = \widetilde{B} - B[K(X,X) + \sigma_n^2 I]^{-1}K(X,\widetilde{X})$.

2.2.2 Covariance Functions

Choosing the right covariance function is a crucial component when modelling with Gaussian process regression. They determine how points close to a test point influence the prediction at that point. In this section, we give some examples of commonly used covariance functions and also provide ways to create new covariance functions from existing ones.

Firstly, not all functions can be used as covariance functions. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is the input space can be called a *covariance function* if it satisfies the following properties:

- Symmetric, i.e., $k(a,b) = k(b,a)$ for $a,b \in \mathcal{X}$;
- The associated matrix $K := (k(x_i,x_j))_{i,j=1}^n$ is positive semidefinite[1] for any $n \in \mathbf{N}$ and for any set of points $\{x_1, x_2, \dots, x_n\}$ in the input space.

We recall that if a covariance function $k(a,b)$ is a function of $a-b$ then it is called *stationary* and it is invariant to translations in the input space. Additionally, if it is a function of $|a-b|$, it is called *isotropic* and is invariant to all rigid motions. A special class of covariance functions are those that are given by $k(a,b) = a \cdot b$ and we call these *dot product covariance functions*. Dot product covariance functions are invariant to rotations of the coordinate system.

2.2.2.1 Examples of Covariance Functions

We now give some examples of commonly used covariance functions.

[1] A $n \times n$ symmetric real matrix K is positive semidefinite if $c^T K c \geq 0$ for all $c \in \mathbb{R}^n$.

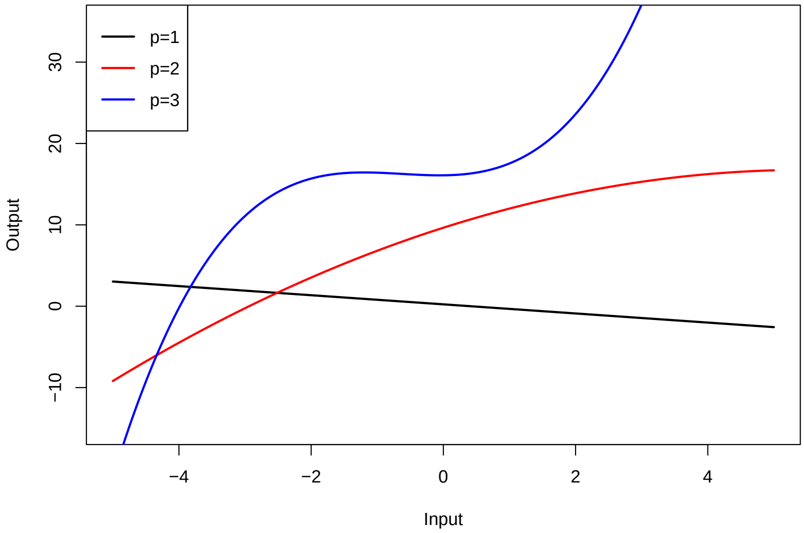


Figure 2.2: Random functions drawn from Gaussian processes with the polynomial covariance function with $\sigma = 1$ and various values of p. The functions were obtained by discretizing the x-axis into 1000 equally spaced points.

Linear and Polynomial Covariance Functions

The polynomial covariance function is given by

$$k^{Po}(a,b) = (a^T b + \sigma^2)^p,$$

where p is a positive integer and σ is a non-negative parameter. When $p = 1$, it is a linear covariance function and is obtainable from linear regression by placing $\mathcal{N}(0,1)$ priors on the coefficients of regressors and a $\mathcal{N}(0,\sigma^2)$ prior on the bias term.

Squared Exponential Covariance Function

The squared exponential covariance function has the form

$$k^{SE}(a,b) = \exp\left(-\frac{|a-b|^2}{2l^2}\right)$$

with parameter l. A Gaussian process with squared exponential covariance function is infinitely mean square differentiable, and thus very smooth.

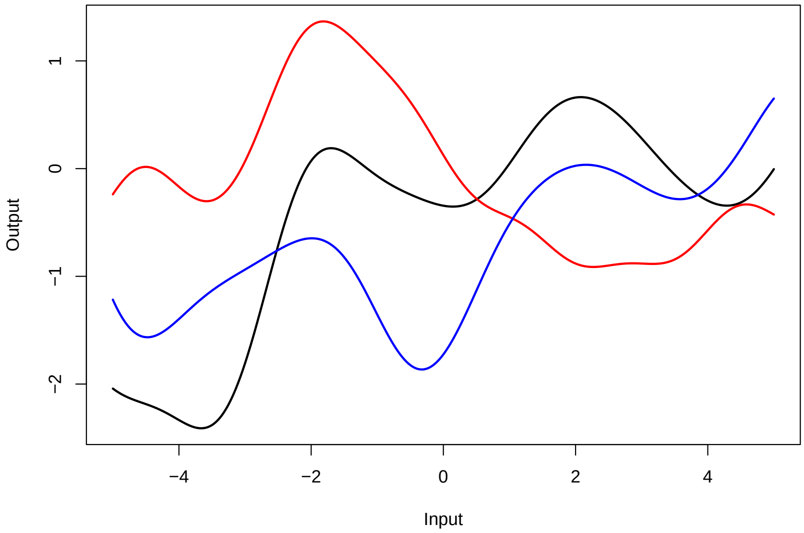


Figure 2.3: Three random functions drawn from Gaussian processes with the squared exponential covariance function with $l = 1$. The functions were obtained by discretizing the x-axis into 1000 equally spaced points.

Matérn Covariance Function

The Matérn covariance function is defined as

$$k^M(a,b) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|a-b|}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|a-b|}{l}\right)$$

where ν and l are positive parameters, $\Gamma(\cdot)$ is the Gamma function, and K_ν is a modified Bessel function (Abramowitz and Stegun, 1965). A Gaussian process $f(\mathbf{x})$ with Matérn covariance function is k times mean square differentiable if and only if $\nu > k$. If $\nu = p + \frac{1}{2}$, where p is a non-negative integer, then the Matérn covariance function can be simplified to

$$k^M_{\nu=p+\frac{1}{2}}(a,b) = \exp\left(-\frac{\sqrt{2\nu}|a-b|}{l}\right)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}|a-b|}{l}\right)^{p-i}.$$

As $\nu \to \infty$, the Matérn covariance function converges to the squared exponential covariance function. As shown in Figure 2.4, the Matérn covariance function offers great flexibility in modelling as the user is able to adjust the smoothness of the function being modelled by changing the parameter ν.

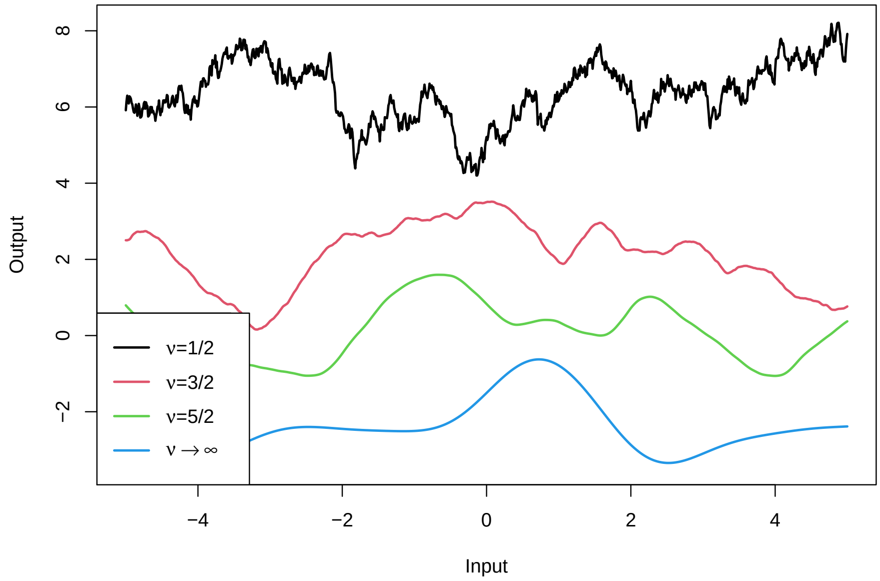


Figure 2.4: Random functions drawn from Gaussian processes with the Matérn covariance function with $l = 1$ and various values of ν. The functions were obtained by discretizing the x-axis into 1000 equally spaced points.

Exponential Covariance Function

The exponential covariance function has the following expression

$$k^E(a,b) = \exp\left(-\frac{|a-b|}{l}\right)$$

The exponential covariance function can be obtained from the Matérn covariance function by setting $\nu = \frac{1}{2}$. A Gaussian process with the exponential covariance function is continuous but not differentiable. When the dimension of the input space is 1, this is the covariance function of the Ornstein-Uhlenbeck process.

Periodic Covariance Function

The periodic covariance function (MacKay, 1998) is given by

$$k^P(a,b) = \exp\left(\frac{-2\sin^2(\frac{\pi(a-b)}{p})}{l^2}\right)$$

where p is the period of repetition. The periodic covariance function allow the user to model repeating patterns.

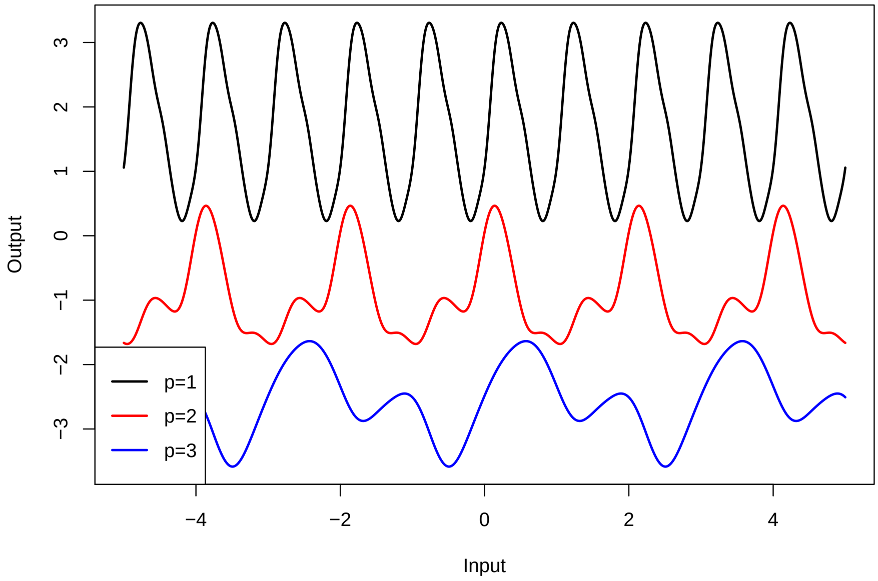


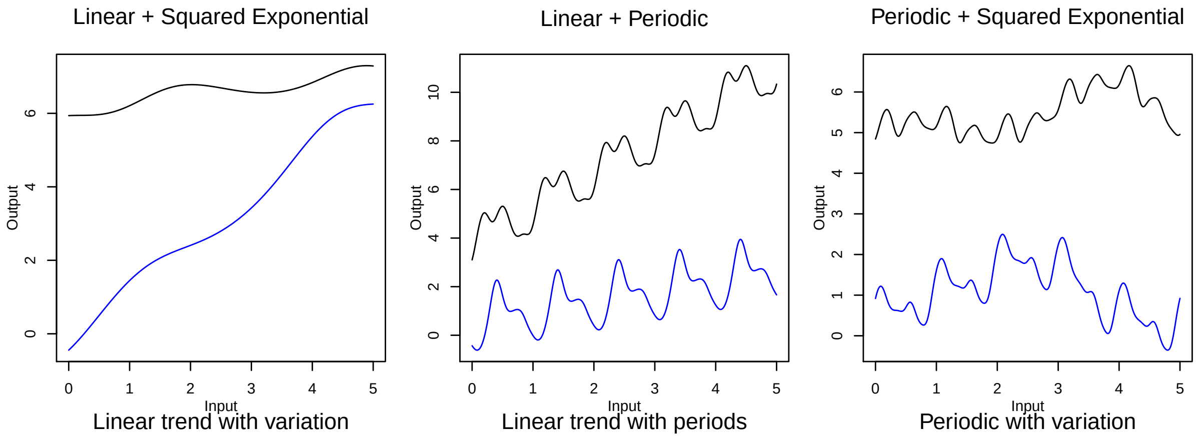
Figure 2.5: Random functions drawn from Gaussian processes with the periodic covariance function with $l = 1$ and various values of p. The functions were obtained by discretizing the x-axis into 1000 equally spaced points.

2.2.2.2 Combining Covariance Functions

Existing covariance functions can be combined to make new covariance functions that can be used to model data with more complex properties. We will focus our discussion on two ways of combining covariance functions: summation and multiplication.

Summation of Covariance Functions

The sum of two covariance functions is also a covariance function. Consider two independent Gaussian processes, $f_1 \sim \mathcal{GP}(m_1, k_1)$ and $f_2 \sim \mathcal{GP}(m_2, k_2)$. Then the sum of f_1 and f_2 is also a Gaussian process, $f_1 + f_2 \sim \mathcal{GP}(m_1 + m_2, k_1 + k_2)$. This allow us to build additive models by summing up independent functions with different properties. Additive models can help us explore individual effects by looking at component functions, and help us interpret the models. Figure 2.6 shows some examples of Gaussian processes generated by adding two covariance functions together. As shown, we can decompose complex structure into



Linear trend with variation

Linear trend with periods

Periodic with variation

Figure 2.6: Examples of random functions drawn from Gaussian processes with covariance functions that are the sums of two existing covariance functions. **Left:** The sum of a linear covariance function with $\sigma = 1$ and a squared exponential covariance function with $l = 1$. **Middle:** The sum of a linear covariance function with $\sigma = 1$ and a periodic covariance function with $p = 1$ and $l = 1$. **Right:** The sum of a periodic covariance function with $p = 1$ and $l = 1$ and a squared exponential covariance function with $l = 1$. The functions were obtained by discretizing the x-axis into 500 equally spaced points.

simpler components and model each component with a function then adding them together. For example, if we have data with structure like the one in the second plot of Figure 2.6, we can break the structure into two parts: an increasing linear trend and periodic variations; and each can be modelled with the covariance functions introduced in 2.2.2.1.

Multiplication of Covariance Functions

The product of two covariance functions is also a covariance function. Since covariance functions are positive semidefinite, the product of positive semidefinite functions is always positive semidefinite. Therefore, the resulting function is a valid covariance function. Taking the product of covariance functions would combine the properties of the covariance functions, producing a covariance function that is able to adapt to data with more complex characteristics. Figure 2.7 shows some examples of how taking the product of two covariance functions results in more sophisticated structures that are not achievable by a single covariance function or adding several covariance functions together. Figure 2.7 only shows examples of multiplying two covariance functions together, but one can take the product of

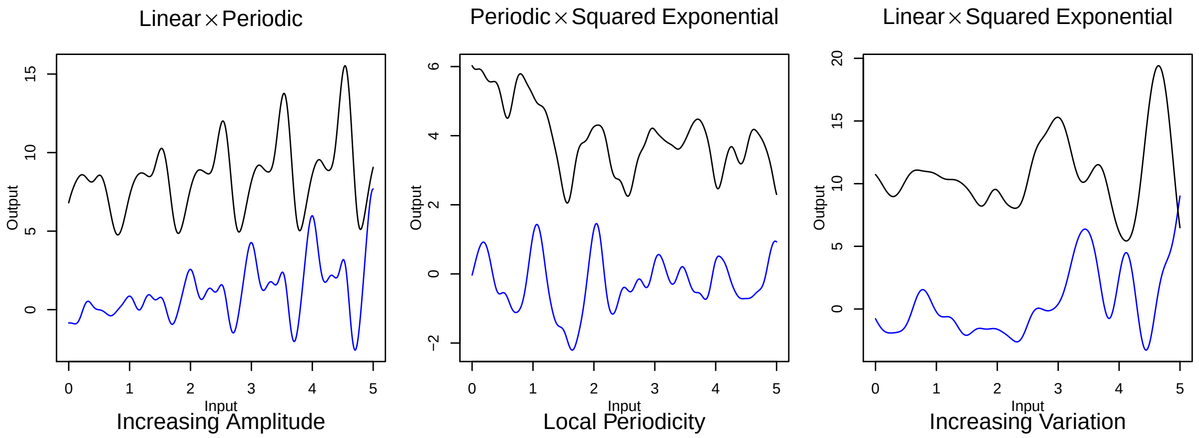


Figure 2.7: Examples of random functions drawn from Gaussian processes with covariance functions that are the products of two existing covariance functions. **Left:** The product of a linear covariance function with $\sigma = 1$ and a periodic covariance function with $p = 1$. **Middle:** The product of a periodic covariance function with $p = 1$ and $l = 1$, and a squared exponential covariance function with $l = 1$. **Right:** The product of a linear covariance function with $\sigma = 1$ and a squared exponential covariance function with $l = 0.2$. The functions were obtained by discretizing the x-axis into 500 equally spaced points.

any number of covariance functions to combine multiple properties together.

2.2.3 Model Selection

We have seen earlier that there are a number of specifications needs to be made in the implementation of a Gaussian process regression, such as specifying the covariance function(s) and the parameters involved e.g. l. While some may be easily determined, others may be difficult due to a lack of information, e.g. the variance of noise σ_n^2 and the parameters of covariance functions. We need a way to help us find the best values for the free parameters and compare models with different specifications so we can choose the one that best describes the data. We now present one way of addressing the model selection problem for Gaussian process regression.

Let $\boldsymbol{\theta}$ be the vector of all parameters, e.g., $\boldsymbol{\theta} = (\sigma_n^2, l, p)$. Then the log marginal likelihood is given by

$$\log p(\mathbf{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} - \frac{1}{2}\log|K(X,X) + \sigma_n^2 I| - \frac{n}{2}\log(2\pi)$$

The parameters can be set by maximising the log marginal likelihood, which can be done using gradient based methods such as the conjugate gradient method. Furthermore, the log marginal likelihood value can be used to compare between models. For example, two models with different selections of covariance functions can be compared after optimising the parameters, and the one with a higher likelihood would be the better model.

2.3 Limitation

In this chapter, we have shown how Gaussian process regression is a powerful method with great flexibility that is able to model a variety of statistical structures through the use of existing covariance functions or a combination of them. However, it has one major limitation which is its computational cost. As shown in Section 2.2, to compute the posterior mean or covariance matrix one must invert the $n_1 \times n_1$ matrix $K(X,X) + \sigma_n^2 I$ (where n_1 is the number of training observations). This is an operation with complexity $\mathcal{O}(n_1^3)$. For large datasets, this is too computationally expensive in both time and space. Fortunately, a wide range of methods have been proposed to tackle this problem. For example, incomplete Cholesky factorization (Fine and Scheinberg, 2001), the Nyström method (Williams and Seeger, 2001) and Bayesian committee machine (Tresp, 2000).

Chapter 3

Non-colliding Gaussian Processes

In this chapter, we present some theory on non-colliding Gaussian processes. We will first explain what we meant by “non-colliding” and introduce a different way of looking at the non-colliding problem in Section 3.1. Then, a fundamental formula used in the study of non-colliding stochastic processes will be presented in Section 3.2. Lastly, two kinds of non-colliding Gaussian processes, namely Brownian motion and Ornstein-Uhlenbeck process, will be discussed in Section 3.3 and Section 3.4 respectively.

3.1 The Weyl Chamber

We say that n one-dimensional stochastic processes are *non-colliding* up to time T when their paths do not cross each other at any point before time T (i.e. $X_i(t) \neq X_j(t)\ \forall t \in [0,T]$ and any $i \neq j$; $i,j = 1,2,\ldots,n$). This means the order at which the processes start with has to be maintained throughout the duration. That is, if $X_1(0) < X_2(0) < \ldots < X_n(0)$, then the processes $X_1(t),\ldots,X_n(t)$ are *non-colliding* up to time T only if $X_1(t) < X_2(t) < \ldots < X_n(t)$ for all $t \in [0,T]$.

Alternatively, we can rephrase the non-colliding condition in terms of staying within a special domain. Let W_n denote the n-dimensional *Weyl chamber*, which is the set of all points $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ whose coordinates are ordered, i.e. $W_n := \{\mathbf{x} \in \mathbb{R}^n : x_1 <$

$x_2 < \ldots < x_n\}$. Then the event that a n-dimensional process does not exit W_n coincides with the event that n one-dimensional processes experience no collision. As the Weyl chamber perspective is equivalent to our first definition, in the following sections we equivalently use $\{\mathbf{X}(t) \in W_n, \forall t \in [0,T]\}$ where $\mathbf{X}(t) = (X_1(t), \ldots, X_n(t))$ to mean the event that the n one-dimensional processes $X_1(t), \ldots, X_n(t)$ experience no collisions up to some time T.

3.2 Non-colliding Stochastic Processes

Before we start discussing non-colliding Gaussian processes, we must present a fundamental formula for the analysis of non-colliding stochastic processes, which of course is also used to derive many results in non-colliding Gaussian processes. The study of non-colliding stochastic processes started before non-colliding Gaussian processes. It was first sparked by Karlin and McGregor. The formula they derived in their 1959 paper forms the basis for the analysis of non-colliding stochastic processes. It provides the transition probability of n independent stochastic processes having not collided during the time of going from $\mathbf{x} \in W_n$ to $\mathbf{y} \in W_n$.

Theorem 3.1 (Karlin and McGregor, 1959). *Let $\mathbf{X}(t) = (X_1(t), \ldots, X_n(t))$ be a n-dimensional process where $X_i(t)$ are independent strong Markov processes that have the same transition probability (density) $p(t_1, t_2, x, y)$ for $i = 1, \ldots, n$. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in W_n$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in W_n$. Given $\mathbf{X}(0) = \mathbf{x}$, the probability that $\mathbf{X}(T) = \mathbf{y}$ and $\mathbf{X}(t) \in W_n$ for all $t \in [0,T]$ (i.e. $X_1(t), \ldots, X_n(t)$ do not collide up to time T) is given by*

$$\begin{aligned}
&\mathbb{P}^{\mathbf{x}}\{\mathbf{X}(T) = \mathbf{y}; \mathbf{X}(t) \in W_n, \ \forall t \in [0,T]\} \\
&= \mathbb{P}^{\mathbf{x}}\{\mathbf{X}(T) = \mathbf{y}; \textit{no collision up to time } T\} \\
&= \begin{vmatrix} p(0,T,x_1,y_1) & \ldots & p(0,T,x_1,y_n) \\ \vdots & & \vdots \\ p(0,T,x_n,y_1) & \ldots & p(0,T,x_n,y_n) \end{vmatrix} \\
&= \det(p(0,T,x_i,y_j))_{i,j=1}^n.
\end{aligned}$$

The Karlin-McGregor formula can be applied to any stochastic process with the *strong Markov property* (see A.5) to find their non-colliding transition probability. It has been used to derive many non-colliding results on a variety of stochastic processes, as well as aiding in the analysis and development of other models. Some examples of the processes and models that have been studied in the literature include, but not limited to, random walk (König et al., 2002), corner-growth model (Johansson, 2002), Poisson process (O'Connell, 2002), Yule process (Doumerc, 2005) and squared Bessel process (König et al., 2001), as well as specific cases of Gaussian processes such as Brownian motion and the Ornstein-Uhlenbeck process.

3.3 Non-colliding Brownian Motions

The study of non-colliding Gaussian processes started with Brownian motion. The interest of non-colliding Brownian motions was sparked by Dyson (1962). Dyson proposed a model of n Brownian particles system with mutual electrostatic repulsions which are proportional to the inverse of the distance between particles to describe the eigenvalues of a Hermitian matrix whose elements execute independent Brownian motions,

$$d\lambda_i = dB_i + \sum_{j \neq i} \frac{1}{\lambda_i - \lambda_j} dt,$$

where λ_i are the eigenvalues and B_i are independent standard one-dimensional Brownian motion. It turned out the dynamics of the eigenvalues through time are the same as a system of Brownian motions conditioned to never collide with each other. Dyson's seminal paper paved the way for research in non-colliding Brownian motions and advances in random matrix theory. Many still refer to non-colliding Brownian motions conditioned to never collide as Dyson's Brownian motions or a Dyson process. Since the discovery, further research has occurred on non-colliding Brownian motions, some developments include non-colliding Brownian motions on a circle (Hobson and Werner, 1996), non-colliding Brownian motions with an absorbing wall (Katori et al., 2003), and non-colliding system of reflecting Brownian motions (Katori and Tanemura, 2004).

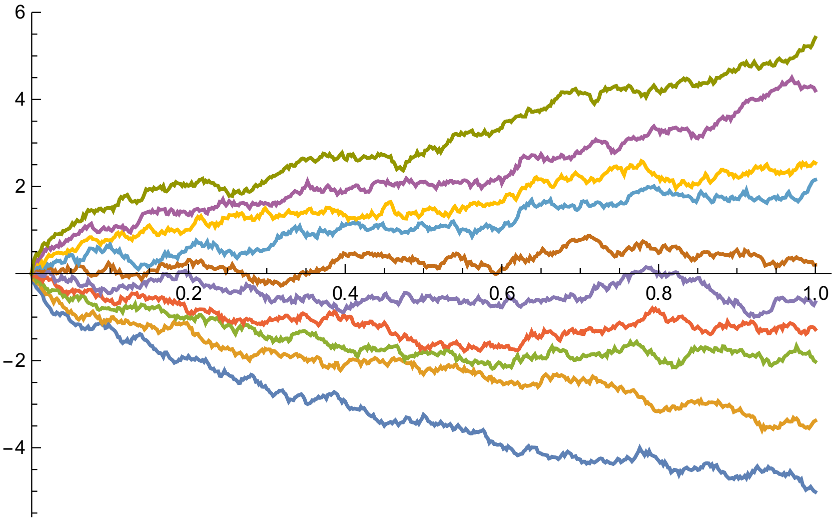


Figure 3.1: An illustration of Dyson Brownian motions. The paths are obtained by discretizing the x-axis into 500 equally spaced points.

Later, Grabiner (1999) derived a classic asymptotic result for non-colliding Brownian motions.

Theorem 3.2 (Grabiner 1999). *Let $X_1(t), \ldots, X_n(t)$ be n independent Brownian motions, and let $\tau = \inf\{t \geq 0 : X_i(t) = X_j(t); i \neq j\}$ be the first time any of the n processes experience collision. Then for any starting positions $\mathbf{x} \in W_n$, as $T \to \infty$,*

$$\mathbb{P}^{\mathbf{x}}\{\tau > T\} \sim T^{\frac{-n(n-1)}{4}} (2\pi)^{-\frac{n}{2}} \frac{h(\mathbf{x})}{\prod_{j=0}^{n-1} j!} \int_{\mathbf{y} \in W_n} \exp\left(-\frac{|\mathbf{y}|^2}{2}\right) h(\mathbf{y}) d\mathbf{y}$$

where

$$h(\mathbf{x}) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

is the Vandermonde determinant.

The Weyl Chamber perspective of approaching the non-colliding problem was actually introduced by Grabiner. Grabiner's asymptotic result showed that the probability of n independent Brownian motions experiencing no collisions up to time T is asymptotic to a constant multiple of $T^{\frac{-n(n-1)}{4}}$ as $T \to \infty$, and the constant is a polynomial of the starting positions $\mathbf{x}$. Next, we give a similar asymptotic result for non-colliding Ornstein-Uhlenbeck processes in Section 3.4.

3.4 Non-colliding Ornstein-Uhlenbeck Processes

Similar to non-colliding Brownian motions, non-colliding Ornstein-Uhlenbeck processes have mostly been studied in the random matrix literature due to its connection with the eigenvalues of a matrix-valued Ornstein-Uhlenbeck process. In this section, we present an asymptotic result for non-colliding Ornstein-Uhlenbeck processes.

Let us first show that the Ornstein-Uhlenbeck process possesses the strong Markov property so that the Karlin-McGregor formula can be applied. We use a theorem proven in Oksendal (2013) which states that

Theorem 3.3 (Oksendal 2013). *If a stochastic process $\{X(t); t \in T\}$ is a Itô diffusion, then it has the strong Markov property.*

Therefore, we only need to show that the Ornstein-Uhlenbeck process is a Itô diffusion. We recall that a (time-homogeneous) *Itô diffusion* is a stochastic process $X(t) : [0, \infty) \times \Omega \to \mathbb{R}^n$ satisfying a stochastic differential equation of the form

$$dX(t) = b(X(t))dt + \widetilde{\sigma}(X(t))dB(t), \qquad t \geq s; \quad X(s) = x \tag{3.1}$$

where $B(t)$ is m-dimensional Brownian motion and $b : \mathbb{R}^n \to \mathbb{R}^n$, $\widetilde{\sigma} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ satisfying the Lipschitz condition

$$|b(x) - b(y)| + |\widetilde{\sigma}(x) - \widetilde{\sigma}(y)| \leq D|x - y| \qquad x, y \in \mathbb{R}^n$$

for some constant $D > 0$ where $|\widetilde{\sigma}(x)| := (\sum |\widetilde{\sigma}_{ij}(x)|^2)^{1/2}$. This is easily checked, as it is well known that an Ornstein-Uhlenbeck process satisfies the stochastic differential equation

$$dX(t) = -\theta X(t)dt + \sigma dB(t)$$

where $\theta > 0$ and $\sigma > 0$ are parameters. Substituting into (3.1), we obtain $b(x) = -\theta x$ and $\widetilde{\sigma}(x) = \sigma$. Then we can see that

$$|b(x) - b(y)| + |\widetilde{\sigma}(x) - \widetilde{\sigma}(y)| = \theta|y - x|$$

which is less than or equal to $D|x-y|$ if we choose $D \geq \theta$. Thus, the Ornstein-Uhlenbeck process is a Itô diffusion and possesses the strong Markov property by Theorem 3.3.

We now present an asymptotic non-colliding result similar to Grabiner (1999) but for n Ornstein-Uhlenbeck processes. To the best of our knowledge, this result has not been shown in the literature.

Theorem 3.4. *Let $X_1(t), \ldots, X_n(t)$ be n independent Ornstein-Uhlenbeck processes, and let $\tau = \inf\{t \geq 0 : X_i(t) = X_j(t); i \neq j\}$ be the first time any of the n processes experience collision. Then for any starting positions $\mathbf{x} \in W_n$, as $T \to \infty$,*

$$\mathbb{P}^{\mathbf{x}}\{\tau > T\} \sim e^{\frac{-\theta n(n-1)}{2}T} 2^{\frac{n(n-1)}{2}} \pi^{-\frac{n}{2}} \left(\frac{\theta}{\sigma^2}\right)^{\frac{n(n-1)}{4}} \frac{h(\mathbf{x})}{\prod_{j=0}^{n-1} j!} \int_{\mathbf{z} \in W_n} \exp(-|\mathbf{z}|^2) h(\mathbf{z}) d\mathbf{z}.$$

We use the same method as Puchała (2005) to prove Theorem 3.4. A lemma given in Puchała (2005) will also be used in our proof.

Lemma 3.5 (Lemma 2 from Puchała (2005))**.** *Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in W_n$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in W_n$ where W_n is the n-dimensional Weyl chamber. Define E_k to be the following*

$$E_k := \sum_{\sigma \in S_n} \frac{\operatorname{sgn}(\sigma)}{k!} (x_1 y_{\sigma(1)} + \ldots + x_n y_{\sigma(n)})^k$$

where S_n denotes the set of permutations of $\{1, 2, \ldots, n\}$, $\operatorname{sgn}(\sigma)$ is the sign of permutation σ and k is a non-negative integer. Then for $i = 0, 1, \ldots, \frac{n(n-1)}{2} - 1$, we have $E_i = 0$ and

$$E_{\frac{n(n-1)}{2}} = \frac{h(\mathbf{x}) h(\mathbf{y})}{\prod_{j=0}^{n-1} j!}.$$

We now present the proof for Theorem 3.4.

Proof. The transition density of an Ornstein-Uhlenbeck process $X := \{X(t); t \geq 0\}$ going from x and reaching y at time t that satisfies

$$dX(t) = -\theta X(t) dt + \sigma dB(t), \qquad X(0) = x,$$

is given by

$$p^{OU}(0, t, x, y) = \sqrt{\frac{\theta}{\pi \sigma^2 (1 - e^{-2\theta t})}} \exp\left[\frac{-\theta (y - x e^{-\theta t})^2}{\sigma^2 (1 - e^{-2\theta t})}\right], \qquad y \in \mathbb{R}.$$

Since Ornstein-Uhlenbeck process is a strong Markov process, we can use the Karlin-McGregor formula to find the non-colliding transition probability. According to the Karlin-McGregor formula, the transition probability of n independent Ornstein-Uhlenbeck processes, $\mathbf{X}(t) := (X_1(t), \ldots, X_n(t))$, starting at $\mathbf{x} \in W_n$ and arriving at $\mathbf{y} \in W_n$ at time T without having experienced any collisions in the intervening time is given by

$$\begin{aligned}
&\mathbb{P}^{\mathbf{x}}\{\mathbf{X}(T) = \mathbf{y}; \mathbf{X}(t) \in W_n, \forall t \in [0, T]\} \\
&= \det(p^{OU}(0, T, x_i, y_i))_{i,j=1}^n \\
&= \left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}} \exp\left[\frac{-\theta(|\mathbf{y}|^2 + |\mathbf{x}|^2 e^{-2\theta T})}{\sigma^2(1-e^{-2\theta T})}\right] \det\left[\exp\left(\frac{2\theta e^{-\theta T} x_i y_i}{\sigma^2(1-e^{-2\theta T})}\right)\right]_{i,j=1}^n \\
&= \left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}} \exp\left[\frac{-\theta(|\mathbf{y}|^2 + |\mathbf{x}|^2 e^{-2\theta T})}{\sigma^2(1-e^{-2\theta T})}\right] \sum_{k=0}^{\infty} \left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k
\end{aligned}$$

where $|\mathbf{x}|^2 = \sum_{i=1}^n x_i^2$ and

$$E_k = \sum_{\delta \in S_n} \frac{\text{sgn}(\delta)}{k!} (x_1 y_{\delta(1)} + \ldots + x_n y_{\delta(n)})^k$$

S_n is the set of permutations of $\{1, 2, \ldots, n\}$ and $\text{sgn}(\delta)$ is the sign of permutation δ.

Let $\tau = \inf\{t \geq 0 : X_i(t) = X_j(t); i \neq j\}$ be the first time any of the n processes experience collision. Then the probability that $\mathbf{X}(t)$ starting from $\mathbf{x} \in W_n$ and experience no collisions up to at least time T is given by

$$\mathbb{P}^{\mathbf{x}}\{\tau > T\} = \int_{\mathbf{y} \in W_n} \det(p^{OU}(0, T, x_i, y_i))_{i,j=1}^n d\mathbf{y}$$

Then as $T \to \infty$,

$$\begin{aligned}
\lim_{T\to\infty} \frac{\mathbb{P}^{\mathbf{x}}\{\tau > T\}}{e^{\frac{-\theta n(n-1)}{2}T}} &= \lim_{T\to\infty} e^{\frac{\theta n(n-1)}{2}T} \int_{\mathbf{y} \in W_n} \det(p^{OU}(0, T, x_i, y_i))_{i,j=1}^n d\mathbf{y} \\
&= \lim_{T\to\infty} e^{\frac{\theta n(n-1)}{2}T} \left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}} \exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right] \\
&\quad \int_{\mathbf{y} \in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right] \sum_{k=0}^{\infty} \left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k d\mathbf{y} \\
&= \lim_{T\to\infty} e^{\frac{\theta n(n-1)}{2}T} \left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}} \exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]
\end{aligned}$$

$$\begin{aligned}
&\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right]\sum_{k=0}^{\frac{n(n-1)}{2}}\left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k d\mathbf{y}\\
&+e^{\frac{\theta n(n-1)}{2}T}\left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]\\
&\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right]\sum_{k=\frac{n(n-1)}{2}+1}^{\infty}\left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k d\mathbf{y}\\
=\;&(1)+(2).
\end{aligned}$$

Consider the first part of the sum above,

$$\begin{aligned}
(1)=\lim_{T\to\infty} &\, e^{\frac{\theta n(n-1)}{2}T}\left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]\\
&\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right]\sum_{k=0}^{\frac{n(n-1)}{2}}\left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k d\mathbf{y}
\end{aligned}$$

By Lemma 3.5, this can be simplified to

$$\begin{aligned}
&=\lim_{T\to\infty} e^{\frac{\theta n(n-1)}{2}T}\left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]\\
&\qquad\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right]\left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n(n-1)}{2}} E_{\frac{n(n-1)}{2}} d\mathbf{y}\\
&=\lim_{T\to\infty} 2^{\frac{n(n-1)}{2}}\pi^{-\frac{n}{2}}\left(\frac{\theta}{\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n^2}{2}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]\frac{h(\mathbf{x})}{\prod_{j=0}^{n-1}j!}\\
&\qquad\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right]h(\mathbf{y})d\mathbf{y}
\end{aligned}$$

Let $z_i=\sqrt{\frac{\theta}{\sigma^2(1-e^{-2\theta T})}}y_i$, and since $h(ax)=a^{\frac{n(n-1)}{2}}h(x)$, we have

$$\begin{aligned}
&=\lim_{T\to\infty} 2^{\frac{n(n-1)}{2}}\pi^{-\frac{n}{2}}\left(\frac{\theta}{\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n(n-1)}{4}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]\frac{h(\mathbf{x})}{\prod_{j=0}^{n-1}j!}\\
&\qquad\int_{\mathbf{z}\in W_n}\exp(-|\mathbf{z}|^2)h(\mathbf{z})d\mathbf{z}\\
&=2^{\frac{n(n-1)}{2}}\pi^{-\frac{n}{2}}\left(\frac{\theta}{\sigma^2}\right)^{\frac{n(n-1)}{4}}\frac{h(\mathbf{x})}{\prod_{j=0}^{n-1}j!}\int_{\mathbf{z}\in W_n}\exp(-|\mathbf{z}|^2)h(\mathbf{z})d\mathbf{z}.
\end{aligned}$$

Now let us consider the second part of the sum from earlier,

$$(2)=\lim_{T\to\infty} e^{\frac{\theta n(n-1)}{2}T}\left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}}\exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]$$

$$\int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right] \sum_{k=\frac{n(n-1)}{2}+1}^{\infty} \left(\frac{2\theta e^{-\theta T}}{\sigma^2(1-e^{-2\theta T})}\right)^k E_k d\mathbf{y}$$

$$= \lim_{T\to\infty} \left(\frac{\theta}{\pi\sigma^2(1-e^{-2\theta T})}\right)^{\frac{n}{2}} \exp\left[\frac{-\theta|\mathbf{x}|^2}{\sigma^2(e^{2\theta T}-1)}\right]$$

$$\sum_{k=\frac{n(n-1)}{2}+1}^{\infty} e^{-\theta T[k-\frac{n(n-1)}{2}]} \left(\frac{2\theta}{\sigma^2(1-e^{-2\theta T})}\right)^k \int_{\mathbf{y}\in W_n} \exp\left[\frac{-\theta|\mathbf{y}|^2}{\sigma^2(1-e^{-2\theta T})}\right] E_k d\mathbf{y}$$

$$= 0.$$

We have shown that

$$\lim_{T\to\infty} \frac{\mathbb{P}^{\mathbf{x}}\{\tau > T\}}{e^{\frac{-\theta n(n-1)}{2}T}} = 2^{\frac{n(n-1)}{2}} \pi^{-\frac{n}{2}} \left(\frac{\theta}{\sigma^2}\right)^{\frac{n(n-1)}{4}} \frac{h(\mathbf{x})}{\prod_{j=0}^{n-1} j!} \int_{\mathbf{z}\in W_n} \exp(-|\mathbf{z}|^2) h(\mathbf{z}) d\mathbf{z}$$

$$\therefore \mathbb{P}^{\mathbf{x}}\{\tau > T\} \sim e^{\frac{-\theta n(n-1)}{2}T} 2^{\frac{n(n-1)}{2}} \pi^{-\frac{n}{2}} \left(\frac{\theta}{\sigma^2}\right)^{\frac{n(n-1)}{4}} \frac{h(\mathbf{x})}{\prod_{j=0}^{n-1} j!} \int_{\mathbf{z}\in W_n} \exp(-|\mathbf{z}|^2) h(\mathbf{z}) d\mathbf{z}.$$

□

The asymptotic result we proved here show that the probability of n independent Ornstein-Uhlenbeck processes experiencing no collisions up to time T is asymptotic to a constant multiple of $e^{\frac{-\theta n(n-1)}{2}T}$ as $T \to \infty$, and the constant is a polynomial of the starting positions $\mathbf{x}$.

3.5 Conclusion

In this chapter, we presented the fundamental theorem derived by Karlin and McGregor that is used extensively in the analysis of not just non-colliding Gaussian processes but also other stochastic processes, as well as a key asymptotic result for non-colliding Brownian motions by Grabiner. Then, by utilising the strong Markov property of Ornstein-Uhlenbeck process and the Karlin-McGregor formula, we derived a similar asymptotic result for non-colliding Ornstein-Uhlenbeck processes. In the next chapter, we propose a methodology of imposing a non-colliding constraint when modelling with multiple Gaussian process regressions at the same time.

Chapter 4

Non-colliding GP Regressions

In this chapter, we present new results on how to perform regression in the situation where we have a number of Gaussian processes under a non-colliding constraint. We first show that the non-colliding Gaussian processes problem can be mapped to a sequence of Gaussian processes with inequality constraints in 4.1. Then, in 4.2, we will review the literature on Gaussian processes with inequality constraints and present a method that we will use to tackle the non-colliding problem. Lastly, we will demonstrate how to perform non-colliding Gaussian process regressions in 4.3.

4.1 Reposing the Non-colliding Problem

Recall from Chapter 3 that the non-colliding condition on n Gaussian processes Y_1, Y_2, …, Y_n with index set T can be written as $Y_1(t) < Y_2(t) < ... < Y_n(t)$ for all $t \in T$. Another way to look at this is that for any $Y_i(t)$, its distance to the value of the adjacent processes $Y_{i-1}(t)$ and $Y_{i+1}(t)$ has to be greater than 0. This means that the non-colliding condition is equivalent to the processes satisfying the inequalities

$$\begin{aligned} Y_n(t) - Y_{n-1}(t) &> 0, \\ &\vdots \\ Y_3(t) - Y_2(t) &> 0, \end{aligned}$$

$$Y_2(t) - Y_1(t) > 0.$$

The difference of two independent Gaussian processes is also a Gaussian process. This follows directly from the property that a linear combination of independent Gaussian random variables is a Gaussian random variable. Suppose A and B are two independent Gaussian processes defined by

$$A \sim \mathcal{GP}(m_A, k_A)$$
$$B \sim \mathcal{GP}(m_B, k_B)$$

Then the difference $A - B$ is given by

$$A - B \sim \mathcal{GP}(m_A - m_B, k_A + k_B).$$

Since the difference of two Gaussian processes is also a Gaussian process, the non-colliding condition of n Gaussian processes can be viewed as $n-1$ Gaussian processes satisfying the positivity constraint

$$\begin{aligned} Z_{n-1}(t) &> 0, \\ &\vdots \\ Z_2(t) &> 0, \\ Z_1(t) &> 0, \end{aligned}$$

where $Z_i(t) := Y_{i+1}(t) - Y_i(t)$ are Gaussian processes. This leads us to consider the problem of Gaussian process regression with inequality constraints.

4.2 GP Regression with Inequality Constraints

As discussed in Chapter 1, so far, there are mainly two approaches to add inequality constraints to Gaussian process regression. One approach discretises the input space into a set of virtual observation locations and simulates a conditional Gaussian process which satisfies the inequality constraints at these locations. However, since the inequality constraints are only guaranteed to hold at these virtual observation locations, only a finite number of input locations satisfies the constraints under this approach.

The second approach uses a piecewise linear approximation of Gaussian process which allows the constraints to hold in the entire domain. This approach is more appropriate for our purpose as we would like the non-colliding constraint to hold in the entire domain. We now briefly review the latest paper following this approach by López-Lopera et al. (2018)

Consider the compact input space $\mathcal{X} := [0, 1]$. For a zero-mean Gaussian process Y with covariance function k, López-Lopera et al. use a finite-dimensional approximation Y_m with knots at $t_1, ..., t_m$ to handle the inequality constraints. Let

$$Y_m(x) := \sum_{j=1}^{m} Y(t_j)\phi_j(x)$$

where $\phi_1, ..., \phi_m$ are hat basis functions given by

$$\phi_j(x) := \begin{cases} 1 - \left|\frac{x-t_j}{\Delta_m}\right| & \text{if } \left|\frac{x-t_j}{\Delta_m}\right| \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

This is the case considered in López-Lopera et al. (2018) where the knots are equally spaced, i.e. $t_j = (j-1)\Delta_m$ with $\Delta_m = 1/(m-1)$. However, this assumption can be relaxed and we will show in Section 4.3.2 how the hat basis function would be modified when the knots are not equally spaced.

Let $\mathcal{E}_j := Y(t_j)$ for $j = 1, 2, ..., m$. Let C be the set of q linear inequalities given by

$$C := \{\mathbf{c} \in \mathbb{R}^m; \forall k = 1, ..., q : l_k \leq \sum_{j=1}^{m} \lambda_{k,j} c_j \leq u_k\},$$

where the $\lambda_{k,j}$'s encode the linear operations, and the l_k's and u_k's are the lower and upper bounds. Denote $\Lambda := (\lambda_{k,j})_{1 \leq k \leq q, 1 \leq j \leq m}$, $\mathbf{l} := (l_k)_{1 \leq k \leq q}$, and $\mathbf{u} := (u_k)_{1 \leq k \leq q}$. The vector $\boldsymbol{\mathcal{E}}$, which contains the values at the knots, is a zero-mean Gaussian vector with covariance matrix $\Gamma = (k(t_i, t_j))_{1 \leq i,j \leq m}$.

Let $\mathbf{y} = [y_1, y_2, ..., y_n]^T$ be a realisation of the Gaussian process Y at the points $x_1, x_2, ..., x_n$. Then we can construct the $n \times m$ matrix of hat basis functions $\mathbf{\Phi}$, which is given by $\mathbf{\Phi}_{i,j} = \phi_j(x_i)$. By using the hat functions and our process approximation Y_m, the inequality constraints on the process Y are now approximated by a finite number of constraints on $\boldsymbol{\mathcal{E}} = [\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_m]^T$ (Maatouk and Bay, 2017). As a result, the interpolation conditions and inequality conditions on the finite approximation Y_m can then be written as

$$\boldsymbol{\mathcal{E}} \sim \mathcal{N}(\mathbf{0}, \Gamma) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\Phi}\boldsymbol{\mathcal{E}} = \mathbf{y} & \text{(Interpolation conditions),} \\ \boldsymbol{\mathcal{E}} \in C \Leftrightarrow \mathbf{l} \le \Lambda\boldsymbol{\mathcal{E}} \le \mathbf{u} & \text{(Inequality conditions).} \end{cases}$$

Therefore, we want to find the posterior distribution of $\boldsymbol{\mathcal{E}}$ such that both interpolation and inequality conditions are met. The conditional distribution satisfying the interpolation conditions is $\boldsymbol{\mathcal{E}}|\{\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} = \mathbf{y}\} \sim \mathcal{N}(\mu, \Sigma)$ with

$$\mu = \Gamma\boldsymbol{\Phi}^T[\boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T]^{-1}\mathbf{y}, \quad \text{and} \quad \Sigma = \Gamma - \Gamma\boldsymbol{\Phi}^T[\boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T]^{-1}\boldsymbol{\Phi}\Gamma.$$

It follows that the posterior distribution satisfying both interpolation conditions and inequality conditions is

$$\Lambda\boldsymbol{\mathcal{E}}|\{\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} = \mathbf{y}, \mathbf{l} \le \Lambda\boldsymbol{\mathcal{E}} \le \mathbf{u}\} \sim \mathcal{TN}(\Lambda\mu, \Lambda\Sigma\Lambda^T, \mathbf{l}, \mathbf{u})$$

where $\mathcal{TN}(\Lambda\mu, \Lambda\Sigma\Lambda^T, \mathbf{l}, \mathbf{u})$ is the truncated multivariate normal distribution with mean $\Lambda\mu$, covariance matrix $\Lambda\Sigma\Lambda^T$ and lower and upper bounds $\mathbf{l}, \mathbf{u}$. Sampling from the posterior distribution will give us $\Lambda\boldsymbol{\mathcal{E}}$. Then we can solve the linear system $\Lambda\boldsymbol{\mathcal{E}}$ to obtain $\boldsymbol{\mathcal{E}}$. Finally, let $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_m]^T$ be the posterior modes or means of $\boldsymbol{\mathcal{E}}$, then the prediction at a point $\widetilde{x}$ is given by $\sum_{j=1}^m \lambda_j \phi_j(\widetilde{x})$.

4.2.1 Higher Dimensional Input Spaces

López-Lopera et al. also present 2-dimensional input spaces in their paper. The model can be extended to higher dimensional cases following the same idea. With 2-dimensional input spaces $\mathcal{X} = [0, 1]^2$, we have $m_1 \times m_2$ knots, and the finite approximation is given by

$$Y_{m_1,m_2}(x, z) = \sum_{j=1}^{m_1}\sum_{k=1}^{m_2} \mathcal{E}_{j,k}\phi_j^1(x)\phi_k^2(z).$$

Let $\boldsymbol{\mathcal{E}} = [\mathcal{E}_{1,1}, \mathcal{E}_{1,2}, ..., \mathcal{E}_{1,m_2}, \mathcal{E}_{2,1}, ..., \mathcal{E}_{m_1,m_2}]$ be a Gaussian vector with zero mean and covariance matrix $\boldsymbol{\Gamma}$ as before. Now, let $\boldsymbol{\Phi}$ be the $n \times (m_1 \times m_2)$ matrix of hat basis functions where the

ith row is given by

$$[\phi_1^1(x_i)\phi_1^2(z_i)\cdots\phi_1^1(x_i)\phi_{m_2}^2(z_i)\cdots\phi_{m_1}^1(x_i)\phi_1^2(z_i)\cdots\phi_{m_1}^1(x_i)\phi_{m_2}^2(z_i)].$$

Finally, the posterior distribution can be computed as before.

4.3 GP Regressions with Non-colliding Constraint

We now show how to make use of the constrained Gaussian process regression model in López-Lopera et al. (2018) to impose the non-colliding constraint. For readability, we focus our discussion on a one-dimensional input space. However, the algorithms given below can be extended to higher dimensional input spaces by the approach in Section 4.2.1.

Consider the $n_1 \times 1$ vector $X = [x_1, x_2, ..., x_{n_1}]^T$ which contains the independent variable of the training set and the $n_2 \times 1$ test set $\widetilde{X} = [\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_{n_2}]^T$. We also observe l dependent variables $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$, where $\mathbf{y}_i$ is a $n_1 \times 1$ vector. For now, we only consider the case where $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ correspond to the same training points X. In Section 4.3.3 we will discuss how to deal with the situation of having different training input locations for each $\mathbf{y}_i$.

We first apply min-max normalisation[1] on X and $\widetilde{X}$ together, so that the input space $\mathcal{X}$ becomes $[0,1]$. Next, we order $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ in ascending order. This can be done by comparing either their minimums or maximums. Let $\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_l$ be $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ arranged in ascending order, i.e. $\mathbf{y}'_1 < \mathbf{y}'_2 < ... < \mathbf{y}'_l$. For each $\mathbf{y}'_i$, the user can specify a covariance function to model it with a Gaussian process. Let $f_1(\cdot), f_2(\cdot), ..., f_l(\cdot)$ be the Gaussian processes with covariance functions $k_1, k_2, ..., k_l$ that are used to model $\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_l$ and satisfy the non-colliding constraint. Then the Gaussian process regressions' predictions at $\widetilde{X}$, $f_1(\widetilde{X}), f_2(\widetilde{X}), ..., f_l(\widetilde{X})$, given the non-colliding constraint can be computed following Algorithm 1.

The idea is to take the differences between the processes that are next to each other and bound it by 0. After fitting Gaussian process regressions constrained to not cross 0 to the differences, then we transform it back to obtain non-colliding regressions as desired.

[1] For each x_i in $\mathbf{x}$, the min-max normalisation returns $z_i = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$

Algorithm 1: Non-colliding GP Regressions

```
Input: $X$: training data; $\widetilde{X}$: test data; $(\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_l)$: observed dependent variables
        in ascending order; $(k_1, k_2, ..., k_l)$: covariance functions; $(t_1, t_2, ..., t_m)$:
        locations of knots
Output: $f_1(\widetilde{X}), f_2(\widetilde{X}), ..., f_l(\widetilde{X})$
Compute the posterior mean for the first Gaussian process:
  $f_1(\widetilde{X}) = k_1(\widetilde{X}, X)k_1(X, X)^{-1}\mathbf{y}'_1$
for $i = 1, 2, ..., l-1$ do
    $\mathbf{z}_i = \mathbf{y}'_{i+1} - \mathbf{y}'_i$
    Construct the $n_1 \times m$ matrix of hat basis functions: $\mathbf{\Phi}$
    Construct the $m \times m$ covariance matrix $\Gamma = (k_i(t_j, t_p) + k_{i+1}(t_j, t_p))_{1 \leq j,p \leq m}$ where
      $t$ are the locations of knots.
    Compute the posterior mean of $\boldsymbol{\mathcal{E}}$: $\mu = \Gamma\mathbf{\Phi}^T[\mathbf{\Phi}\Gamma\mathbf{\Phi}^T]^{-1}\mathbf{z}_i$
    Compute the posterior variance of $\boldsymbol{\mathcal{E}}$: $\Sigma = \Gamma - \Gamma\mathbf{\Phi}^T[\mathbf{\Phi}\Gamma\mathbf{\Phi}^T]^{-1}\mathbf{\Phi}\Gamma$
    Simulate the distribution $\mathcal{TN}(\mu, \Sigma, \mathbf{0}, \infty)$, and compute the mean or mode $\boldsymbol{\lambda}$
    Calculate the vector of predicted differences: $\widetilde{\mathbf{z}}_i = \sum_{s=1}^{m} \lambda_s \phi_s(\widetilde{X})$
    Calculate the predictions of the Gaussian process $f_{i+1}$: $f_{i+1}(\widetilde{X}) = f_i(\widetilde{X}) + \widetilde{\mathbf{z}}_i$
```

4.3.1 Parameters Estimation

To estimate the parameters of the covariance functions k_i for $i = 1, \ldots, l$, we use the same approach as Section 2.2.3, where we choose the parameters that maximise the log-likelihood function. For non-colliding Gaussian process regressions, the parameters are estimated when fitting the differences of the processes. Let $\mathbf{z}_i := \mathbf{y}'_{i+1} - \mathbf{y}'_i$ be the difference between the vector of observed (or training) output values of the $(i+1)$th process and the vector of observed output values of the ith process as defined in Algorithm 1.

Also, let k_i^* denote the covariance function of the ith process f_i using the optimised parameters. Then, the parameters of covariance function k_i is estimated by fitting $\mathbf{z}_{i-1}$ using the covariance function $k_{i-1}^* + k_i$ and maximising the log-likelihood. The first process is fitted without constraints so the parameters of its covariance function can be estimated in the same way as Section 2.2.3.

The likelihood function of $\mathbf{z}_i$ with the non-colliding constraint can be found using Bayes' theorem,

$$p(\mathbf{z}_i | X, \boldsymbol{\theta}, \boldsymbol{\mathcal{E}} \in C) = \frac{p(\mathbf{z}_i | X, \boldsymbol{\theta}) p(\boldsymbol{\mathcal{E}} \in C | \mathbf{z}_i, X, \boldsymbol{\theta})}{p(\boldsymbol{\mathcal{E}} \in C | X, \boldsymbol{\theta})},$$

where $C := \{\mathbf{c} \in \mathbb{R}^m; \mathbf{c} > \mathbf{0}\}$ and $\boldsymbol{\theta}$ is the vector containing parameters of k_i. Then the estimated parameters $\hat{\boldsymbol{\theta}}$ is given by

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{z}_i|X,\boldsymbol{\theta},\boldsymbol{\mathcal{E}} \in C)\\ &= \arg\max_{\boldsymbol{\theta}} \big(\log p(\mathbf{z}_i|X,\boldsymbol{\theta}) + \log p(\boldsymbol{\mathcal{E}} \in C|\mathbf{z}_i,X,\boldsymbol{\theta}) - \log p(\boldsymbol{\mathcal{E}} \in C|X,\boldsymbol{\theta})\big).\end{aligned}$$

The first term is the log-likelihood without any constraints. The second and third term are Gaussian orthant probabilities given by

$$\begin{aligned}p(\boldsymbol{\mathcal{E}} \in C|\mathbf{z}_i,X,\boldsymbol{\theta}) &= \int_{(0,\infty)^m} (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}(\boldsymbol{\mathcal{E}}-\mu)^T \Sigma^{-1}(\boldsymbol{\mathcal{E}}-\mu)\Big)\, d\boldsymbol{\mathcal{E}}\\ p(\boldsymbol{\mathcal{E}} \in C|X,\boldsymbol{\theta}) &= \int_{(0,\infty)^m} (2\pi)^{-\frac{m}{2}} |\Gamma|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}\boldsymbol{\mathcal{E}}^T \Gamma^{-1}\boldsymbol{\mathcal{E}}\Big)\, d\boldsymbol{\mathcal{E}}\end{aligned}$$

where m is the number of knots, $\Gamma = (k_i(t_j,t_p) + k^*_{i-1}(t_j,t_p))_{1\leq j,p\leq m}$ and Σ, μ are given in Algorithm 1. These Gaussian orthant probabilities have to be computed numerically. One method is the minimax exponential tilting proposed in Botev (2017) that we consider in the next chapter.

4.3.2 Local Non-colliding Constraint

Sometimes one may wish to impose the non-colliding constraint only on an interval of the domain instead of the entire domain. Suppose we wish to impose the non-colliding constraint in the range (a,b) where $0 < a < b < 1$. Additionally, we have m knots $t_1,\dots,t_m$ and $t_1 < \dots < t_m$. If $t_j \neq a,b$ for $j = 1,\dots,m$, we add two additional knots at a and b. Since now the knots are likely to be no longer equally spaced, we need to modify the hat basis function to

$$\phi_j(x) := \begin{cases} 1 - \left|\frac{x-t_j}{t_j-t_{j-1}}\right| & \text{if } t_{j-1} \leq x \leq t_j,\\ 1 - \left|\frac{x-t_j}{t_{j+1}-t_j}\right| & \text{if } t_j \leq x \leq t_{j+1},\\ 0 & \text{otherwise.}\end{cases}$$

This modification achieves the same effect as the hat basis function formula when the knots are equally spaced given in López-Lopera et al. (2018), where $\phi_j(x) = 1$ when $x = t_j$ and

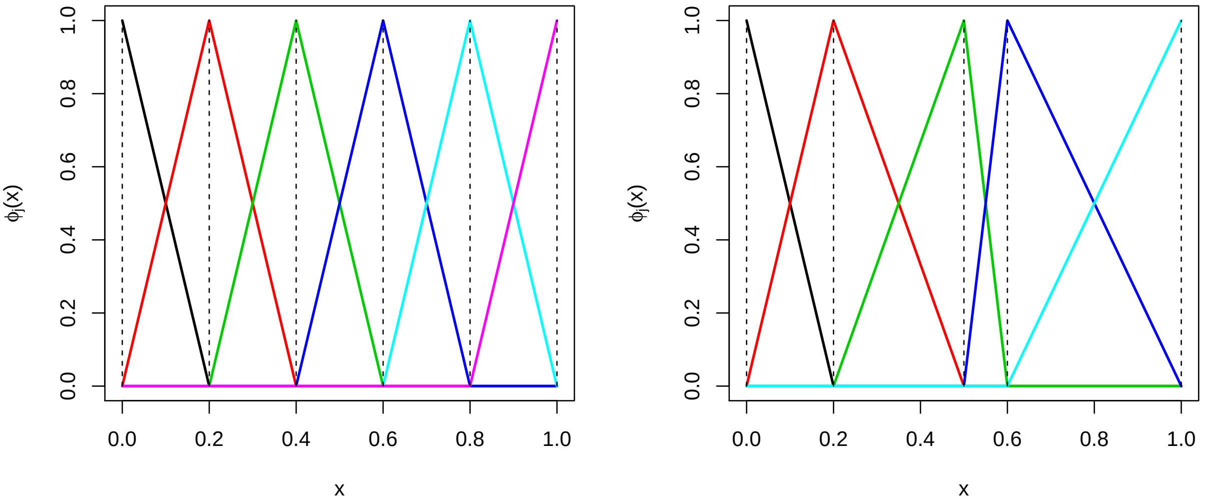


Figure 4.1: Left: Hat basis functions ϕ_j of equally spaced knots at (0, 0.2, 0.4, 0.6, 0.8, 1). **Right**: Hat basis functions ϕ_j of unequally spaced knots at (0, 0.2, 0.5, 0.6, 1).

decreases linearly to 0 as x moves towards adjacent knots t_{j-1} and t_{j+1}. Once x reaches the adjacent knots, the function becomes 0 and stays 0 as x moves beyond the adjacent knots. Furthermore, the sum of all hat basis functions equals 1 for all $x \in [0,1]$ just as before. This is illustrated in Figure 4.1, where each colour represents the hat basis function corresponding to a knot and the dotted lines show the locations of the knots.

Let $t_\alpha = a$ and $t_\beta = b$. We re-order the knots so we have $t_1 < \ldots < t_\alpha < \ldots < t_\beta < \ldots < t_{m+2}$. To implement Gaussian process regressions with local non-colliding constraint, we follow the steps given in Algorithm 1 as before, but now the matrix of hat basis functions is constructed using the modified hat basis function, and instead of simulating $\mathcal{TN}(\mu, \Sigma, \mathbf{0}, \infty)$, we now simulate $\mathcal{TN}(\mu, \Sigma, \mathbf{L}, \infty)$ where $\mathbf{L} := (L_j)_{1 \le j \le m}$ and L_j is given by

$$L_j = \begin{cases} -\infty & \text{for } j = 1, \ldots, \alpha \\ 0 & \text{for } j = \alpha + 1, \ldots, \beta - 1 \\ -\infty & \text{for } j = \beta, \ldots, m+2 \end{cases}$$

This enforces the non-colliding constraint only on the interval (a, b) and no restrictions are in place when outside this interval. Local non-colliding constraint can also be applied to multiple intervals. It can be easily done in the same way by adding more knots to the desired locations.

4.3.3 Different Training Input Locations

We have shown how to implement non-colliding Gaussian process regressions when the observed $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ all share the same training inputs X and have the same length. However, in practice $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ may be collected independently instead of all at the same time. This may result in $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ correspond to different input locations and have different lengths. To tackle this problem, we modify Algorithm 1 a little so that instead of taking the differences using $\mathbf{y}_i$'s we use the predicted values from previous regressions. However, this process can be quite expensive computationally as each Gaussian process needs to predict not only the test set but also the other inputs. Suppose now we have training inputs $X_1, X_2, ..., X_l$ and they may not have the same values or lengths. Consequently, $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l$ could have different lengths. The modified algorithm is then given by Algorithm 2.

Algorithm 2: Non-colliding GP Regressions with Different Training Input Locations

```
Input: $(X_1, X_2, ..., X_l)$, $\widetilde{X}$, $(\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_l)$, $(k_1, k_2, ..., k_l)$, $(t_1, t_2, ..., t_m)$
Output: $f_1(\widetilde{X}), f_2(\widetilde{X}), ..., f_l(\widetilde{X})$
Compute the posterior mean for the first Gaussian process:
  $f_1(\widetilde{X}) = k_1(\widetilde{X}, X_1)k_1(X_1, X_1)^{-1}\mathbf{y}'_1$
Let $X' = [X_2^T, X_3^T, ..., X_l^T]^T$ and calculate $f_1(X')$
for $i = 1, 2, ..., l-1$ do
    $\mathbf{z}_i = \mathbf{y}'_{i+1} - f_i(X_i)$
    Construct the $n_1 \times m$ matrix of hat basis functions: $\mathbf{\Phi}$
    Construct the $m \times m$ covariance matrix $\Gamma = (k_i(t_j, t_p) + k_{i-1}(t_j, t_p))_{1 \le j,p \le m}$ where
      $t$ are the locations of knots.
    Compute the posterior mean of $\mathcal{E}$: $\mu = \Gamma\mathbf{\Phi}^T[\mathbf{\Phi}\Gamma\mathbf{\Phi}^T]^{-1}\mathbf{z}_i$
    Compute the posterior variance of $\mathcal{E}$: $\Sigma = \Gamma - \Gamma\mathbf{\Phi}^T[\mathbf{\Phi}\Gamma\mathbf{\Phi}^T]^{-1}\mathbf{\Phi}\Gamma$
    Simulate the distribution $\mathcal{TN}(\mu, \Sigma, \mathbf{0}, \infty)$, and compute the mean or mode $\boldsymbol{\lambda}$
    Calculate the vector of predicted differences: $\widetilde{\mathbf{z}}_i = \sum_{s=1}^{m} \lambda_s \phi_s(\widetilde{X})$
    Calculate the predictions of the Gaussian process $f_{i+1}$: $f_{i+1}(\widetilde{X}) = f_i(\widetilde{X}) + \widetilde{\mathbf{z}}_i$
    Let $X' = [X_{i+1}^T, X_{i+2}^T, ..., X_l^T]^T$, compute $f_{i+1}(X')$
```

4.3.4 Noisy Observations

López-Lopera et al. (2018) only considered the interpolation problem which assumes the observations are noise-free. The resulting Gaussian process regression would pass through all the observations. However, we don't always have access to noise-free data so it is important to also consider the case where our observations are contaminated by noise.

Let $K = (k(x_i, x_j))_{1 \leq i,j \leq n_1}$ where k is a covariance function and $x_i \in X$ for $i = 1, 2, \ldots, n_1$ are the training inputs. Then instead of having $\mathbf{y} \sim \mathcal{N}(0, K)$ in the noise-free case, we now have $\mathbf{y} \sim \mathcal{N}(0, K + \sigma_n^2 I)$, where σ_n^2 is the variance of the noise term and I is the $n_1 \times n_1$ identity matrix. Under the approximation method given in López-Lopera et al. (2018), the Gaussian process is approximated by hat basis functions and a Gaussian random vector $\boldsymbol{\mathcal{E}} \sim \mathcal{N}(0, \Gamma)$ such that $\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} = \mathbf{y}$ where $\boldsymbol{\Phi}$ is the matrix of hat basis functions. In the noise-free case, $\mathbf{y} \sim \mathcal{N}(0, K)$ is approximated by $\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} \sim \mathcal{N}(0, \Phi\Gamma\Phi^T)$. To add noise, we make the same adjustment as earlier by adding $\sigma_n^2 I$, so now we have $\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} \sim \mathcal{N}(0, \Phi\Gamma\Phi^T + \sigma_n^2 I)$.

The covariance between $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\Phi}\boldsymbol{\mathcal{E}}$ can be found by

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{\mathcal{E}}, \boldsymbol{\Phi}\boldsymbol{\mathcal{E}}) &= \mathbb{E}[\boldsymbol{\mathcal{E}}(\boldsymbol{\Phi}\boldsymbol{\mathcal{E}})^T] \\ &= \mathbb{E}[\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{E}}^T]\boldsymbol{\Phi}^T \\ &= \Gamma\boldsymbol{\Phi}^T.\end{aligned}$$

Then the joint distribution of $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\Phi}\boldsymbol{\mathcal{E}}$ is given by

$$\begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\Phi}\boldsymbol{\mathcal{E}} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Gamma & \Gamma\boldsymbol{\Phi}^T \\ \boldsymbol{\Phi}\Gamma^T & \boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T + \sigma_n^2 I \end{bmatrix} \right)$$

Now we can use Theorem 2.1 to find the conditional distribution of $\boldsymbol{\mathcal{E}} | \{\boldsymbol{\Phi}\boldsymbol{\mathcal{E}} = \mathbf{y}\} \sim \mathcal{N}(\mu^*, \Sigma^*)$, where

$$\begin{aligned}\mu^* &= \Gamma\boldsymbol{\Phi}^T[\boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T + \sigma_n^2 I]^{-1}\mathbf{y} \\ \Sigma^* &= \Gamma - \Gamma\boldsymbol{\Phi}^T[\boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T + \sigma_n^2 I]^{-1}\boldsymbol{\Phi}\Gamma^T.\end{aligned}$$

When implementing non-colliding Gaussian process regressions in the case of noisy observations, simply compute μ^* and Σ^* in Step 7 and 8 of the algorithms, instead of μ and Σ.

The user can specify the noise variance or use the parameters estimation method given in Section 4.3.1 to find the σ_n^2 that maximises the likelihood.

4.3.5 Modelling the Differences Explicitly

In previous sections, when performing non-colliding Gaussian process regressions, even though we fit the differences then transforming them back, we still specify what model to use to fit each of the underlying function by specifying the covariance functions. The covariance functions used in the modelling of the differences are not specified directly. They are simply a transformation of the covariance functions used to model the underlying functions. This is not unusual and is in fact common practice since we want to be able to control the characteristics of the resulting regressions, and the differences between the regressions are usually not objects of interest. However, there could be times when the underlying functions do not exhibit obvious structures or characteristics but their differences do. We give two examples of such functions. Consider two pairs of functions $f_1(\cdot)$ and $f_2(\cdot)$, and $f_3(\cdot)$ and $f_4(\cdot)$ given by

$$
\begin{aligned}
f_1(x) &= 0.4\cos(2\pi x) + 0.4\cos(3\pi x - 1)\\
f_2(x) &= 0.4\cos(2\pi x) + 0.4\cos(3\pi x - 1) + 0.2\sin(7\pi x) + 0.5\\
f_3(x) &= \sin(2\pi x) + \sin(3\pi x)\\
f_4(x) &= \sin(2\pi x) + \sin(3\pi x) + 0.5x\sin(8\pi x) + 1
\end{aligned}
$$

The functions are illustrated in Figures 4.2. As shown in plots, the functions do not display obvious characteristics such as linearity, trend, and periodicity that we can encode in covariance functions. However, their differences do and we can use covariance functions to explicitly express such characteristics. The difference of the pair $f_1(\cdot)$ and $f_2(\cdot)$ can be modelled by a periodic covariance function. The difference of the pair $f_3(\cdot)$ and $f_4(\cdot)$ can be modelled by the product of a linear covariance function and a periodic covariance function. If we wish to model the differences directly with certain covariance function, we only need to make a simple modification to Algorithm 1. Instead of using the sum of the covariance

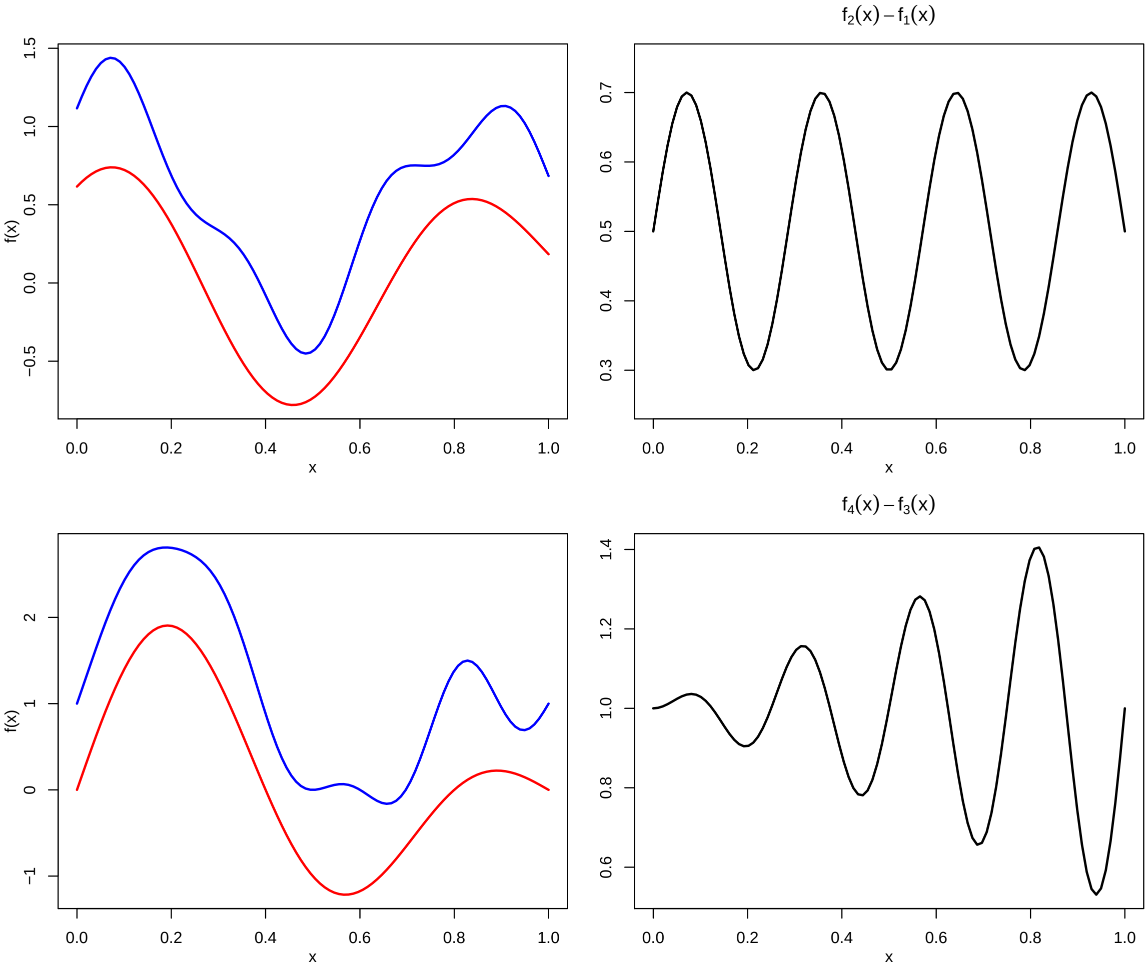


Figure 4.2: Examples of functions that don't exhibit obvious structures but their differences do. The left column shows two pairs of functions and the right column shows their respective differences. The first row depicts the functions f_1 and f_2 given in Section 4.3.5 in red and blue respectively, as well as their difference in black. The second row depicts the functions f_3 and f_4 given in Section 4.3.5 in red and blue respectively, as well as their difference in black. The functions are obtained by discretizing the x-axis into 100 equally spaced points.

functions of the regressions to construct the matrix Γ in step 6, use the covariance function that one wishes to model the differences with. One disadvantage of this approach is that the resulting regressions become difficult to interpret.

4.4 Conclusion

In this chapter we have developed a methodology for performing Gaussian process regression in the case where we have multiple processes with a non-colliding constraint. We have seen that the actual implementation of the methodology requires the simulation of truncated multivariate Gaussian random variables. In the next chapter, we take a small detour to consider some recent results on how to achieve this.

Chapter 5

Simulating Truncated Multivariate Gaussian Distribution

Developing numerical techniques for sampling from the truncated multivariate Gaussian distribution is an active area of research as it is needed for fitting many statistical models. For example, constrained Bayesian linear regression (Rodriguez-Yam et al., 2004), censored models (Tan et al., 2002), order restricted models (Robertson, 1988), and truncated multivariate probit models in market research (Liechty et al., 2001). In the previous chapter, we showed that implementing Gaussian process regression with non-colliding or inequality constraints also requires sampling from the truncated multivariate Gaussian distribution.

We recall that a n-dimensional truncated multivariate Gaussian distribution $\mathcal{TN}(\mu, \Sigma, \mathbf{l}, \mathbf{u})$ is derived from a multivariate normal distribution with mean $\mu \in \mathbb{R}^n$, covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ that has been constrained so that samples have lower bound $\mathbf{l} \in \mathbb{R}^n$ and upper bound $\mathbf{u} \in \mathbb{R}^n$. Of course, this means that the naïve way to sample from this distribution is to draw from the appropriate multivariate normal and then to only accept a sample x if it satisfies $\mathbf{l} \leq x \leq \mathbf{u}$. This is of course terribly inefficient.

In this chapter, we give a brief introduction to three recent and more computationally efficient methods: rejection sampling from the mode (RSM), Hamiltonian Monte Carlo (HMC), and minimax exponential tilting (MET). We finish the chapter by comparing the computational cost of the three methods on an example problem.

5.1 Rejection Sampling

RSM is an extension of the classic rejection sampling approach. We recall that the general approach to rejection sampling is as follows. Suppose f and g are two probability density functions such that $f(x) < cg(x)$ for all x in the support of f, where $c \geq 1$. Then the random sample X resulting from Algorithm 3 follows the distribution described by f.

```
Algorithm 3: Rejection sampling
Generate $X$ with density $g$.
Generate $U$ from a uniform distribution on $[0,1]$.
If $cg(X)U \leq f(X)$, accept X; otherwise, go back to step 1.
```

It can be shown that the expected number of draws before acceptance of the sample is equal to $\frac{1}{c}$ and this means this approach is slow when sampling from truncated multivariate Gaussian distribution due to the low acceptance rate (Maatouk and Bay, 2016). In Maatouk and Bay (2016), they propose the RSM approach that is specifically tailored for sampling from truncated multivariate Gaussian distribution and demonstrate noticeable performance gains over rejection sampling. We now describe their approach.

Consider the case where we wish to sample from a Gaussian distribution in a convex subset C of $\mathbb{R}^p$. Let f be the probability density function of the Gaussian distribution with mean μ and covariance matrix Σ. Assume that $\mu \notin C$, and without loss of generality, let $\mu = 0$. We can determine the mode μ^* of f restricted to C by solving

$$\mu^* = \arg\min_{x \in C} \frac{1}{2} x^T \Sigma^{-1} x.$$

Let g be the the probability density function of the Gaussian distribution centred at the mode μ^* with the same covariance matrix Σ as f. Define two truncated Gaussian distributions based on f and g, given by

$$\widetilde{f}(x) = f(x)\mathbb{1}_{x \in C}, \qquad \widetilde{g}(x) = g(x)\mathbb{1}_{x \in C}.$$

Then the RSM algorithm to obtain samples from the truncated multivariate Gaussian distribution $\widetilde{f}(x)$ is given in Algorithm 4.

```
Algorithm 4: RSM Algorithm
1 Generate $X$ with density $\widetilde{g}$
2 Generate $U$ from a uniform distribution on $[0,1]$
3 If $U \leq \exp[(\mu^*)^T\Sigma^{-1}\mu^* - X^T\Sigma^{-1}\mu^*]$, accept $X$; otherwise go back to step 1
```

In practice, we note that in line 1 of Algorithm 4, rejection sampling is used to generate X from density $\widetilde{g}$. As such, RSM can be viewed as a generalisation of rejection sampling and is equivalent to rejection sampling if $\mu \in C$ as we have $\mu = \mu^*$ so $f = g$ and $\widetilde{f} = \widetilde{g}$. Through an extensive simulation study, Maatouk and Bay (2016) have shown noticeable improvement in performance over rejection sampling and the acceptance rate does not decrease as rapidly as rejection sampling when C becomes smaller or the dimension becomes larger.

5.2 Hamiltonian Monte Carlo

Duane et al. (1987) introduced a hybrid Monte Carlo method which was initially intended for computer simulations in lattice field theory. The method utilises the properties of Hamiltonian dynamics and combines them with Gibbs sampling and an elaborated Metropolis algorithm. Duane et al.'s method was later extended to statistical applications by Neal (2012) and was called Hamiltonian Monte Carlo (HMC). The major benefit of HMC is the avoidance of random walks. HMC introduces a momentum variable allowing it to move with larger steps. HMC trajectories tend to move in the same direction until they enter a region of low probability, after which they are "reflected" (Neal, 2012). These properties permit more efficient exploration of the sample space than random walks and give less correlated samples. On the other hand, sampling methods that produce random walks, such as Gibbs and Metropolis, have no tendency to move in the same direction and have smaller step size, resulting in slower convergence to the target distribution. HMC has also been applied to sampling from the truncated multivariate Gaussian distribution. The algorithm presented in

Pakman and Paninski (2014) will be used to compare with other methods.

5.3 Minimax Exponential Tilting

The last method that we consider is the minimax exponential tilting (MET) method proposed by Botev (2017). Exponential tilting is a technique to transform distributions and is commonly used in rare-event simulation. When sampling from a truncated multivariate Gaussian distribution, the acceptance rate of rejection sampling becomes a rare-event probability as the dimensions increases. This has been shown in numerous experiments (see for example Maatouk and Bay 2016) and is the reason why rejection sampling is slow and inefficient. Exponential tilting is able to mitigate this problem by supplying a family of distributions that can be used as the proposal distributions in rejection sampling. Botev (2017) provides a method to find the optimal tilting parameter by solving a minimax optimisation problem. This approach shows substantial improvement in acceptance rate over rejection sampling.

5.4 Other Methods

We note that there are other methods such as Gibbs sampling (e.g. Damien and Walker 2001) and the separation of variables method (Genz, 1992). We decide not to include Gibbs sampling because in Pakman and Paninski (2014), the authors demonstrated that HMC has better effective sample size over time performance than the Gibbs sampler given in (Damien and Walker, 2001). Similarly, the separation of variables method given in Genz (1992) is not included because Botev (2017) showed that MET improves on it in terms of accuracy (especially in the tails of the distribution) and computational cost.

5.5 Computational Comparison

In this section, we will compare the computational performance of the three methods: RSM, HMC and MET. The methods will be tested on different dimensions p of the multivariate

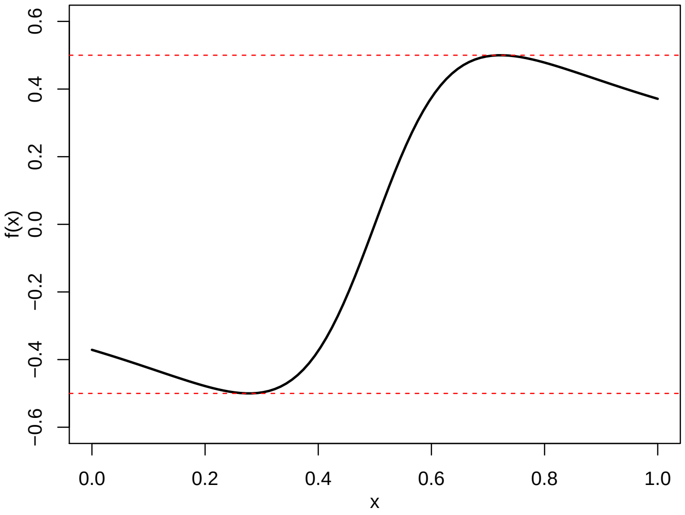


Figure 5.1: The function used in the experiment to compare the sampling methods is depicted. The function is given in Section 5.5. The red dotted lines indicate the lower and upper bound of the function which is $\frac{-1}{2}$ and $\frac{1}{2}$ respectively.

Gaussian distribution so we can see how increasing the number of knots in our model would affect the performance. We will run the algorithms to obtain 10,000 samples, and for each p, every method will run 20 times.

The performance of the methods will be evaluated by training a Gaussian process regression with inequality constraints using the function

$$f(x) = \frac{4.5(x-0.5)}{1+[4.5(x-0.5)]^2}.$$

As shown in Figure 5.1, the function is bounded by $-\frac{1}{2} \leq f(x) \leq \frac{1}{2}$. We could consider this as an example function used to generate training data in our model. The number of knots will be chosen depending on the number of dimensions of the multivariate truncated Gaussian distribution we want to simulate. The training points and the knots share the same locations and are equally spaced on $[0,1]$. The posterior mean and covariance matrix will be computed and be used as the parameters of the distribution that we will simulate. We will use the squared exponential covariance function with length scale parameter 0.2. The truncation is at the lower and upper bound of the function, which is $-\frac{1}{2}$ and $\frac{1}{2}$ respectively. We test the methods on the dimensions (number of knots) 25, 50, 75, 100.

To evaluate the performance of the methods, we will record the time taken in seconds and compute the effective sample size. Effective sample size gives us an idea on how many samples are drawn independently and is defined as

$$\text{ESS} = \frac{n}{1+2\sum_{i=1}^{n}\rho_i},$$

where n is the number of samples and ρ_i is the autocorrelation of lag i. If there is no autocorrelation, implying that the samples are independent, then $\rho_i = 0$ for all i and the ESS equals to n.

ESS is only defined for one dimension. Here, we consider dimensions greater than one so we will evaluate ESS on each dimension, then take the minimum as well as the median of the ESS's, which will be denoted by Min-ESS and Med-ESS respectively. However, ESS does not take into account the cross-correlations between the components of the multivariate distribution so we will also consider a multivariate approach to ESS proposed by Vats et al. (2019), which will be denoted by Multi-ESS. Let $\{Y_1, Y_2, \ldots, Y_n\}$ be a sequence of n draws from a sampling method. Then Multi-ESS is given by

$$\text{Multi-ESS} = n\left(\frac{|\Lambda|}{|\Sigma|}\right)^{\frac{1}{p}},$$

where $|\cdot|$ denotes determinant, Λ is the sample covariance matrix and Σ is the multivariate batch means variance estimator

$$\Sigma = \frac{b}{a-1}\sum_{k=0}^{a-1}(\bar{Y}_k - \hat{\theta})(\bar{Y}_k - \hat{\theta})^T,$$

where a is the number of batches, b is the batch size, $\bar{Y}_k$ is the mean vector of batch k and $\hat{\theta}$ is the sample mean

$$\hat{\theta} = \frac{1}{n}\sum_{t=1}^{n} Y_t.$$

Choosing the optimal batch size for the estimator Σ is an ongoing open research problem. Flegal et al. (2010) showed that an asymptotically optimal batch size in terms of MSE is proportional to $n^{1/3}$ but the proportionality constant is unknown. In our experiment we will simply choose the floor of $n^{1/3}$ to be the batch size.

During our experiment, we find that RSM is significantly slower than the other two methods. When $p = 25$, RSM takes 1107s to obtain 10,000 samples as opposed to HMC's 0.25s and MET's 0.14s. As higher dimensions would reduce the acceptance rate further, resulting in even longer runtime, we decide not to proceed with RSM in the test of higher dimensions. Therefore, the results presented do not include RSM. The different measures of ESS over time as well as the time taken in seconds are presented in Figure 5.2. As shown, MET has better ESS over time performance than HMC in all dimensions. HMC's computation time is close to MET's initially but as the number of dimensions grows, it increases exponentially. On the other hand, MET's computation time grows much slower, scaling well into higher dimensions. From our experiment, MET is the best method out of the three for sampling from truncated multivariate Gaussian distributions.

5.6 Conclusion

In this chapter, we compared the performance of three methods of simulating truncated multivariate Gaussian distribution in terms of effective sample size over time. From our experiment, we found that MET performed the best out of the three. In the next chapter, we will perform a simulation study to demonstrate the non-colliding model in various settings. When there is a need to simulate the truncated multivariate Gaussian distribution, we will use MET to perform this task.

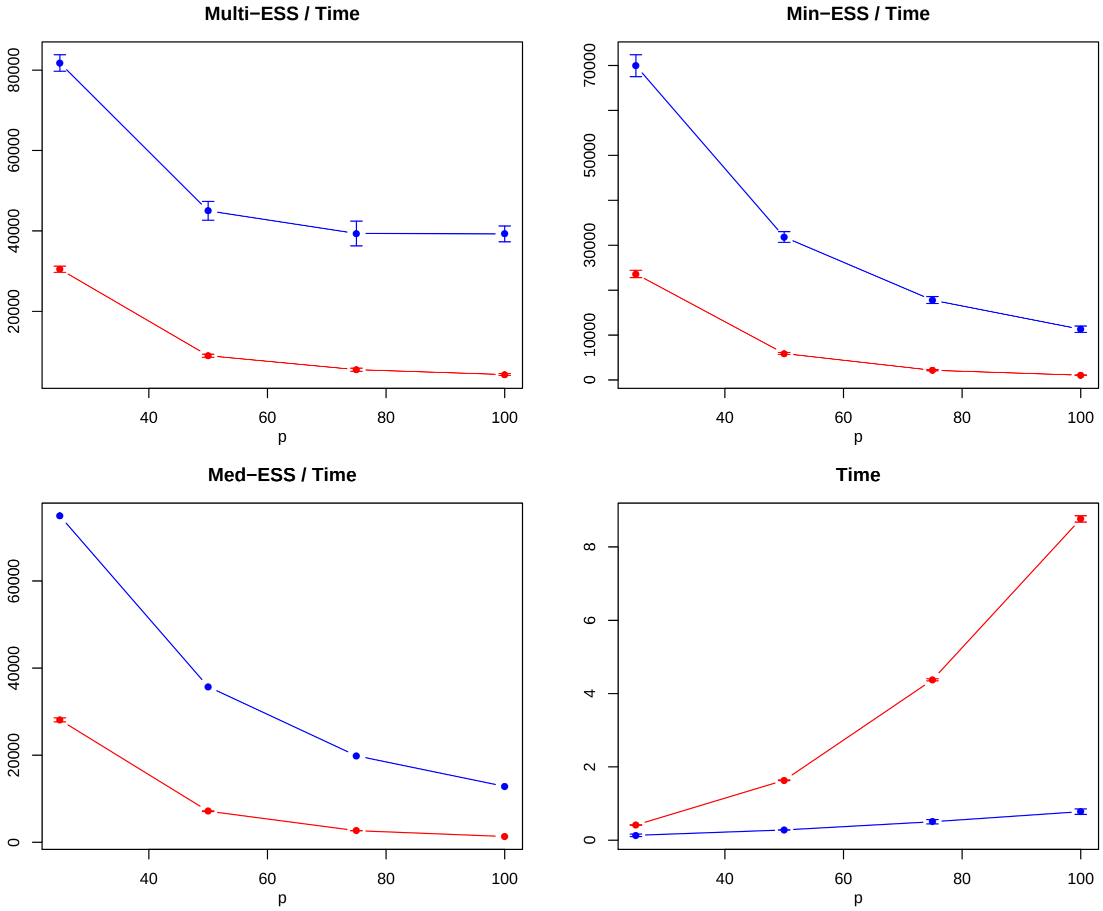


Figure 5.2: Red indicates the results from HMC, whereas blue represents MET. The results for $p = 25, 50, 75, 100$ averaged over 20 runs as well as one standard deviation from the mean are shown for various metrics. We have omitted the RSM approach due to its extensive runtime.

Chapter 6

Simulation

In this chapter, we conduct a number of simulation experiments to demonstrate the non-colliding Gaussian process regressions as well as its performance relative to the non-constrained model. We first consider the model with one-dimensional input in Section 6.1. Noise-free observations, noisy observations, as well as local non-colliding constraint are explored. Then in Section 6.2, an example of two-dimensional input with noisy observations is shown. From our simulations, the non-colliding model demonstrates better performance and is able to model the relationship between the functions better, especially when the training set is small. Small training set results in greater uncertainty and thus greater model variability and higher likelihood of modelled regressions colliding. In such cases, the non-colliding constraint restricts the model limiting its variability and provides additional information to guide the model to the correct path. Lastly, in Section 6.3, we discuss some limitations of the non-colliding model as well as when it is the appropriate choice over the non-constrained model.

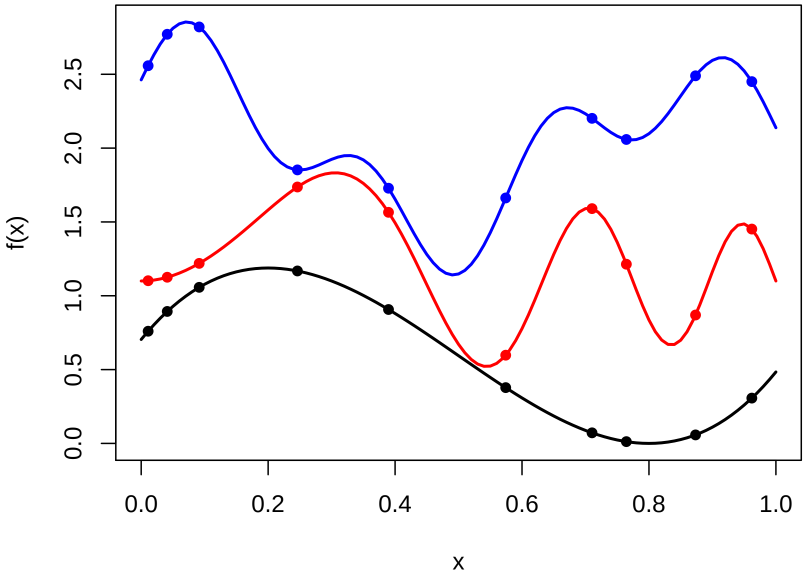


Figure 6.1: The three functions that are used to generate the training data are shown by solid lines in different colours. Ten noiseless training points generated by each function are indicated by solid dots in corresponding colours. The functions were obtained by discretizing the x-axis into 100 equally spaced points. The black line is $f_1(\cdot)$, red line is $f_2(\cdot)$ and blue line is $f_3(\cdot)$. Also shown are the training points at ten random locations (denoted by points).

6.1 One-dimensional Input

Let us first consider a one-dimensional example, where we have three non-colliding functions given by

$$\begin{aligned} f_1(x) &= 11(x+0.1)(x-0.8)^2, \\ f_2(x) &= \frac{\sin(5\pi x^2)}{\exp(x)} + 1.1, \\ f_3(x) &= 0.3[\cos(2\pi x) + \cos(3\pi x - 1) + \sin(7\pi x)] + 2. \end{aligned}$$

These functions are plotted in Figure 6.1. Training points at ten random locations are generated from the three functions giving the 10-dimensional vectors y_1 from $f_1(\cdot)$, y_2 from $f_2(\cdot)$, and y_3 from $f_3(\cdot)$. We trained the Gaussian process regressions with and without the non-colliding constraint on these training points. We note that our approximation method depends on the number of knots m. The larger the m, the closer the approximation gets to

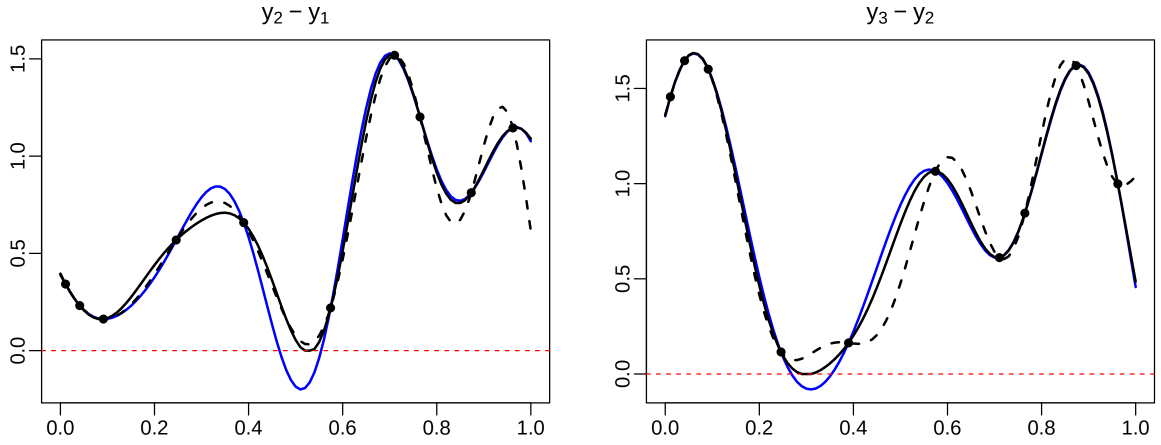


Figure 6.2: Differences between the observed y_i's are shown as black dots. The differences modelled by the non-colliding model are represented by black solid lines. The differences between non-constrained Gaussian process regressions are represented by blue solid lines. The black dotted lines are the true differences between the underlying functions. The regressions are obtained by discretizing the x-axis into 100 equally spaced points.

the true process it approximates. Since we are mainly interested in the difference between constrained and non-constrained Gaussian processes and not in the approximation error (at this point in time), we will use the same approximation method for the non-constrained Gaussian process regressions (instead of the exact Gaussian process regression described in Chapter 2) with the same number of knots as the constrained ones.

One hundred equally spaced knots between 0 and 1 are used to generate the Gaussian process regressions with and without the non-colliding constraint. The squared exponential covariance function is used and its parameters are optimised by maximising the likelihood function. The non-constrained model is trained on the observations directly. On the other hand, the non-colliding model is trained on the differences between the observations from different functions and are conditioned to not cross zero, as shown in Algorithm 1. The resulting regressions are shown in Figure 6.3. The differences between regressions, as well as the differences between the underlying functions are shown in Figure 6.2.

As we can see from Figure 6.2 and Figure 6.3, without the non-colliding constraint, the Gaussian process regressions intersect with each other in four places. If we have prior knowledge that the functions do not collide, then the Gaussian process regressions without

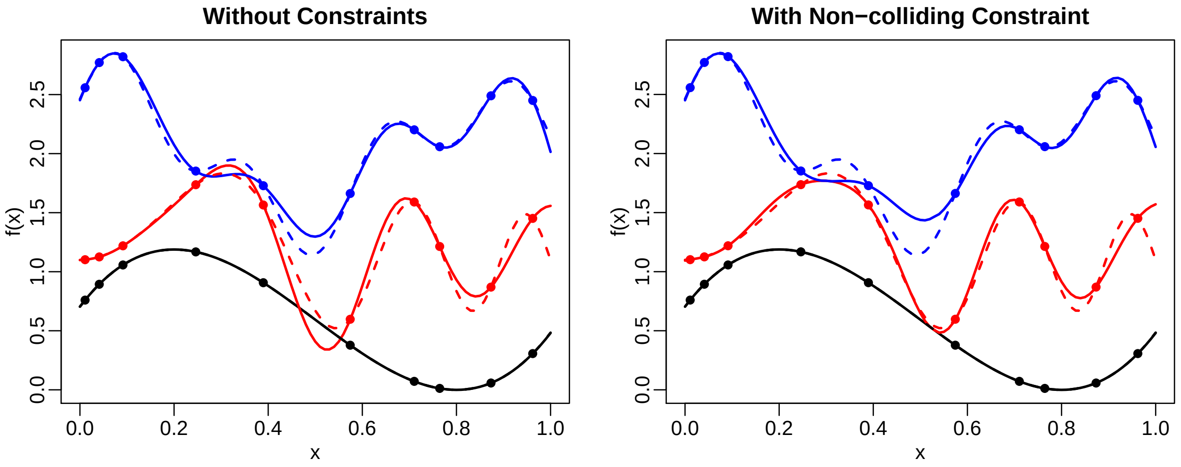


Figure 6.3: Gaussian process regressions with and without non-colliding constraint are shown. Solid lines are the posterior modes of the Gaussian process regressions. Dotted lines are the true functions used to generate the training observations which are represented by solid dots. The constrained and non-constrained model are obtained by discretizing the x-axis into 100 equally spaced points.

constraints do not reflect this prior information, resulting in the loss of valuable information and less realistic models. On the other hand, by imposing the non-colliding constraint, the resulting Gaussian process regressions are able to reflect the prior information and model the interplay between the functions more closely to the truth. The non-colliding condition is more likely to be violated in regions with greater uncertainty due to lack of observations. With more training points, Gaussian process regressions would be able to get closer to the underlying functions, and would eventually reflect the non-colliding nature of the underlying functions since the observations are noise-free. This may not be the case when the observations are noisy. We will see in Section 6.1.1 that some noisy observations may not obey the non-colliding nature of the underlying functions. Therefore, without imposing the non-colliding constraint, it is possible that Gaussian process regression would never reflect the non-colliding property of the true functions.

The fit of the models depend on the number of training points and their locations. We conduct an experiment to compare the average performance between the two methods with different number of training points and locations. We consider the training set size of 5, 10, 15 and 20. Training points are randomly chosen from a uniform distribution on $[0, 1]$ and 50 trials are conducted for each size. Root mean square error (RMSE) and Mean Absolute

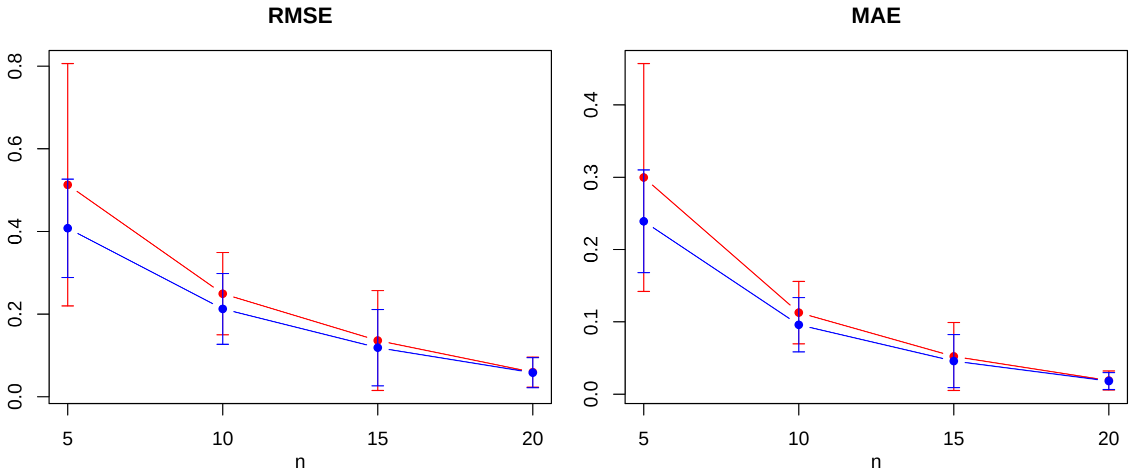


Figure 6.4: Average RMSE and MAE, as well as one standard deviation from the mean are shown for different numbers of training points. Red represents the errors from Gaussian process regressions without constraints and blue represents with non-colliding constraint. The experiment uses the functions from Figure 6.1, and for each training set size, 50 trials are done to obtain the mean and standard deviation. The models are tested on 100 equally spaced points.

Error (MAE) are used to evaluate and compare the fit of the models. They are given by

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \qquad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|,$$

where n is the number of observations in the test set, $\hat{y}$ is the predicted value and y is the true value. The test set is consisted of 100 equally spaced points from 0 to 1. We expect to see the constrained models to perform better when the size of the training set is small, but the difference would become smaller as the number of training points increases because more points would reveal the non-colliding nature of the underlying functions. We also expect the non-colliding models to have smaller variance than the non-constrained models when the size of the training set is small as the constraint limits the variability of the models. The results from our experiment are shown in Figure 6.4. The results are as expected. The constrained model has smaller average errors when the training set is small, but this difference quickly diminishes as we increase the number of training points. Non-constrained model has greater variability but the variances also decrease as the size of the training set increases and are eventually on the same level as the constrained model.

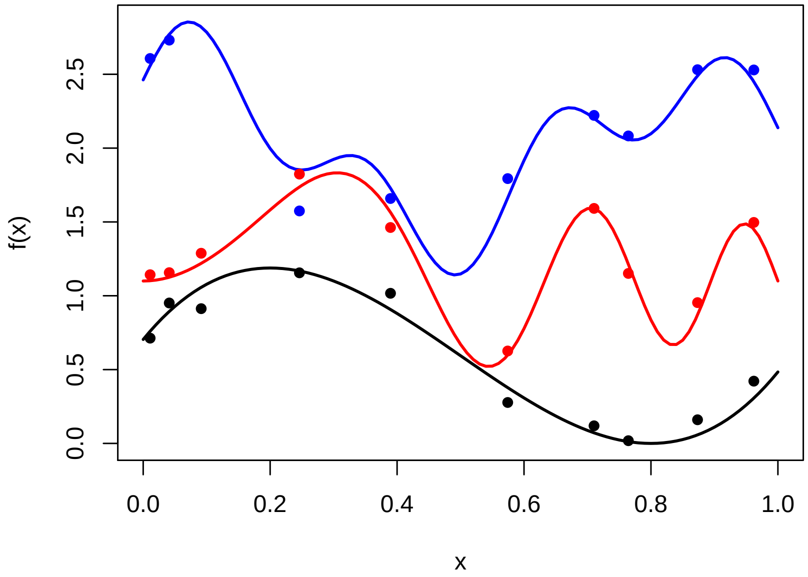


Figure 6.5: The three functions that are used to generate the training data are shown by solid lines in different colours. Ten noisy training points generated by each function are indicated by solid dots in corresponding colours. The functions were obtained by discretizing the x-axis into 100 equally spaced points.

6.1.1 Noisy Observations

We now consider the case where we have noisy observations instead of noise-free ones. We use the same functions as before but this time when generating the training data we add a Gaussian noise term to the y_i's with mean 0 and variance 0.01. The generated training data is shown in Figure 6.5. We can see that the data points no longer sit on the underlying functions due to the noise. One set of observations around 0.2 actually no longer obeys the non-colliding nature of the underlying functions, as the observation from $f_2(\cdot)$ is greater than the one from $f_3(\cdot)$. This would make it harder for the Gaussian process regressions without the non-colliding constraint to not collide with each other.

The same number of knots is used as before to train the regressions and the resulting regressions as well as their differences are shown in Figure 6.6 and Figure 6.7. In the noise-free case, despite intersecting with other regressions at a number of locations, the

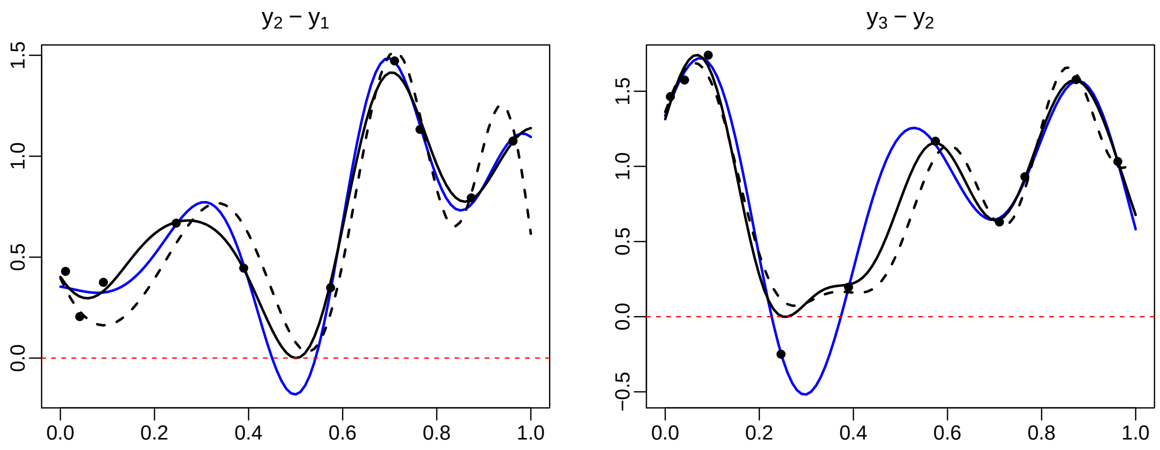


Figure 6.6: Differences between the noisy observations y_i's are shown as black dots. The differences modelled by the non-colliding model are represented by black solid lines. The differences between non-constrained Gaussian process regressions are represented by blue solid lines. The black dotted lines are the true differences between the underlying functions. The regressions are obtained by discretizing the x-axis into 100 equally spaced points.

Gaussian process regressions without the non-colliding constraint are still very close to the underlying functions. However, in the case of noisy observations, without the non-colliding constraint stopping the Gaussian processes from fitting to the observations with high level of noise, Gaussian process regressions tend to overfit as shown in our simulation. As we can see in Figure 6.6 and Figure 6.7, there is a location where y_2 is greater than y_1 indicating high level of noise. As the $y_3 - y_2$ at that location is below 0, the non-colliding constraint stops the Gaussian processes from fitting to that point. By doing so, the differences between the processes are closer to the true differences as shown in the $y_3 - y_2$ plot in Figure 6.6. The resulting regressions do not collide and are closer to the true functions than the regressions without the non-colliding constraints which fit closely to the noisy observations.

The same experiment is conducted to test the average performance of the models. The results, which are shown in Figure 6.8, are consistent with what we observe in the noise-free case. The constrained model has lower average error and smaller variance but the difference in performance between constrained and non-constrained model becomes very small when the size of the training set gets large. We also observe increased variations for both models as a result of noisy data.

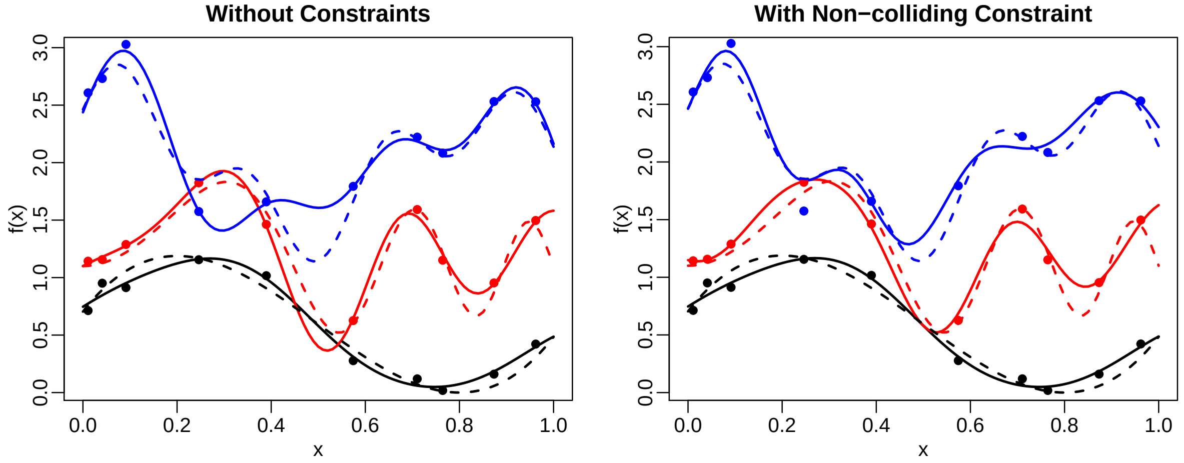


Figure 6.7: Gaussian process regressions with and without non-colliding constraint are shown. Solid lines are the posterior modes of Gaussian process regressions. Dotted lines are the true functions used to generate the training observations represented by solid dots. The Gaussian process regressions with and without non-colliding constraint were obtained by discretizing the x-axis into 100 equally spaced points.

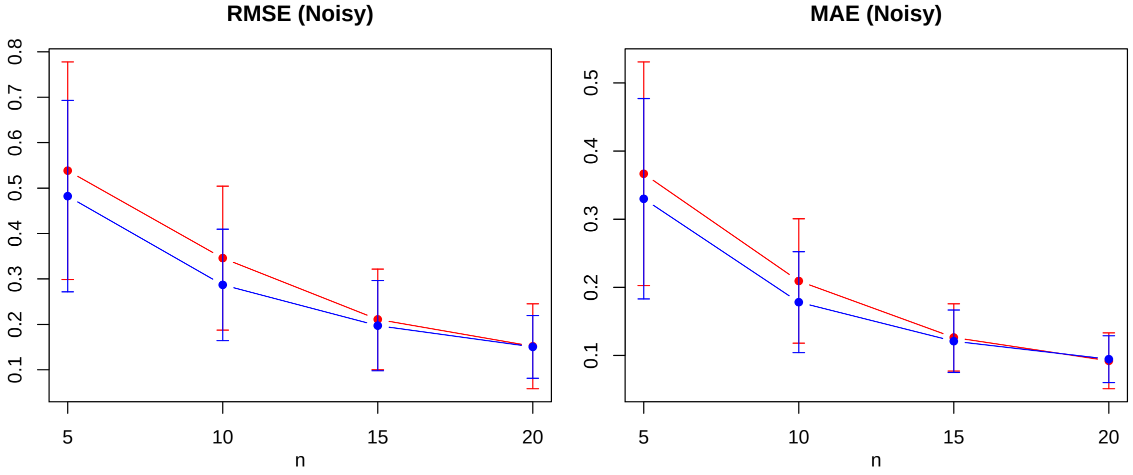


Figure 6.8: Average RMSE and MAE of noisy observations with noise variance 0.01, as well as one standard deviation from the mean are shown for different numbers of training points. Red represents the errors from Gaussian process regressions without constraints and blue represents with non-colliding constraint. The experiment uses the functions from Figure 6.1, and for each training set size, 50 trials are done to obtain the mean and standard deviation. The models are tested on 100 equally spaced points.

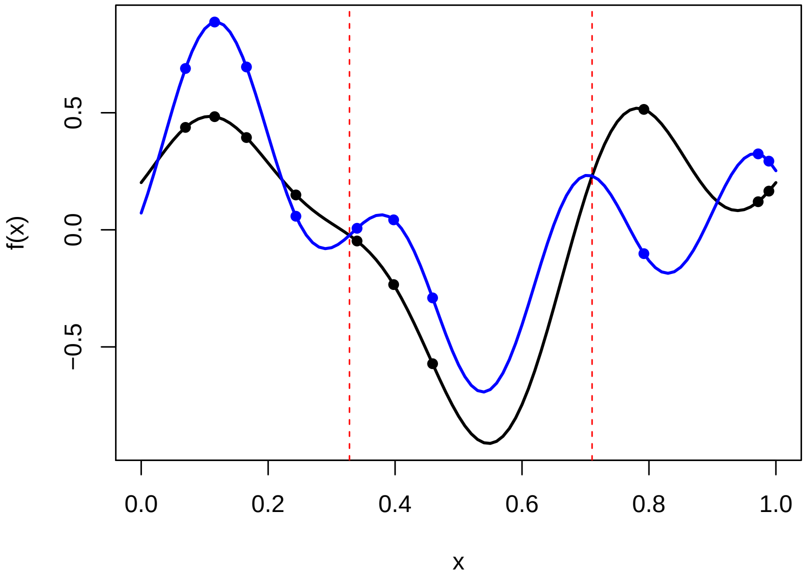


Figure 6.9: The two functions that are used to generate the training data for local non-colliding Gaussian processes are shown by solid lines in different colours. Noise-free training points generated by each function at ten random locations are indicated by solid dots in corresponding colours. The red dotted lines indicate the region of local non-collision where the non-colliding constraint is applied. The functions are obtained by discretizing the x-axis into 100 equally spaced points.

6.1.2 Local Non-colliding Constraint

The local non-colliding constraint is a special case of non-colliding constraint where the non-colliding constraint is only imposed on a subset of the domain. The two functions used for this simulation are shown in Figure 6.9 and defined by

$$f_1(x) = 0.5\sin(2\pi x + 1.5) + 0.3\cos(4\pi x + 3) + 0.2\sin(6\pi x),$$

$$f_2(x) = 0.3\cos(2\pi x - 1) + 0.3\cos(3\pi x - 1) + 0.3\sin(7\pi x - 1).$$

The black line in Figure 6.9 is $f_1(\cdot)$ and the blue line is $f_2(\cdot)$. The red dotted lines indicate the region where the local non-colliding constraint is imposed. Noise-free training points are generated from each function at ten random locations. Again, we use 100 equally spaced knots between 0 and 1 and the squared exponential covariance function for Gaussian process regressions. The parameters of the covariancce function are set by maximising the likelihood

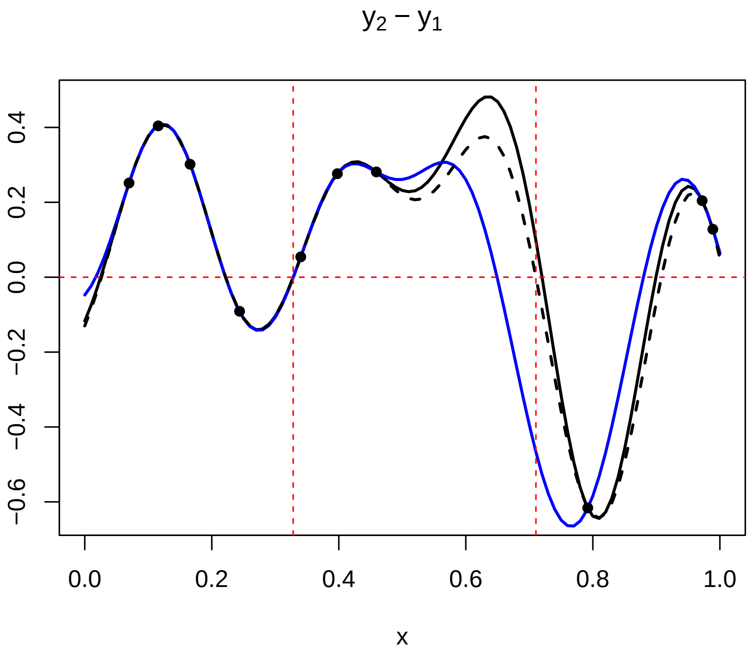


Figure 6.10: Differences between the observed y_i's are shown as black dots. The differences modelled by the non-colliding model are represented by black solid lines. The differences between non-constrained Gaussian process regressions are represented by blue solid lines. The black dotted lines are the true differences between the underlying functions. The vertical red dotted lines indicate the region of local non-collision where the non-colliding constraint is enforced. The regressions are obtained by discretizing the x-axis into 100 equally spaced points.

function. The resulting regressions as well as their differences are shown in Figure 6.10 and Figure 6.11.

The local non-colliding constraint forces the difference between the regressions to not cross 0 in the specified region. We can see how this is achieved in the constrained case and how it is violated in the non-constrained case in Figure 6.10. In the non-constrained case, the regressions intersect within the region of non-collision which also affects the shape of the regression outside the region. As a result, not only is the regression for $f_2(\cdot)$ inaccurately models the underlying function within the region but predictions outside the region would also be negatively affected. On the other hand, the regressions with the local non-colliding constraint are close to the shape of the underlying functions.

We again conduct the same experiment to test the performance of the constrained and non-constrained models with different sets and numbers of training points. As shown in

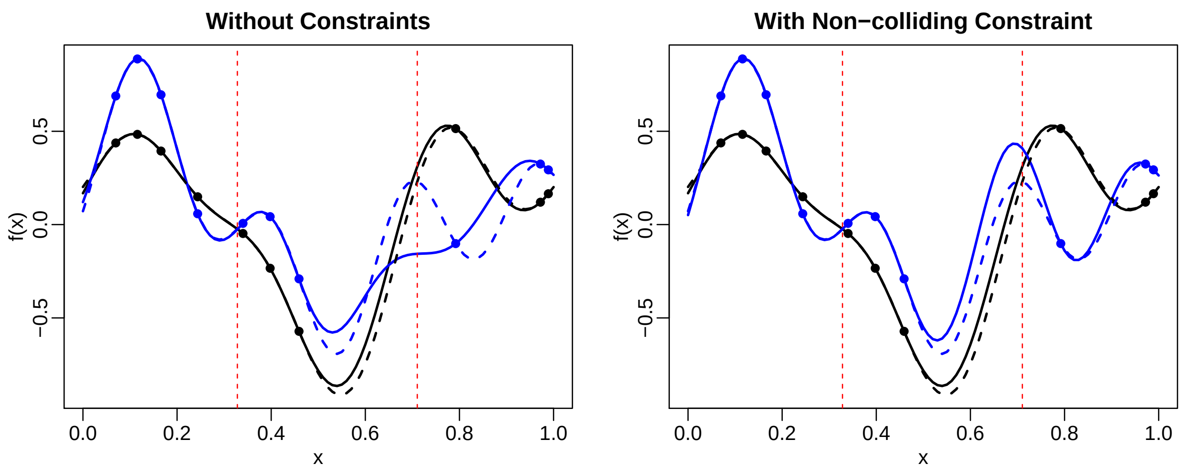


Figure 6.11: Gaussian process regressions with and without local non-colliding constraint are shown. Solid lines are the posterior modes of Gaussian process regressions. Dotted lines are the true functions used to generate the training observations represented by solid dots. The vertical red dotted lines indicate the region of local non-collision where the non-colliding constraint is enforced. The Gaussian process regressions with and without non-colliding constraint were obtained by discretizing the x-axis into 100 equally spaced points.

Figure 6.12, the constrained model still shows lower average errors when the training set is small but this time the variances of the two models are comparable. This could be due to the fact that the non-colliding constraint is only imposed on a subset of the domain. Therefore, the variability of the model is not limited as much as the case where the constraint is enforced on the entire domain.

6.2 Two-dimensional Input

We have seen that non-colliding Gaussian process regressions work quite well in the one-dimensional input case. As discussed in Section 4.2.1, this method can be extended to higher dimensional input spaces. However, due to the difficulty in visualising higher dimensions as well as the computational burden that comes with it, we will only consider the two-dimensional input space. We consider the two functions:

$$\begin{aligned} f_1(x_1, x_2) &= \frac{1}{2}[\sin(3\pi x_1) + \sin(3\pi x_2)], \\ f_2(x_1, x_2) &= \frac{1}{2}\left[\cos(2\pi x_1 - 6) + \frac{1}{2}\sin(6\pi x_1 - 3) + \cos(2\pi x_2 - 6) + \frac{1}{2}\sin(6\pi x_2 - 3)\right] + 1. \end{aligned}$$

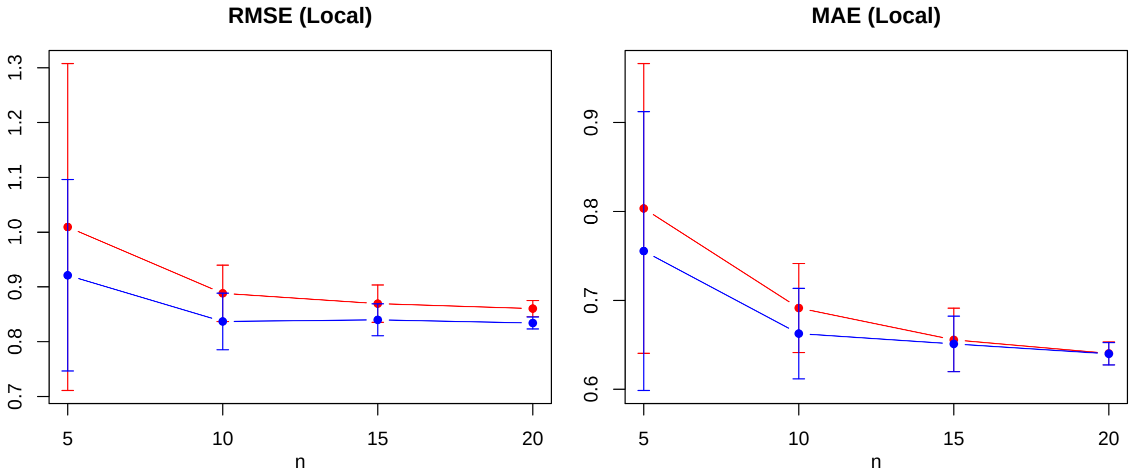


Figure 6.12: Average RMSE and MAE, as well as one standard deviation from the mean are shown for different numbers of training points. Red represents the errors from Gaussian process regressions without constraints and blue represents with local non-colliding constraint. The experiment uses the functions from Figure 6.9, and for each training set size, 50 trials are done to obtain the mean and standard deviation. The models are tested on 100 equally spaced points.

The surface plot of the two functions are shown in Figure 6.13 and their heat maps are shown in the first row of Figure 6.15. The surface plot and heat map of their difference $f_2 - f_1$ is given in Figure 6.14 and Figure 6.15 respectively. The two functions are non-colliding in the domain $[0, 1]^2$ by design. Noisy training points are generated from each of the two functions at thirty random locations. The noise variance is set to 0.01. The locations of the training points are generated by *maximin Latin hypercube sampling* that we shall now describe.

Latin hypercube sampling is a popular sampling method which is widely used in simulations and experimental designs. It has desirable properties such as space filling and good uniformity with respect to each dimension (Viana, 2016). Monte Carlo method requires large number of samples to approximate the distribution well and can be inefficient when samples are close together. On the other hand, Latin hypercube sampling has shown better performance than Monte Carlo method when the sample size is small (McKay et al., 1979). In our simulation, since we do not use large number of points and wish to cover the domain as much as possible with the number of points available in order to help the models learn all areas of the functions, we choose to use Latin hypercube sampling over Monte Carlo

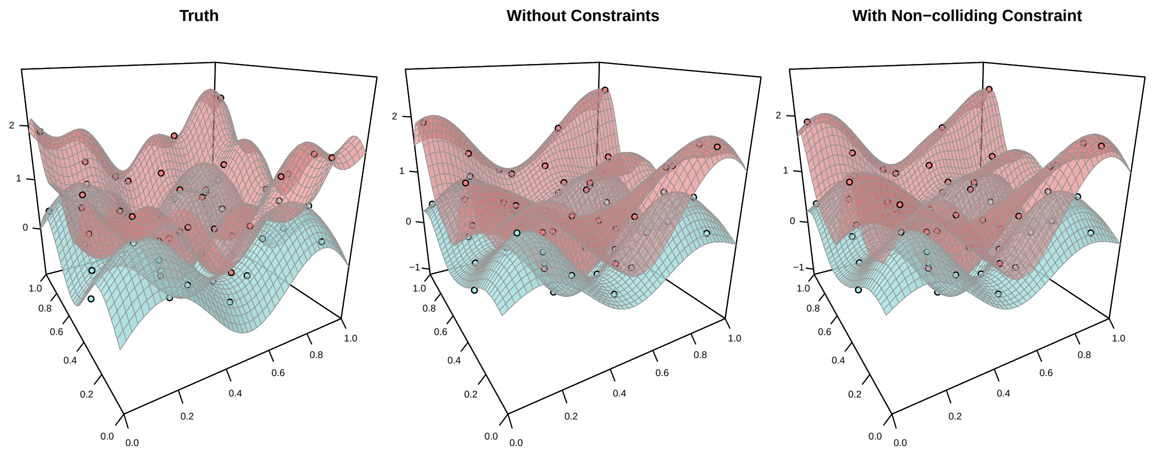


Figure 6.13: Left: The two functions that are used to generate the training data for two-dimensional non-colliding Gaussian processes. Thirty training points generated by each function are indicated by dots in corresponding colours. **Middle**: The surfaces produced by Gaussian process regressions without constraints. **Right**: The surfaces produced by non-colliding Gaussian process regressions. The surfaces are obtained by discretizing the domain into 50 × 50 equally spaced grid.

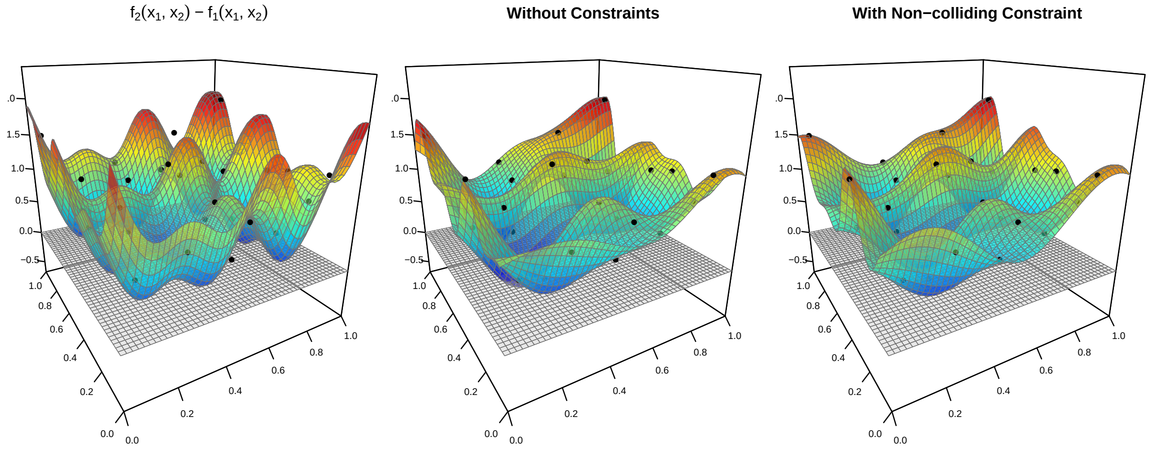


Figure 6.14: Left: The surface of the difference $f_2 - f_1$. The black dots indicate the difference between the training points generated by f_1 and f_2 at those locations. The grey flat surface represents the 0 bound that the surface should not cross if the functions are non-colliding. **Middle**: The difference between the two functions modelled by non-constrained Gaussian process regressions with squared exponential covariance functions. **Right**: The surface of the difference modelled by non-colliding Gaussian process regressions. The surfaces are obtained by discretizing the domain into 50 × 50 equally spaced grid.

sampling.

A Latin hypercube refers to the hypercube where each point in the hypercube is the only one in each axis-aligned hyperplane that contains it. In our case of two dimensional input, it means that there is only one point in each row and column. To achieve this, suppose we wish to sample n points in p dimensions, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$, where $\mathbf{x}_i = [x_1^{(1)}, x_i^{(2)}, \ldots, x_i^{(p)}]$, we divide the range of each dimension into n equal bins and only sample once from each bin. Then the components of $\mathbf{x}_i$'s are matched randomly.

In our simulation, we use a variant of the Latin hypercube method called maximin Latin hypercube (Johnson et al., 1990). The resulting samples also satisfy the structure of a Latin hypercube but the the distance between the points closest to each other is maximised. The algorithm begins with building a Latin hypercube by dividing the range of the dimensions into n equal bins just as before. Then a random starting point is chosen and the next point is chosen at the available locations in the Latin hypercube that has the maximum distance to the point closest to it. The algorithm proceeds by adding one point at a time until the n points are generated. The resulting design is a Latin hypercube with increased multidimensional uniformity (Deutsch and Deutsch, 2012).

Thirty locations are chosen randomly using maximin Latin hypercube sampling and training points are generated at these locations using the two functions. We use the squared exponential covariance function for the Gaussian processes and the parameters of the covariance function are optimised by maximising the likelihood function. The surface plot and the heat maps of the resulting constrained and non-constrained model are shown in Figure 6.13 and Figure 6.15. As before, the non-constrained model is trained on the observations directly, then we take the difference of the two non-constrained regressions to compare with the difference modelled by the constrained model in Figure 6.14 and Figure 6.15. As shown in Figure 6.15, without the non-colliding constraint, the Gaussian process regressions have three regions of collisions outlined in white in the third plot of the second row. We do observe some differences between the constrained and non-constrained model especially near the regions of collisions but the two models are still quite similar. The

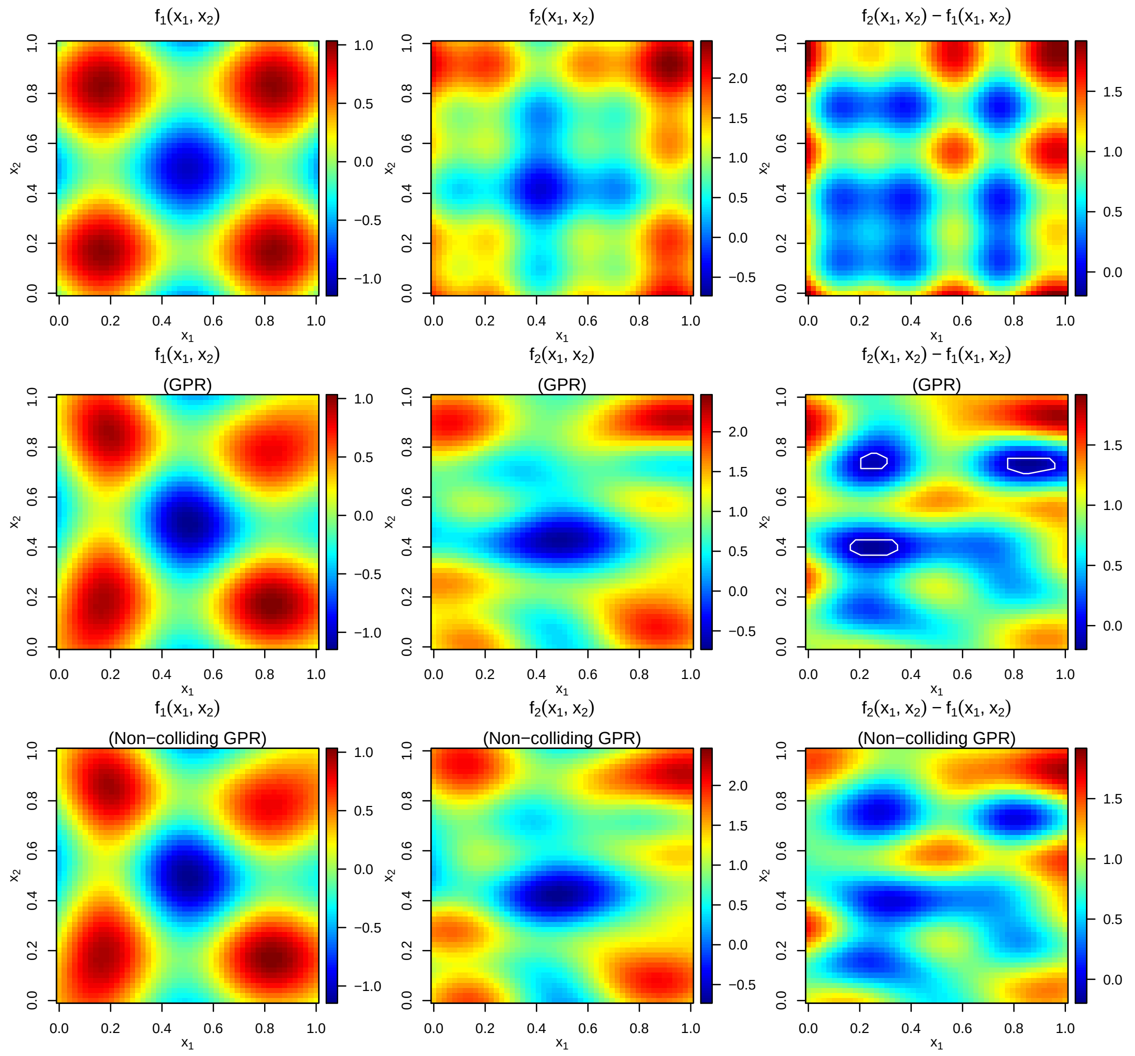


Figure 6.15: First row: The heat maps of f_1, f_2 and their difference $f_2 - f_1$. **Second row**: The heat maps of the surfaces produced by non-constrained Gaussian process regressions. The white outlines in the third plot shows the regions where the surface is below 0 indicating the regions of collision. **Third row**: The heat maps of the surfaces produced by non-colliding Gaussian process regressions.The heat maps are obtained by discretizing the domain into 50 × 50 equally spaced grid.

non-colliding model has slightly lower errors with 0.214 RMSE versus 0.219, and 0.149 MAE versus 0.154, tested on 50 × 50 equally spaced grid.

We now conduct an experiment to test how well the two models perform with different training points and sample size. The locations of the training points are again randomly chosen by maximin Latin hypercube sampling. The sample size we will investigate are 30, 35, 40 and 45. Due to computational constraint, we are only able to do 20 trials for each sample size. The models are tested on 50 × 50 equally spaced grid. The resulting average

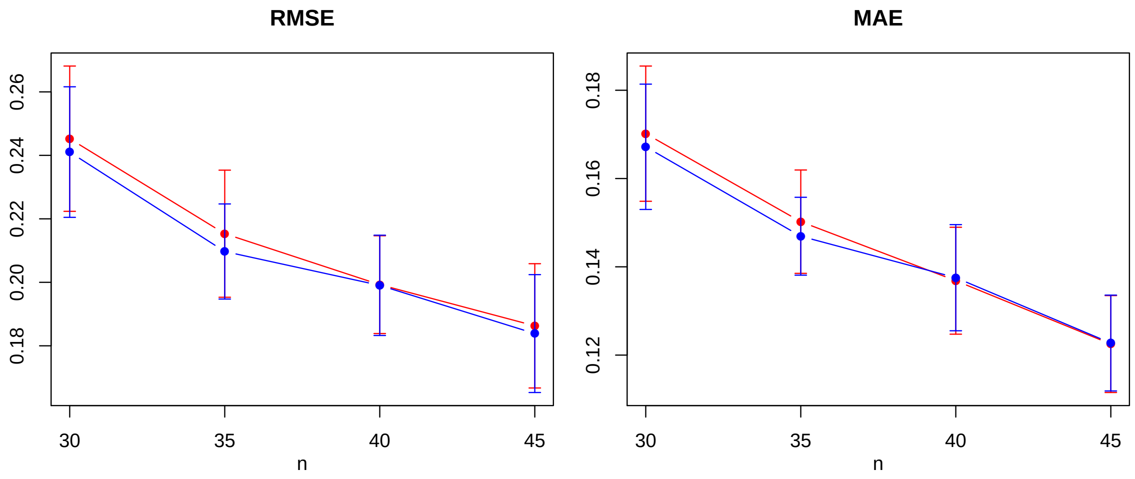


Figure 6.16: Average RMSE and MAE, as well as one standard deviation from the mean are shown for different numbers of training points. Red represents the errors from Gaussian process regressions without constraints and blue represents with non-colliding constraint. The experiment uses the functions from Figure 6.13 to generate the training data, and for each training set size, 20 trials are done to obtain the mean and standard deviation. The models are tested on a 50×50 equally spaced grid.

RMSEs and MAEs as well as one standard deviation away from the mean are shown in Figure 6.16. This time the performance of the two models are very close in terms of both average errors and the variability. Unlike the previous experiments, the variances of the two models do not decrease much when the size of the training set increases. The non-colliding model still has lower average errors and smaller variance when the training set is small but the difference between the two models are considerably smaller than before.

6.3 Discussion

In this chapter, we have shown a number of examples of non-colliding Gaussian process regressions in different settings. The constrained model has demonstrated better performance when the number of training points is not enough for the non-constrained model to not collide. When there is enough training observations, the two models have shown similar performance in our experiments. Given enough training observations, the non-constrained model does not need the constraint to satisfy the non-colliding condition in the noise-free case, and depending on the magnitude of the noise, some noisy cases as well. When this happens, the difference in performance between the two models would be from modelling the training observations directly versus modelling the difference of the training observations.

The examples we have shown in this chapter are all cases where the functions are close to each other, making it easier to have occurrences of collision when models do not follow the underlying functions closely. However, we may have cases where the functions are far apart and modelling the observations directly would have no chance of collisions. In such cases, using non-constrained Gaussian process regressions to model the observations directly may offer better performance than the non-colliding model. This is due to the non-colliding constraint no longer restricting the regressions in any way, thus rendering it useless. Furthermore, by modelling the difference then transforming it back would mean that the resulting regression would inherit errors from previous regressions and the errors from modelling the difference. Since the difference has greater variability, the errors are likely to be larger as well. Therefore, if the functions are far apart and the non-colliding constraint is not binding, then using the non-constrained Gaussian process regressions should offer better performance.

Chapter 7

Conclusion

7.1 Summary of Contributions

In this thesis, we proved an asymptotic result for non-colliding Ornstein-Uhlenbeck processes similar to the one given in Grabiner (1999) for non-colliding Brownian motions. Our result showed that the probability of n independent Ornstein-Uhlenbeck processes experiencing no collisions up to time T is asymptotic to a constant multiple of $e^{\frac{-\theta n(n-1)}{2}T}$ as $T \to \infty$, and the constant is a polynomial of the starting positions $\mathbf{x}$. Furthermore, we developed a way to impose non-colliding constraint on Gaussian process regressions by building on the piecewise linear approximation model by López-Lopera et al. (2018), which enables inequality constraints to hold in the entire domain. Our method involves transforming a sequence of Gaussian processes into their differences. Then, by utilising the property of Gaussian process that a linear combination of independent Gaussian processes is also a Gaussian process, we fit Gaussian process regressions on these differences with the constraint that they are bounded by zero. These differences are then transformed back to obtain the desired non-colliding regressions.

We demonstrated the non-colliding model through an extensive simulation study, covering noise-free data, noisy data, local non-collision, and two-dimensional input. Our simulation study showed that the constrained model is able to produce more realistic models that reflect the prior information of no collisions, as well as smaller errors with less variability especially when the training set is small. However, as expected, the benefit of introducing non-colliding constraint to the model diminishes as the training set gets larger. This is due to more of the non-colliding information is revealed in the data and eventually with enough data the non-constrained model would also be able to obey the non-colliding condition.

Furthermore, as the implementation of the proposed non-colliding model requires simulating the truncated multivariate Gaussian distribution, we compared three methods, namely RSM, HMC and MET. Our experiment shows their relative performance in terms of effective sample size over time as well as how well each method scales to higher dimensions. We found that MET performed the best out of the three and its computation time scales considerably better than the second best method HMC.

Lastly, we pointed out that there could be situations where the underlying functions are well separated, and just by fitting the non-constrained Gaussian process regressions would have no chance of intersecting. Then the non-colliding model would provide little to no value in feeding additional information to the model. Therefore, the difference between the two models would simply be modelling the training observations directly versus modelling their differences.

7.2 Future Work

In this thesis, we considered the situation where the underlying processes are independent. However, the non-colliding nature of the processes could sometimes be induced by correlations. In such cases, perhaps introducing cross-correlations between the processes may be sufficient to prevent them from colliding with each other. This is in fact an approach used when modelling multiple outputs simultaneously with Gaussian process regressions (see for example Seeger et al. 2005). It would be interesting to see in the case of dependent non-colliding processes whether simply using existing multi-output modelling methods would be sufficient to prevent them from colliding. If not, ways to introduce the non-colliding constraint while also inducing cross-correlations could be explored.

As discussed in Chapter 2, one major limitation of Gaussian process regression is its computational complexity of $\mathcal{O}(n^3)$. During our exploration of Gaussian process regression, we came across an interesting approach proposed by Särkkä et al. (2013). They showed how to convert spatio-temporal Gaussian process regressions to stochastic partial differential equations, which can then be solved with Kalman filter (Kalman, 1960) and Rauch-Tung-Striebel smoother (Rauch et al., 1965). This approach reduces the computational complexity significantly to only scaling linearly with respect to the number of observations. One could consider ways to impose various constraints under this approach, which would be a significantly more efficient way to add constraints when modelling spatio-temporal data.

Appendix A

Appendix

Definition A.1. A Brownian motion is a stochastic process, $\{W(t); t \geq 0\}$, that has the following properties: (i) $W(0) = 0$, (ii) $W(t)$ is continuous in $t \geq 0$, (iii) for any $0 \leq s \leq t$, $W(t) — W(s) \sim \mathcal{N}(0, t—s)$, (iv) $W(t)$ has independent increments, i.e. if $0 \leq s_1 \leq t_1 \leq s_2 \leq t_2$, then $W(t_1) - W(s_1)$ and $W(t_2) - W(s_2)$ are independent.

Proposition A.1. A Brownian motion is a Gaussian process with mean 0 and covariance function $k(s, t) = \min\{s, t\}$.

Proof. Let $W = W(t) : t \geq 0$ be a Brownian motion and $t_1, t_2, \ldots, t_k$ be a subset of the index set such that $t_1 \leq t_2 \leq \ldots \leq t_k$. Let $a_1, a_2, \ldots, a_k \in \mathbb{R}$.

$$\begin{aligned}
&a_1 W(t_1) + a_2 W(t_2) + \ldots + a_k W(t_k) \\
=&a_k W(t_k) - a_k W(t_{k-1}) + a_k W(t_{k-1}) \\
&\quad + a_{k-1} W(t_{k-1}) - (a_k + a_{k-1}) W(t_{k-2}) + (a_k + a_{k-1}) W(t_{k-2}) \\
&\quad + a_{k-2} W(t_{k-2}) - (a_k + a_{k-1} + a_{k-2}) W(t_{k-3}) + (a_k + a_{k-1} + a_{k-2}) W(t_{k-3}) \\
&\qquad \vdots \\
&\quad + a_2 W(t_2) - (a_2 + a_3 + \ldots + a_k) W(t_1) + (a_2 + a_3 + \ldots + a_k) W(t_1) \\
&\quad + a_1 W(t_1) \\
=&a_k [W(t_k) - W(t_{k-1})]
\end{aligned}$$

$$
\begin{aligned}
&+(a_k+a_{k-1})[W(t_{k-1})-W(t_{k-2})]\\
&+(a_k+a_{k-1}+a_{k_2})[W(t_{k-2})-W(t_{k-3})]\\
&\;\;\vdots\\
&+(a_2+a_3+\ldots+a_k)[W(t_2)-W(t_1)]\\
&+(a_1+a_2+\ldots+a_k)W(t_1)
\end{aligned}
$$

Since $t_1 \leq t_2 \leq \ldots \leq t_k$, by the independent increment property of Brownian motion, $W(t_1)$, $W(t_2)-W(t_1)$, $W(t_3)-W(t_2)$, ..., $W(t_{k-1})-W(t_{k-2})$, $W(t_k)-W(t_{k-1})$ are independent and they are normally distributed. Linear combinations of normally distributed random variables have a (univariate) normal distribution. Therefore, by Definition A.2, $(W(t_1), W(t_2), \ldots, W(t_k))$ is a multivariate Gaussian random variable and it follows that Brownian motion is a Gaussian process according to Definition 2.1.

We proceed to find Brownian motion's mean and covariance functions.

Let $t \geq 0$, then

$$
m(t) = \mathbb{E}[W(t)] = \mathbb{E}[W(t)-W(0)] = 0
$$

Suppose $0 \leq s \leq t$,

$$
\begin{aligned}
\text{Cov}(W(s), W(t)) &= \text{Cov}(W(s), W(t)-W(s)+W(s))\\
&= \text{Cov}(W(s), W(t)-W(s)) + \text{Cov}(W(s), W(s))\\
&= 0 + \mathbb{V}(W(s))\\
&= s
\end{aligned}
$$

$\text{Cov}(W(s), W(t)-W(s)) = 0$ is due to the independent increments property of Brownian motion. If $0 \leq t \leq s$, then $\text{Cov}(W(s), W(t)) = t$. Therefore, the covariance function of Brownian motion is

$$
k(s,t) = \min\{s,t\}
$$

□

Definition A.2 (Multivariate Gaussian distribution)**.** A random vector $\mathbf{X} = (X_1, \ldots, X_k)^T$ follows a multivariate Gaussian distribution if for any constant vector $\mathbf{a} = (a_1, \ldots, a_k)^T \in \mathbb{R}^k$, the random variable $Y = \mathbf{a}^T \mathbf{X} = a_1 X_1 + \ldots + a_k X_k$ has a univariate normal distribution.

Theorem A.1 (Partitioned matrix inversion Press et al. (2007))**.** *Suppose the $N \times N$ matrix P is partitioned into*

$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

A is a $m \times m$ matrix, D is $p \times p$, B is $m \times p$ and C is $p \times m$ $(m + p = N)$. Then the inverse of P can be partitioned into

$$P^{-1} = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}$$

where $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, $\tilde{D}$ have the same sizes as A, B, C, D respectively and can be found by either

$$\begin{aligned}
\tilde{A} &= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \\
\tilde{B} &= -A^{-1}B(D - CA^{-1}B)^{-1} \\
\tilde{C} &= -(D - CA^{-1}B)^{-1}CA^{-1} \\
\tilde{D} &= (D - CA^{-1}B)^{-1}
\end{aligned}$$

or equivalently

$$\begin{aligned}
\tilde{A} &= (A - BD^{-1}C)^{-1} \\
\tilde{B} &= -(A - BD^{-1}C)^{-1}BD^{-1} \\
\tilde{C} &= -D^{-1}C(A - BD^{-1}C)^{-1} \\
\tilde{D} &= D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1}
\end{aligned}$$

Theorem A.2 (Partitioned matrix determinant Press et al. (2007))**.** *Suppose the $N \times N$ matrix P is partitioned into*

$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Then the determinant of P can be found by

$$\det(P) = \det(A)\det(D - CA^{-1}B) = \det(D)\det(A - BD^{-1}C)$$

Proof of Theorem 2.1. The probability density function of $\mathbf{y}$, which follows a multivariate normal distribution, is defined as follows

$$p(\mathbf{y}) = p(\mathbf{y}_1, \mathbf{y}_2) = (2\pi)^{-\frac{N}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right]$$

Let us focus on $(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})$ and let it be E,

$$\begin{aligned}
E =& (\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \\
=& \begin{bmatrix} (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T & (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \end{bmatrix} \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} \begin{bmatrix} (\mathbf{y}_1 - \boldsymbol{\mu}_1) \\ (\mathbf{y}_2 - \boldsymbol{\mu}_2) \end{bmatrix}
\end{aligned}$$

Let $\widetilde{\boldsymbol{\Sigma}} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$, then applying Theorem A.1 to $\boldsymbol{\Sigma}^{-1}$, we get

$$\begin{aligned}
E =& \begin{bmatrix} (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T & (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\Sigma}}^{-1} & -\widetilde{\boldsymbol{\Sigma}}^{-1}\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}\widetilde{\boldsymbol{\Sigma}}^{-1} & \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}\widetilde{\boldsymbol{\Sigma}}^{-1}\Sigma_{12}\Sigma_{22}^{-1} \end{bmatrix} \begin{bmatrix} (\mathbf{y}_1 - \boldsymbol{\mu}_1) \\ (\mathbf{y}_2 - \boldsymbol{\mu}_2) \end{bmatrix} \\
=& (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T \widetilde{\boldsymbol{\Sigma}}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1) \\
& - (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}\Sigma_{21}\widetilde{\boldsymbol{\Sigma}}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1) \\
& - (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T \widetilde{\boldsymbol{\Sigma}}^{-1}\Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\
& + (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T (\Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}\widetilde{\boldsymbol{\Sigma}}^{-1}\Sigma_{12}\Sigma_{22}^{-1})(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\
=& (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T \widetilde{\boldsymbol{\Sigma}}^{-1}[(\mathbf{y}_1 - \boldsymbol{\mu}_1) - \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2)] \\
& - (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}\Sigma_{21}\widetilde{\boldsymbol{\Sigma}}^{-1}[(\mathbf{y}_1 - \boldsymbol{\mu}_1) - \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2)] \\
& + (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\
=& [(\mathbf{y}_1 - \boldsymbol{\mu}_1)^T - (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}\Sigma_{21}]\widetilde{\boldsymbol{\Sigma}}^{-1}[(\mathbf{y}_1 - \boldsymbol{\mu}_1) - \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2)] \\
& + (\mathbf{y}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2)
\end{aligned}$$

Since $\boldsymbol{\Sigma}$ is a symmetric matrix, $\Sigma_{21} = \Sigma_{12}^T$. In addition, the inverse of a symmetric matrix is also symmetric, therefore, $\Sigma_{22}^{-1} = (\Sigma_{22}^{-1})^T$. So we have

$$E = [\mathbf{y}_1 - (\boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2))]^T \widetilde{\boldsymbol{\Sigma}}^{-1}[\mathbf{y}_1 - (\boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2))]$$

$$
\begin{aligned}
&+(\mathbf{y}_2-\boldsymbol{\mu}_2)^T\Sigma_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)\\
=&(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})+(\mathbf{y}_2-\boldsymbol{\mu}_2)^T\Sigma_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)
\end{aligned}
$$

where $\widetilde{\boldsymbol{\mu}}=\boldsymbol{\mu}_1+\Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)$.

Substituting E back into the probability density function $p(\mathbf{y}_1,\mathbf{y}_2)$ and applying Theorem A.2 to det($\boldsymbol{\Sigma}$), we get

$$
\begin{aligned}
p(\mathbf{y}_1,\mathbf{y}_2)=&(2\pi)^{-\frac{p}{2}}\det(\widetilde{\boldsymbol{\Sigma}})^{\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})\right]\\
&(2\pi)^{-\frac{q}{2}}\det(\Sigma_{22})^{\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{y}_2-\boldsymbol{\mu}_2)^T\Sigma_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)\right]\\
=&(2\pi)^{-\frac{p}{2}}\det(\widetilde{\boldsymbol{\Sigma}})^{\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})\right]p(\mathbf{y}_2)
\end{aligned}
$$

Using the Bayes' theorem, we find the conditional distribution of $\mathbf{y}_1$ given y_2

$$
\begin{aligned}
p(\mathbf{y}_1|\mathbf{y}_2)&=\frac{p(\mathbf{y}_1,\mathbf{y}_2)}{p(\mathbf{y}_2)}\\
&=\frac{(2\pi)^{-\frac{p}{2}}\det(\widetilde{\boldsymbol{\Sigma}})^{\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})\right]p(\mathbf{y}_2)}{p(\mathbf{y}_2)}\\
&=(2\pi)^{-\frac{p}{2}}\det(\widetilde{\boldsymbol{\Sigma}})^{\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}(\mathbf{y}_1-\widetilde{\boldsymbol{\mu}})\right]
\end{aligned}
$$

$$
\therefore \mathbf{y}_1|\mathbf{y}_2\sim\mathcal{N}(\widetilde{\boldsymbol{\mu}},\widetilde{\boldsymbol{\Sigma}})
$$

□

Definition A.3 (Stopping time (Oksendal, 2013))**.** Let $\{\mathcal{N}_t\}$ be an increasing family of σ-algebras of subsets of Ω. A function $\tau:\Omega\to[0,\infty]$ is called a *stopping time* with respect to $\{\mathcal{N}_t\}$ if

$$
\{\omega;\tau(\omega)\le t\}\in\mathcal{N}_t,\qquad\text{for all } t\ge 0.
$$

Definition A.4. (Oksendal, 2013) Let τ be a stopping time with respect to $\{\mathcal{N}_t\}$ and let $\mathcal{N}_\infty$ be the smallest σ-algebra containing $\mathcal{N}_t$ for all $t\ge 0$. Then the σ-algebra $\mathcal{N}_\tau$ consists of all sets $N\in\mathcal{N}_\infty$ such that

$$
N\bigcap\{\tau\le t\}\in\mathcal{N}_t\quad\text{for all } t\ge 0.
$$

Definition A.5 (Strong Markov property (Oksendal, 2013)). Let f be a bounded Borel function on $\mathbb{R}^n$, τ a stopping time with respect to $\mathcal{F}_t^{(m)}$, $\tau < \infty$ almost surely. Then

$$\mathbb{E}^x[f(X(\tau+h))|\mathcal{F}_\tau^{(m)}] = \mathbb{E}^{X(\tau)}[f(X(h))] \qquad \text{for all } h \geq 0.$$

Bibliography

Abrahamsen, P. and Benth, F. E. (2001). Kriging with inequality constraints. *Mathematical Geology*, 33(6):719–744.

Abramowitz, M. and Stegun, I. A. (1965). Handbook of mathematical functions with formulas, graphs, and mathematical table. In *US Department of Commerce*. National Bureau of Standards Applied Mathematics series 55.

Acton, S. T. and Bovik, A. C. (1998). Nonlinear image estimation using piecewise and local image models. *IEEE Transactions on Image Processing*, 7(7):979–991.

Agrell, C. (2019). Gaussian processes with linear operator inequality constraints. *Journal of Machine Learning Research*, 20(135):1–36.

Aıt-Sahalia, Y. and Duarte, J. (2003). Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, 116(1-2):9–47.

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148.

Chang, M.-W., Ratinov, L.-A., Rizzolo, N., and Roth, D. (2008). Learning and inference with constraints. In *AAAI*, pages 1513–1518.

Cramér, H. and Leadbetter, M. R. (2013). *Stationary and related stochastic processes: Sample function properties and their applications*. Courier Corporation.

Da Veiga, S. and Marrel, A. (2012). Gaussian process modeling with inequality constraints. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 529–555.

Damien, P. and Walker, S. G. (2001). Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2):206–215.

Deutsch, J. L. and Deutsch, C. V. (2012). Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference*, 142(3):763–772.

Dole, D. (1999). Cosmo: A constrained scatterplot smoother for estimating convex, monotonic transformations. *Journal of Business & Economic Statistics*, 17(4):444–455.

Doumerc, Y. (2005). *Matrices aléatoires, processus stochastiques et groupes de réflexions*. PhD thesis, Toulouse 3.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Dyson, F. J. (1962). A brownian-motion model for the eigenvalues of a random matrix. *Journal of Mathematical Physics*, 3(6):1191–1198.

Fine, S. and Scheinberg, K. (2001). Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2(Dec):243–264.

Flegal, J. M., Jones, G. L., et al. (2010). Batch means and spectral variance estimators in markov chain monte carlo. *The Annals of Statistics*, 38(2):1034–1070.

Fritsch, F. N. and Carlson, R. E. (1980). Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):141–149.

Golchi, S., Bingham, D. R., Chipman, H., and Campbell, D. A. (2015). Monotone emulation of computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):370–392.

González, S., Herrera, F., and García, S. (2015). Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity. *New Generation Computing*, 33(4):367–388.

Grabiner, D. J. (1999). Brownian motion in a weyl chamber, non-colliding particles, and random matrices. In *Annales de l'IHP Probabilités et statistiques*, volume 35, pages 177–204.

Hobson, D. G. and Werner, W. (1996). Non-colliding brownian motions on the circle. *Bulletin of the London Mathematical Society*, 28(6):643–650.

Israeli, A., Rokach, L., and Shabtai, A. (2019). Constraint learning based gradient boosting trees. *Expert Systems with Applications*, 128:287–300.

Johansson, K. (2002). Non-intersecting paths, random tilings and random matrices. *Probability theory and related fields*, 123(2):225–280.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Karlin, S. and McGregor, J. (1959). Coincidence probabilities. *Pacific J. Math.*, 9:1141–1164.

Katori, M. and Tanemura, H. (2004). Symmetry of matrix-valued stochastic processes and noncolliding diffusion particle systems. *Journal of mathematical physics*, 45(8):3058–3085.

Katori, M., Tanemura, H., Nagao, T., and Komatsuda, N. (2003). Vicious walks with a wall, noncolliding meanders, and chiral and bogoliubov–de gennes random matrices. *Physical Review E*, 68(2):021112.

König, W., O'Connell, N., et al. (2001). Eigenvalues of the laguerre process as non-colliding squared bessel processes. *Electronic Communications in Probability*, 6:107–114.

König, W., O'Connell, N., Roch, S., et al. (2002). Non-colliding random walks, tandem queues, and discrete orthogonal polynomial ensembles. *Electronic Journal of Probability*, 7.

Lee, C.-I. C. (1996). On estimation for monotone dose—response curves. *Journal of the American Statistical Association*, 91(435):1110–1119.

Liechty, J., Ramaswamy, V., and Cohen, S. H. (2001). Choice menus for mass customization: An experimental approach for analyzing customer demand with an application to a web-based information service. *Journal of Marketing research*, 38(2):183–196.

López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

Maatouk, H. and Bay, X. (2016). A new rejection sampling method for truncated multivariate gaussian random variables restricted to convex sets. In *Monte carlo and quasi-monte carlo methods*, pages 521–530. Springer.

Maatouk, H. and Bay, X. (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Micchelli, C. A. and Utreras, F. I. (1988). Smoothing and interpolation in a convex subset of a hilbert space. *SIAM Journal on Scientific and Statistical Computing*, 9(4):728–746.

Migliorati, S., Di Brisco, A. M., Ongaro, A., et al. (2018). A new regression model for bounded responses. *Bayesian Analysis*, 13(3):845–872.

Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

O'Connell, N. (2002). Random matrices, non-colliding processes and queues. *Séminaire de probabilités de Strasbourg*, 36:165–182.

Oksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Pakman, A. and Paninski, L. (2014). Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

Puchała, Z. (2005). A proof of grabiner theorem on non-colliding particles. *Probab. Math. Statist*, 25:129–132.

Ramsay, J. O. (1998). Estimating smooth monotone functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):365–375.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press.

Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450.

Riihimäki, J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 645–652.

Robertson, T. (1988). Order restricted statistical inference. Technical report.

Rodriguez-Yam, G., Davis, R. A., and Scharf, L. L. (2004). Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression. *Unpublished manuscript*.

Särkkä, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *IEEE Signal Process. Mag.*, 30(4):51–61.

Seeger, M., Teh, Y.-W., and Jordan, M. (2005). Semiparametric latent factor models. Technical report.

Tan, M., Fang, H.-B., Tian, G.-L., and Houghton, P. J. (2002). Small-sample inference for incomplete longitudinal data with truncation and censoring in tumor xenograft models. *Biometrics*, 58(3):612–620.

Tresp, V. (2000). A bayesian committee machine. *Neural computation*, 12(11):2719–2741.

Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate output analysis for markov chain monte carlo. *Biometrika*, 106(2):321–337.

Viana, F. A. (2016). A tutorial on latin hypercube design of experiments. *Quality and reliability engineering international*, 32(5):1975–1985.

Villalobos, M. and Wahba, G. (1987). Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities. *Journal of the American Statistical Association*, 82(397):239–248.

Wang, X. and Berger, J. O. (2016). Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1–25.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688.

Wolberg, G. and Alfy, I. (2002). An energy-minimization framework for monotonic cubic spline interpolation. *Journal of Computational and Applied Mathematics*, 143(2):145–188.

Wright, I. W., Wegman, E. J., et al. (1980). Isotonic, convex and related splines. *The Annals of Statistics*, 8(5):1023–1035.