Inference using Gaussian processes in animal movement modelling

Ionut Alexandru Paun

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

September 2022

Sine victoria non erit pax! (Without victory there will be no peace!)

Abstract

In recent years, the field of movement ecology has been changed dramatically by the capacity to collect accurate high-frequency telemetry data. In this thesis I present new statistical methods scalable to very large volumes of data being generated as there is a problem of scale dependence in most popular animal movement models.

Popular and widely used movement models in ecology are discrete-time movement models, where animals' positions are observed at discrete times. However, discrete-time models do not perform well when problems such as missing or irregular data are present. A remedy to the inefficiency of discrete-time movement models is to use continuous-time movement models, however the formulation of continuous-time movement models is often difficult and hard to interpret.

In this thesis, I first focus on discrete-time movement models, where through a study I illustrate one of the problems that discrete-time movement models pose - the specification in advance of the discretisation time-step. I then move on to probabilistic methods, widely used in the machine learning community, Gaussian processes (GPs), and I show that they are equivalent to many continuous-time movement models. Given that the primary goal of machine learning methods is to learn from large scale datasets, using robust continuous-time movement models such as Gaussian processes is highly advantageous for multiple reasons. These include their flexibility in choosing various covariance functions, their scalability to large datasets and their ability to analyse data, infer parameters of interest and quantify uncertainty within a non-parametric Bayesian approach.

I extend the standard Gaussian process (GP) into a non-stationary hierarchical Gaussian process, where both the movement process and the dynamic parameters of the movement model are Gaussian processes, which allows for increased flexibility to a wide range of behaviour modes that animals can exhibit. Throughout this thesis, I implement Gaussian processes on simulated and real tracking data using statistical libraries such as TensorFlow, which provide an accessible way to implement the model and gain access to GPU/HPC-accelerated machine learning libraries. I perform inference using optimisation methods such as maximum-a-posteriori (MAP) estimation, approximate sampling based inference methods such as Markov Chain Monte Carlo (MCMC) and variational inference methods on both synthetic and real datasets.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to express my gratitude to my PhD supervisors Dirk and Colin for helping and guiding me throughout my difficult adventure as a PhD student. I would also like to thank my family, especially my sister, Mihaela, who has always responded kindly to my many PhD related questions. Lastly, I would like to give thanks to the University of Glasgow, who has kindly accepted me as a undergraduate student almost 8 years ago.

Declaration of authorship

I hereby declare that the contents of this thesis are original, except where specific reference is made, and have been created under the supervision of my supervisors Dirk Husmeier and Colin Torney and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Nomenclature

Abbreviations:

GPS: Global Positioning System
CRW: Correlated random walk
CTCRW: Continuous-time correlated random walk
BRW: Biased random walk
SRW: Simple random walk
MSD: Mean squared displacement
KDE: Kernel density estimation
pdf: Probability distribution function
KS: Kolmogorov–Smirnov
HMM: Hidden Markov model
GP: Gaussian Process
DGP: Deep Gaussian process
MGP: Mixture of Gaussian processes
HMC: Hamiltonian Monte Carlo
MC: Monte Carlo
MCMC: Markov Chain Monte Carlo
MH: Metropolis-Hastings
CI: Credible interval
RTS: Rauch-Tung-Striebel
SPDE: Stochastic partial differential equation
SDE: Stochastic differential equation
OU: Orstein-Uhlenbeck
OUV: Orstein-Uhlenbeck velocity
OUF: Orstein-Uhlenbeck foraging
SE: Squared exponential
RBF: Radial basis function
RQ: Rational quadratic
GPU: Graphics processing unit
HPC: High Performance Computing

CPU: Central processing unit

Mathematical notation:

x: scalar value

$\mathbf{x}$: vector

$\mathbf{X}$: matrix

Chapter 1

Introduction to animal movement

Animal movement is a fundamental ecological process that is often complex and difficult to analyse and interpret, driven by processes that operate on multiple spatio-temporal scales. Understanding animal movement is essential as movement determines an individual fecundity rates and survival chances, thus affecting the trophic interactions between species. The movement of animals also has a crucial role in the stability of the ecosystems and spread of infectious diseases [Nathan et al., 2008].

The increased possibility to collect high accuracy, high frequency telemetry data from individual animals has lead to the growing development of statistical methods that can infer the dynamics of animal movement. Topics of interest are the underlying motivations and mechanism behind animal movement i.e. the drivers of movement such as internal state, landscape characteristics, habitat selection, motion capacity and/or navigational capacity [Nathan et al., 2008]. Much of the current research is focused on multi-state movement models, where the animal switches between various multiple behavioural states at different times [Morales et al., 2004, McClintock et al., 2012]. When using these kinds of models, the main problems that need to be addressed are how many discrete behavioural states (e.g. foraging, migrating, resting) can be determined from the data, the different structure of the underlying models (each distinct state is modelled by a different parameterised model) present on the movement path, how to model different transitions between the various behavioural modes and how often they occur [Morales et al., 2004].

Ecologists prefer movement models that are intuitive, easily interpretable and feasible to implement such as random walks or some variations of random walks. More complex models may be more realistic in terms of animal movement, but are also harder to implement and computationally demanding. The increasing size of the datasets is also a challenge as ecologists might lack the computational resources to fit complex models that are scalable to large amounts of data. Hence, there is a urgent need for developing realistic animal movement models and statistical techniques that can capture the dynamics of animal movement, and can process and analyse large amounts of data.

1.1 Animal movement data

Movement data usually consists of positions of an animal or a group of animals recorded at a set of discrete points in time. In this thesis, as I study the movement of land-based animals, the telemetry data recorded will be two-dimensional, longitude and latitude. In other cases, where the interest of study is the movement of marine animals and birds, the coordinate along the third dimension (depth or altitude) might be recorded using for example pressure sensors.

There are various methods to collect telemetry data, but using satellite positioning systems such as the Global Positioning System (GPS) or the Argos system, is the predominant data collection method used in recent years. Animals will be equipped with a tag, for example, a collar, and its locations are recorded at a set of discrete time points, until the battery runs out or the tag is removed or falls off the animal. Other types of data can be recorded using (VHF) radio [Cagnacci et al., 2010] or telemetry tags such as accelerometers [Brown et al., 2013].

In recent years the technological advances have made it possible to collect telemetry data at high temporal resolutions and over longer periods of time. The sampling scheme of observations that is employed might vary considerably depending on the battery's size and life and on the aim of the study. If the goal of the study is to research the long term behaviour of an individual animal or a group, then a sampling frequency of months can be a good option, however if the aim is the study of small-range behaviour, then a high sampling frequency of minutes is a better choice.

Various problems might arise while collecting telemetry data such as irregular or missing observations that can complicate statistical modelling of animal movement. Observations might be missing due to faulty tags and irregularities might be introduced due to sensor, battery or memory limitations. Other issues might arise from the lack of accuracy of the measurements, such as systematic biases and wrong conclusions (for example, a large measurement error has a negative impact on habitat selection) [Bjørneraas et al., 2010]. The measurement error of GPS tags might vary as it often depends on the number of satellites used, their position in space [Bjørneraas et al., 2010], but on average it is less than 30 m and this does not pose a problem if the scale of animal movement is comparatively large [Frair et al., 2010] and if the sampling step is sensibly chosen (for example, a sampling step of 1 second causes problems when measurement errors are large). Argos devices are even less accurate and might have measurement errors ranging from 150 m to several kilometres [Patterson et al., 2010].

A characteristic of movement data is its autocorrelated nature. That is, an animal's location in the near future will be dependent on its current location. If the data is recorded at high sampling frequencies, the autocorrelation will be high and should be accounted for when modelling the movement data. The statistical methodology developed to deal with autocorrelated observations centers around random walk models [Johnson et al., 2008, Fleming et al., 2014a].

After the collection of animal positions, movement models are formulated in a variety of ways. The model can be formulated in terms of raw positions, the displacement of positions i.e.

step, in terms of the velocity between consecutive positions, or in terms of change of directions between consecutive positions [McClintock et al., 2014].

Modern telemetry data together with the fact that animal movement is an inherently complex and multiscale process driven by many factors [Nathan et al., 2008, Fryxell et al., 2008] gives raise to many challenges for ecologists. The large-scale data with the added challenges of strong autocorrelation observations recorded with possible measurement error further complicates the already existing challenges. Hence, developing strong, robust and flexible statistical methodology to deal with these problems and enhance the knowledge of animal movement is a necessity.

1.2 Aim of the thesis

In this thesis, I focus on developing new statistical methods that are interpretable from an ecological perspective and scalable to large datasets. In Chapter 1, I present the ecological data and the problems that arise with analysing it. In Chapter 2, I discuss the background material necessary for understanding the main chapters of the thesis, namely, Chapters 3-6.

In Chapter 3 of the thesis, I start by using discrete-time movement models, where the discretisation step is specified in advance and is very important in setting up the model [Bovet and Benhamou, 1988, Harris and Blackwell, 2013]. More specifically, I use a basic correlated random walk movement model with Bayesian inference methods such as Markov Chain Monte Carlo (MCMC). In this chapter, I illustrate the limitations of discrete-time movement models by simulating various datasets with different sampling frequencies and then fit the model to the data with a different discretisation step. I test whether I can detect systematic mismatch between the model and the data using different test statistics and by computing posterior p-values. The study showed that the model mismatch was not consistently detected, thus exemplifying a central flaw with the discrete-time models, that is the specification in advance of the discretisation step. In Chapters 4-6, I focus on continuous-time movement models and their corresponding covariance functions, that are more flexible and do not have the limitations of the discrete-time movement models. I focus exclusively on a probabilistic method known as Gaussian processes (GPs), which are flexible non-parametric methods widely used in machine learning community for regression and classification purposes. The focus is on natural extensions of the standard Gaussian process (GP), non-stationary GPs, where all or a subset of the GP's parameters (e.g. lengthscale, amplitude or noise variance) are allowed to vary in time or space.

In Chapter 4, I show that GPs are equivalent to many continuous-time movement models by specifying an appropriate and corresponding covariance function. More specifically, I show how popular continuous-time movement models such as Brownian bridge [Hooten et al., 2017], Orstein-Uhlenbeck (OU) [Uhlenbeck and Ornstein, 1930], Orstein-Uhlenbeck velocity model (OUV) [Johnson et al., 2008] and Orstein-Uhlenbeck-Foraging (OUF) [Fleming et al., 2014a]

can be reintroduced as GPs. Moreover, significant advantages are gained from a computationally point of view from having access to powerful machine learning libraries and various already implemented inference techniques such as maximum-a-posteriori (MAP) or Markov Chain Monte Carlo (MCMC).

In Chapters 5-6, I extend the stationary GP model to a non-stationary GP, where the parameters of the GP are allowed to vary, since stationary GPs lack the flexibility to model non-stationary data [Paciorek and Schervish, 2004, Gibbs, 1997, MacKay, 1997]. I follow the approach described by Heinonen et al. [2016], where the parameters are also modelled by another GPs and I essentially construct a double-layer hierarchical GP. In Chapter 5, the model is a hierarchical spatial GP process, where the parameters on the first layer of the GP are dependent on the GPS locations from multiple individuals. I derive a novel covariance function that links positional data with the dynamic parameters of a velocity model and I aim to infer the drivers of movement, the environment's characteristics, or in other words infer the parameters of the model that characterise the environment using MAP estimation and gradient based MCMC methods. I apply my method to a synthetic dataset and then to telemetry data from the Serengeti wildebeest migration. In Chapter 6, I make the inference approach scalable to potential millions of points, by using a variational inference method instead of sampling-based methods such as MCMC, which are computationally expensive. I apply my method on multiple synthetic datasets and on an empirical dataset - individual household average power consumption. The method developed in this chapter is a general method applicable to different types of data, not only to movement telemetry data.

Chapter 2

Review of background theory

2.1 Discrete-time movement models

Introduction to random walks

Random walks are one of the most common and simple methods that are used to model movement data in a wide range of biological settings such as cell movement [Tweedy et al., 1977, Hall, 1977, Liepe et al., 2012, Taylor et al., 2013, Jones et al., 2015, Panotopoulos et al., 2018], animals movement [Skellam, 1951, 1973, Codling et al., 2004, Morales et al., 2004, Nouvellet et al., 2015, Michelot and Blackwell, 2019] or in a financial setting [Bachelir, 1900, Fama, 1965]. The term 'random walk' was first used by Karl Pearson in 1905 [Pearson, 1905], when in a letter to Nature, he used a simple random walk to model a mosquito infestation in a forest. The letter was answered by Lord Rayleigh [Rayleigh, 1905], who previously used random walks in 1880 to model sound waves through heterogeneous materials. However, the foundation of random walks was laid out previously by the botanist Brown [Brown, 1828] in his work regarding the irregular motion of individual pollen particles, which is now known as Brownian motion (or diffusion). Later on, physicists such as Albert Einstein [Einstein, 1905, 1906], and then Smoluchowski [Smoluchowski, 1916] published papers on random walks, which they used to model the path of a large dust particle in the air.

There are different types of random walks, the simplest of them is the Brownian motion, where the movement is uncorrelated and unbiased. Uncorrelated means that the direction of the movement is not influenced by the previous directions of movement. Unbiased means that there is no preference for a particular direction. One example of a very simple uncorrelated and unbiased random walk would be a random walk restricted on a lattice, where you have equal probability of going either up and down, or left or right. Also, it is worth noting that the process defined by an uncorrelated random walk is Markovian with regards to the location due to the fact that the location at each step is dependent only on the location at the previous step. In addition, Brownian motion can be shown to produce the heat equation (or standard diffusion)

[Vvedensky, 2019], Section 2.1.2.

When modelling animal movement, the diffusion equations might be used as a very basic model to compare with more complex models. This simple model can be extended to more complicated and realistic movement patterns by including correlation between successive steps. The correlated random walk (CRW) is also called 'persistent' as the successive steps are correlated. In a CRW model each step tends to point in the same direction as the previous one, however, the persistence fades away gradually in time and the directions become uniformly distributed in the long term [Benhamou, 2006]. It can be said that a CRW model is a RW with an introduced local bias.

Another extension would be the biased random walk model (BRW), which is a random walk model with global directional bias. The BRW can be biased in several ways, one by having a higher probability of moving into a specific direction, rather than having an equal probability of moving in either direction, or by having the walker move further along a specific direction. An example to illustrate this concept is a random walk particle moving three spaces to the left each time it goes left and one space each time it goes to the right. Another possible extension would be the biased persistent random walk in which the walker is 'biased' and 'persistent'. Moreover, a random walk particle might present different levels of bias and persistence along the path [Codling et al., 2004]. Another very popular extension is the Ornstein-Uhlenbeck process [Uhlenbeck and Ornstein, 1930], which is a mean-reverting process, with the strength of the attraction to the mean being stronger as the random walk particle moves away from the mean. The Ornstein-Uhlenbeck process is discussed in more detail in Section 2.5.

Fundamentals of random walk model

The simple unbiased random walk (SRW) is the foundational model for diffusion processes. In a SRW model, a random walk particle is equally likely to move in each possible direction and its direction is uncorrelated i.e. the direction taken at a particular time is independent of all previous times. Let l be the step-length, τ the time for a single step, p the probability for a step to the right, $q = 1 - p$ the probability for a step to the left and let $\mathrm{P}_N(\mathrm{m})$ the probability to find the walker at position $x = ml$ at time $t = N\tau$. The probability $\mathrm{P}_N(\mathrm{m})$ satisfies the following stochastic difference equation

$$P_{N+1}(m) = pP_N(m-1) + qP_N(m+1). \tag{2.1}$$

I specialise to the case $p = q = \frac{1}{2}$ and from the previous equation I subtract $\mathrm{P}_N(\mathrm{m})$ on both sides and then taking limit as N grows large, the differences become differentials

$$P_{N+1}(m) - P_N(m) = \frac{1}{2}(P_N(m-1) + P_N(m+1) - 2P_N(m)). \tag{2.2}$$

The left hand side term is approximately equal to $\tau\frac{\partial P}{\partial t}$ and the right hand side term is approximately equal to $l^2\frac{\partial^2 P}{\partial x^2}$ (using finite differences of first and second order). Rearranging and denoting the term $D=\frac{l^2}{2\tau}$ (the diffusion coefficient), the diffusion equation is

$$\frac{\partial P}{\partial t}=D\frac{\partial^2 P}{\partial x^2}. \tag{2.3}$$

The diffusion equation can be solved with the following boundary conditions: $P(x,t)\to 0$ as $x\to\pm\infty$ for all t and $P(x,0)=\gamma(x)$. The diffusion equation or heat equation admits a Gaussian function as a solution

$$P(x,t)=\frac{1}{\sqrt{2\pi\sigma^2(t)}}\exp\left(\frac{-x^2}{2\sigma^2(t)}\right), \tag{2.4}$$

where $\sigma^2=2Dt$. The mean location of a random variable X at time t, $\mathbb{E}(X_t)$, and the mean squared displacement (MSD) $\mathbb{E}(X_t^2)$ are defined as

$$\mathbb{E}(X_t)=\int_{-\infty}^{\infty}xP(x,t)dx. \tag{2.5}$$

$$\mathbb{E}(X_t^2)=\int_{-\infty}^{\infty}x^2P(x,t)dx. \tag{2.6}$$

Keeping in mind that $P(x,t)$ is Gaussian distributed with mean 0 and variance σ^2, I get that

$$\mathbb{E}(X_t)=0. \tag{2.7}$$

$$\mathbb{E}(X_t^2)=\sigma^2=2Dt. \tag{2.8}$$

This means that the SRW is unbiased i.e. it has no preferred direction and that the MSD increases linearly with time, a standard property of a diffusive process.

Correlated discrete-time movement models

The discrete-time correlated random walk (CRW) [Kareiva and Shigesada, 1983, Turchin, 1998, Siniff and Jessen, 1969, Bovet and Benhamou, 1988] and its extensions [Morales et al., 2004, McClintock et al., 2012] are the foundation of the movement data models. In a discrete-time model framework methods from the time series literature can be borrowed and implemented [Anderson-Sprecher and Ledolter, 1991]. A very important aspect when working with discrete-time models is the specification in advance of a suitable discretisation time-step. Ideally, the time-step should be as small as possible and carefully chosen based on the aim of the experiment. However, the discretisation time-step might be chosen by different criteria due to experimental, logistical constraints such as for example, battery life when modelling animal movement data, or because of computational efficiency constraints. More specifically, as discussed in Chapter 1, the sampling frequency might vary from recording observations every few minutes to months

depending on whether the goal of the study is to research the short-term or the long-term behaviour of an individual animal. Moreover, there is a trade-off between the sampling frequency and the battery life, as high data resolution recording reduces battery life [Johnson and Ganskopp, 2008].

In a CRW model, the persistence gradually disappears in the long term [Benhamou, 2006]. Therefore, a large time-step will lead to a loss in correlation, transforming the path into a random walk. Turchin [1998], Morales et al. [2004], Haydon et al. [2008], Hopcraft et al. [2014] implemented statistical models for components of discrete-time random walks. Those components include the step-length and the associated observed turning angle relative to the previous step between each pair of successive observations. The distributions commonly used are Gamma or Weibull distributions for the step-length [Morales et al., 2004], respectively Uniform, Von Mises, Wrapped Cauchy or Wrapped Normal distribution for the turning angle [Kareiva and Shigesada, 1983, Siniff and Jessen, 1969, Langrock et al., 2014, Batschelet, 1981, Mardia and Jupp, 1999].

The Weibull distribution has two parameters, one parameter controlling the scale and the other controlling the shape, and it is considered a good option to model the step-lengths [Morales et al., 2004]. More specifically, depending on the value of the shape parameter, the Weibull distribution is equivalent to an exponential distribution (when the shape parameter is 1), it is suitable to model long step lengths due to a long tail (when the shape parameter is less than 1) and it is equivalent to the step-length distribution of a standard diffusion process when the shape parameter is 2 [Morales et al., 2004].

Assigning the Uniform distribution to the turning angles means that the direction is random, thus the random walk is not 'persistent'. The Wrapped Normal, Von Mises and the Wrapped Cauchy distributions are circular distributions and all have two parameters, one controlling the scale and the other one is the mean (location). The Wrapped Cauchy distribution is more peaked and has heavier tails than the aforementioned circular distributions, and when the scale parameter tends to zero, it transforms to a Uniform distribution [Morales et al., 2004].

The probability density function of the Weibull distribution is

$$p(x|a,b) = \frac{b}{a}\left(\frac{x}{a}\right)^{b-1} \exp\left(-\left(\frac{x}{a}\right)^{b}\right), \tag{2.9}$$

where $x > 0$, $b, a > 0$, b is the shape parameter and a is the scale parameter.

The probability density function of the Gamma distribution is

$$p(x|\alpha,\beta) = \frac{\beta^{\alpha} x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)}, \tag{2.10}$$

where $x > 0$, $\alpha, \beta > 0$, $\Gamma(\alpha)$ is the Gamma function, α is the shape parameter, $\beta = 1/\theta$ is the inverse-scale parameter, and θ is the scale parameter.

The probability density function of the standardised Wrapped Cauchy distribution is

$$p(\theta|c) = \frac{1-c^2}{2\pi(1+c^2-2c\cos\theta)}, \tag{2.11}$$

where $0 \leq \theta \leq 2\pi$ and c is the shape parameter, with $0 < c < 1$. The Weibull, Gamma and Wrapped Cauchy distributions are used in Chapter 3 to model the step-lengths, respectively the turning angles and are plotted for various values of the parameters in Figure 3.2.

The basic random walk model, while it is a good model to start with is not realistic in terms of animal movement, as the movement of animals is a complex multiscale process. Certain extensions of random walks like CRW or BRW are better than SRW in modelling movement in the short term, but the 'persistence' fades away gradually. The pattern of animal movement changes according to the habitat the animals find themselves in or by human or animal interactions [Morales et al., 2004], therefore various extensions of the basic random walk are needed to model effectively this change in animal behaviour.

To capture effectively the complexity of animal movement, mixtures of random walk models that contain non-stationary distributions for the components of model (step-length and turning angle), hidden Markov processes and different sources of bias [Morales et al., 2004] have been developed. These models are called discrete-time multistate movement models, where a certain movement model is associated with a distinct behavioural state [Morales et al., 2004, McClintock et al., 2012, Langrock et al., 2014]. These behavioural states might include foraging, predator avoidance, homing, and landscape exploration [Hooten et al., 2017], Section 1.1.2. The behavioural modes might be classified as 'encamped', where you have short step-lengths and low directional persistence or 'exploratory', where you have long step-lengths and high directional persistence [Morales et al., 2004]. An individual animal can switch between different behavioural states with certain probabilities, which are stored in transition matrices [Taylor et al., 2013, Morales et al., 2004, Langrock et al., 2014]. These probabilities will change depending on factors such as habitat type or interactions between individuals, for example the probability of an animal becoming 'encamped' will increase if the habitat has more food [Morales et al., 2004, Haydon et al., 2008].

Among discrete-time movement models, one popular model is the Hidden Markov model (HMM). The HMM is a time-series model that consists of two components, an observed part consisting of the observations and an unobservable or hidden (latent) discrete-states [Patterson et al., 2009, Langrock et al., 2012, Michelot et al., 2016]. A HMM can be considered a special case of a state-space model, where the number of the hidden states is finite [Langrock et al., 2012]. HMMs are flexible and intuitive models due to accounting for multiple underlying behavioural states, can include covariates and can explain the correlation in the movement data, however, they assume that the location measurement error and the missing data were low [Patterson et al., 2009, Langrock et al., 2012, Michelot et al., 2016]. Michelot et al. [2016] created a

package in R that implements HMMs, and is accessible and relatively easy to use for ecologists. Another example is Langrock et al. [2015], where the authors use non-parametric inference and HMMs to model beaked whale dive data.

In conclusion, while multistate discrete-time models are a good tool to model complex movement data, due to their diversity and accessible formulation in terms of step-lengths and turning-angles [Morales et al., 2004, McClintock et al., 2014], an important limitation is the specification in advance of the unknown discretisation step [Bovet and Benhamou, 1988, Harris and Blackwell, 2013, Avgar et al., 2013, Nouvellet et al., 2015, Fleming et al., 2014a]. This causes problems dealing with irregular observations [Harris and Blackwell, 2013, Avgar et al., 2013], as the choice of sampling rate might be chosen due to GPS-collar battery life [Nouvellet et al., 2015, Fleming et al., 2014a] rather than important behavioural events [Bovet and Benhamou, 1988]. Moreover, missing data can occur for a multitude of reasons including weather and terrain [Morales et al., 2004, McClintock et al., 2012], or the collar might fall off or its battery might run out.

2.2 Gaussian processes

A history of Gaussian processes

A Gaussian process (GP) is a stochastic process (a collection of random variables indexed by time or space) such that any subset of those random variables is jointly Gaussian. GPs are named after Carl Friedrich Gauss and can be seen as an infinite-dimensional generalisation of multivariate Normal distributions. GPs have been studied and applied in different domains for decades. For example, the Wiener process is a type of GP. Since GPs are stochastic processes that can be indexed by time, probably they were first used for time series prediction in works that date back to 1940's [Wiener, 1949, Kolmogorov, 1941]. GPs were also used in the field of geostatistics [Matheron, 1973, Whittle, 1963], where prediction using GPs is called kriging, named after the South African mining engineer D. G. Krige by Matheron [1973]. Another field where GPs prediction was widely used is meteorology [Thompson, 1956], where GPs prediction was restricted to 2 and 3 dimensional input spaces. An early reference about the use of a GP as a prior over functions appears in the works of O'Hagan and Kingman [1978]. Another early appearance of GPs in the statistics community would be in Sacks et al. [1989]. In the machine learning community, GPs started being used in the 90's by Rasmussen and Williams [2006] and nowadays, GPs are still used mostly in the spatial statistics field [Gelfand et al., 2010].

Introduction to Gaussian processes

This subsection is based on Murphy [2012], Section 15.1. Assume that at inputs x_i, the outputs y_i are observed and that $y_i = f(x_i)$ for some unknown function f with possible added noise. GPs

are a non-parametric method, that consist of inferring a distribution over functions given the data, $p(f|\mathbf{x},\mathbf{y})$, and then to use this to make predictions given new inputs $\mathbf{x}^*$, i.e. to compute

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}) = \int p(\mathbf{y}^*|f,\mathbf{x}^*)p(f|\mathbf{x},\mathbf{y})df, \tag{2.12}$$

where $\mathbf{x}$ is a set consisting of all the input (training) points and $\mathbf{y}$ is a set consisting of all the observed data points (output) [Murphy, 2012].

A prior distribution is set over the latent function f and likewise the posterior distribution will be obtained over functions. In a GP setting it is sufficient to be able to define a distribution over the functions values at a finite, but arbitrary, set of points, say $x_1, x_2, \ldots, x_N$. A GP assumes that $p(f(x_1), f(x_2), \ldots, f(x_N))$ is jointly Gaussian, with a mean function $\mu(\mathbf{x})$ and covariance matrix $\mathbf{K}$, defined by $\mathbf{K}(x_i, x_j) = k(x_i, x_j)$, where k is a positive semi-definite kernel (covariance function) and x_i, x_j are random input points. The conventional notation is the following

$$f(\mathbf{x}) \sim \mathscr{GP}(\mu(\mathbf{x}), \mathbf{K}). \tag{2.13}$$

Kernels

This subsection is mainly based on Rasmussen and Williams [2006], Chapter 4. A key fact of GPs is that they can be completely defined by their mean and (kernel) covariance functions. Since it is common to assume that the prior mean of the GP to be zero [Murphy, 2012], Section 15.2, the kernel completely defines the process' behaviour. A crucial point is that the covariance function is used to ensure that values that are close together in input space will produce output values that are close together. Figure 15.1 in Murphy [2012] provides an illustration of this key idea.

An important aspect of the process' behaviour is the stationarity property. A stationary process depends only on the difference $x - x'$, and a non-stationary process depends on the actual position of the random points x and x'. Another important aspect is the isotropy property. An isotropic process depends only on the difference $|x - x'|$, thus making the process invariant to all rigid motions [Rasmussen and Williams, 2006], Chapter 4. For example, the Ornstein–Uhlenbeck covariance function is isotropic.

Moreover, a valid kernel is a real-valued function of two arguments, $k(x, x') \in \mathbb{R}$ for x, x' random points in the input space. The function is symmetric i.e. $k(x, x') = k(x', x)$ and non-negative. Also, a valid covariance function must be positive semi-definite. The covariance matrix $\mathbf{K}$, defined by $\mathbf{K}(x, x') = k(x, x')$ is called positive semi-definite if the following relationship holds

$$\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0, \tag{2.14}$$

for all real vectors $\mathbf{v}$.

Standard kernels

I give a list of standard isotropic kernels. Further on, I list a few examples of non-stationary kernels [Rasmussen and Williams, 2006], Chapter 4. For the following kernels, the lengthscale parameter l and the parameter σ are always positive.

Squared exponential kernel

Following Murphy [2012], Section 14.2.1, the squared exponential (SE) kernel or the radial basis function (RBF) kernel has the following form

$$k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\mathbf{x}'\right)\right). \tag{2.15}$$

If $\boldsymbol{\Sigma}$ is diagonal, the automatic relevance determination (ARD) kernel is obtained with the following formula

$$k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{1}{l_j^2}(x_j - x_j')^2\right), \tag{2.16}$$

where l_j is the characteristic lengthscale of the dimension j. If $l_j \to \infty$, then the corresponding dimension is ignored. If $\boldsymbol{\Sigma}$ is spherical[1], the isotropic SE kernel is obtained

$$k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}||\mathbf{x}-\mathbf{x}'||^2\right), \tag{2.17}$$

where $||\mathbf{x}||$ is the L2-norm or Euclidean norm. An Euclidean norm of a vector $\mathbf{x} = (x_1,\ldots,x_n)$ is given by the following relationship

$$||\mathbf{x}|| = \sqrt{x_1^2 + \cdots + x_n^2}. \tag{2.18}$$

The lengthscale l determines the length of the 'wiggles' in your function i.e. the smoothness of the function and the signal variance parameter σ^2 determines the vertical variation. The kernel is infinitely differentiable.

Rational quadratic kernel

The rational quadratic (RQ) kernel has the following form

$$k_{\mathrm{RQ}}(\mathbf{x},\mathbf{x}') = \sigma^2\left(1 + \frac{1}{2\alpha l^2}||\mathbf{x}-\mathbf{x}'||^2\right)^{-\alpha}. \tag{2.19}$$

[1] A matrix $\boldsymbol{\Sigma}$ is called spherical or isotropic if it is proportional to the identity matrix i.e. $\boldsymbol{\Sigma} = \lambda\mathbf{I}$, where λ is a constant.

The RQ kernel is equivalent to the sum of multiple SE kernels with different lengthscales, with the positive scale-mixture parameter α determining the weighting between multiple lengthscales. When $\alpha \to \infty$, the RQ kernel is identical to the SE kernel [Rasmussen and Williams, 2006], Section 4.2.

Matérn kernel

The Matérn kernel which is often used in GP regression has the following form

$$k(\mathbf{r}) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\mathbf{r}}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}\mathbf{r}}{l} \right), \tag{2.20}$$

where $\mathbf{r} = ||\mathbf{x} - \mathbf{x}'||$, $\nu > 0$, and K_{ν} is a modified Bessel function. As $\nu \to \infty$, this is the SE kernel. If $\nu = \frac{3}{2}$ or $\nu = \frac{5}{2}$, the corresponding kernels are Matérn 3/2 and Matérn 5/2. If $\nu = \frac{1}{2}$, the Matérn 1/2 kernel is obtained and the kernel formula simplifies to

$$k(\mathbf{r}) = \sigma^2 \exp\{-\mathbf{r}/l\}. \tag{2.21}$$

In one-dimension the Matérn 1/2 kernel can be used to define the Ornstein-Uhlenbeck (OU) process, which describes the velocity of a particle undergoing Brownian motion. The corresponding function is continuous, but nowhere differentiable, therefore is very ragged.

Non-stationary kernels

Periodic kernel

The periodic kernel has the following form

$$k_{\text{Per}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(\frac{-2\sin^2 \frac{\pi||\mathbf{x}-\mathbf{x}'||}{p}}{l^2} \right), \tag{2.22}$$

where p is the period of the function, $p > 0$.

Wiener kernel

The Wiener process (also called continuous-time Brownian motion) has the following kernel formula

$$k_{\text{Wiener}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \min(\mathbf{x}, \mathbf{x}'), \tag{2.23}$$

where $\mathbf{x}, \mathbf{x}'$ are real vectors .

Neural network kernel

The construction of this kernel and all the formulas are due to Neal [1996], Rasmussen and Williams [2006], Williams [1998]. Consider a network with input $\mathbf{x}$ that has one hidden layer with N_H units. The linear combination of the outputs of the hidden units with a bias b produces $f(\mathbf{x})$. The mapping has the following form

$$f(\mathbf{x}) = b + \sum_{j=1}^{N_j} v_j h(\mathbf{x}, \mathbf{u}_j), \tag{2.24}$$

where the v_js are the hidden-to-output weights and $h(\mathbf{x}, \mathbf{u})$ is the hidden unit transfer function, which depends on the input-to-hidden weights $\mathbf{u}$. If the error function $h(z) = \text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ is the transfer function, then let $h(\mathbf{x}, \mathbf{u}) = \text{erf}\ (u_0 + \sum_{j=1}^{D} u_j x_j)$, and choose $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Thus, the neural network kernel is obtained [Williams, 1998]

$$k_{NN}(\mathbf{x}, \mathbf{x}') = \frac{2}{\pi} \sin^{-1} \left(\frac{2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}}'}{\sqrt{2\mathbf{x}^T \Sigma \mathbf{x}'} \sqrt{2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}}'}} \right), \tag{2.25}$$

where $\tilde{\mathbf{x}}' = (1, x_1, \ldots, x_d)$ is the augmented input vector.

The covariance functions that are mostly used in this thesis are RBF and Matérn 1/2. I also show the full derivations of the Matérn 1/2 and of the Brownian motion (Wiener kernel) covariance functions in Chapter 4 of this thesis. Other examples of kernels or covariance functions, stationary or non-stationary can be found, but it is also possible to obtain other kernels by summing, multiplying or convoluting known kernels to obtain more flexible and complex processes. While these kernels might be useful to detect trends in the data, they might also lead to overfitting and to difficult parameter inference. They might not have closed form solutions and might require computationally demanding inference techniques [Wilson and Adams, 2013].

Simulating from a Gaussian process prior

I simulate from a GP prior and illustrate the differences between various kernels. The training data consists of 100 points between 0 and 2. The hyperparameters, the lengthscale and the signal variance, of every kernel are set to 1.

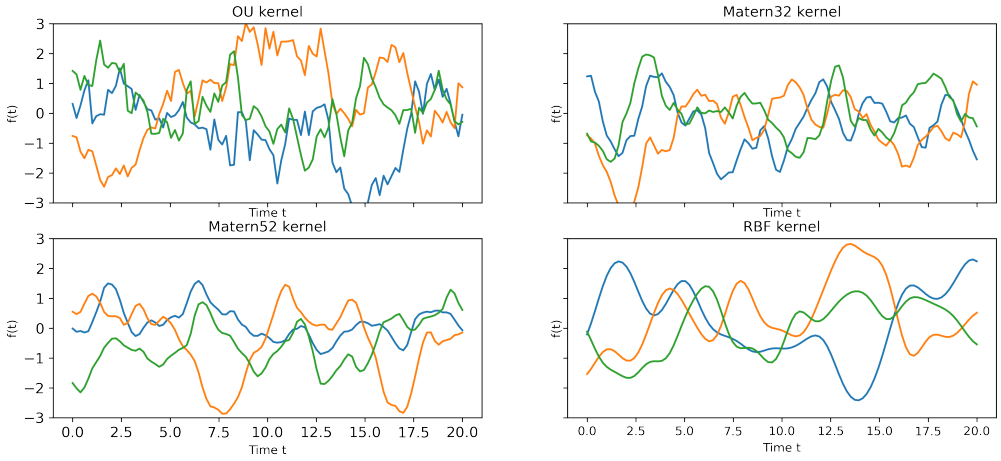


Figure 2.1: Multiple simulations from a Gaussian process prior with different kernels.

The results in Figure 2.1 are as expected, since the plots become smoother from left to right, top to bottom i.e. the Matérn 1/2 (OU) kernel ($\nu = \frac{1}{2}$) is nowhere differentiable, Matérn 3/2 ($\nu = \frac{3}{2}$) is once differentiable, Matérn 5/2 ($\nu = \frac{5}{2}$) is twice differentiable and RBF ($\nu \to \infty$) is infinitely differentiable.

Regression in a Gaussian process model

This section closely follows Murphy [2012], Section 15.2 and Rasmussen and Williams [2006], Chapter 2. A GP prior is set on the latent function f such that

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}),\ \mathbf{K}(x_i, x_j)), \tag{2.26}$$

where $\mathbf{x}$ is the vector of input points, $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ is the mean function and $\mathbf{K}(x_i, x_j) = \mathbb{E}[(f(x_i) - m(x_i))(f(x_j) - m(x_j))^T]$, for two random input points x_i and x_j. For any finite set of points, this process defines a joint Gaussian

$$p(\mathbf{f}|\mathbf{x}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}), \tag{2.27}$$

where $\mathbf{K}(x_i, x_j) = k(x_i, x_j)$, k is a kernel, $\boldsymbol{\mu} = (m(x_1), \ldots, m(x_N))$ and $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_N))$. The vector notation can be written as $\mathbf{f} = f(\mathbf{x})$, and this notation convention is used throughout the thesis. It is common to use a prior mean function of $m(\mathbf{x}) = 0$, since the GP is flexible enough to model the posterior mean arbitrarily well [Murphy, 2012], Section 15.2. However, if there is a trend in the data, a good approach is to use a semi-parametric model, where a linear model is fitted to the mean of the process and a zero-mean GP is applied to the residuals [Murphy, 2012], Section 15.2.6.

Predictions with Gaussian processes without added noise

Suppose that the training set data is $\mathbf{x}$, consisting of N points, and $f(x_i)$ is the function evaluated at a random point x_i. In this section I start with the simple case, where I assume that there is no noise added to the data and I wish to predict the function values $\mathbf{f}^*$ at a new set of N^* test points $\mathbf{x}^*$.

The joint distribution of the GP has the following form

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}^* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}^* \\ \mathbf{K}^{*^T} & \mathbf{K}^{**} \end{pmatrix} \right), \tag{2.28}$$

where $\mathbf{K} = \mathbf{K}(\mathbf{x},\mathbf{x})$ is $N \times N$, $\mathbf{K}^* = \mathbf{K}(\mathbf{x},\mathbf{x}^*)$ is $N \times N^*$ and $\mathbf{K}^{**} = \mathbf{K}(\mathbf{x}^*,\mathbf{x}^*)$ is $N^* \times N^*$. Using standard rules for conditioning Gaussians [Murphy, 2012], Equations 4.120-4.121, the posterior has the following form

$$p(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{f}) = \mathcal{N}(\mathbf{f}^*|\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*). \tag{2.29}$$

$$\boldsymbol{\mu}^* = \mu(\mathbf{x}^*) + \mathbf{K}^{*^T}\mathbf{K}^{-1}(\mathbf{f} - \mu(\mathbf{x})). \tag{2.30}$$

$$\boldsymbol{\Sigma}^* = \mathbf{K}^{**} - \mathbf{K}^{*^T}\mathbf{K}^{-1}\mathbf{K}^*, \tag{2.31}$$

where $\mu(\mathbf{x}) = \boldsymbol{\mu}$.

By having the posterior distribution in closed form, samples can be drawn directly from the posterior $p(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{f})$. Since the observations are noiseless, the GP model goes perfectly through the observed points i.e. act as an interpolator for the training data, and reverts to prior knowledge outside of the observed data. Therefore, the uncertainty increases when moving further away from the observed data, given a valid kernel.

Gaussian processes predictions on noisy observed data

Now consider the case when noise is added to the data, $y_i = f(x_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0,\sigma_y)$, where the noise terms ε_i are independent. Then, the covariance of the noisy data $\mathbf{y}$ is

$$\mathrm{Cov}[y_p,\, y_q] = \mathbf{K}(x_p,\, x_q) + \sigma_y^2 \delta_{pq}, \tag{2.32}$$

where $\delta_{pq} = \mathbf{I}(p = q)$, a Kronecker delta term. Therefore,

$$\mathrm{Cov}[\mathbf{y}|\mathbf{x}] = \mathbf{K} + \sigma_y^2\mathbf{I} = \mathbf{K}_y. \tag{2.33}$$

The second matrix is diagonal because I assumed the noise terms were independently added to each observation. Moreover, I assume the mean is 0 for notational simplification. Using Equation 2.28, but replacing the noise-free matrix $\mathbf{K}$ with the noisy version $\mathbf{K}_y$, the joint density

of the observed data and the latent function on the test points is given by

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{K}^* \\ \mathbf{K}^{*^T} & \mathbf{K}^{**} \end{pmatrix}\right). \tag{2.34}$$

Thus, using the Equations 2.29-2.31, the posterior predictive density is

$$p(\mathbf{f}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{f}^*|\, \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*). \tag{2.35}$$

$$\boldsymbol{\mu}^* = \mathbf{K}^{*^T}\mathbf{K}_y^{-1}\mathbf{y}. \tag{2.36}$$

$$\boldsymbol{\Sigma}^* = \mathbf{K}^{**} - \mathbf{K}^{*^T}\mathbf{K}_y^{-1}\mathbf{K}^*. \tag{2.37}$$

Inference in Gaussian processes

This subsection is mainly based on Murphy [2012], Section 15.2.4. and Rasmussen and Williams [2006], Chapter 2. While working in a GP setting, problems of interest include the computation of the posterior distribution over the latent function given the observed data and of the posterior predictive distribution at a new set of test points. Another important problem of interest is inferring the GP parameters within a Bayesian framework. Usually, in a standard GP regression framework the data is assumed to have Gaussian distributed noise. Thus, the likelihood function is given by

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}). \tag{2.38}$$

The marginal likelihood (the latent function $\mathbf{f}$ is marginalised out) has the following form

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{x}) p(\mathbf{f}|\mathbf{x}) d\mathbf{f}. \tag{2.39}$$

The prior on the latent function f is

$$p(\mathbf{f}|\mathbf{x}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}), \tag{2.40}$$

and the likelihood function factorises over the data such that

$$p(\mathbf{y}|\mathbf{f}) = \prod_i^N \mathcal{N}(y_i|f_i, \sigma_y^2). \tag{2.41}$$

Using the Equations 2.40 and 2.41, the log marginal likelihood is given by

$$\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_y) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}) = -\frac{1}{2}\mathbf{y}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log(2\pi). \tag{2.42}$$

The marginal log likelihood balances between model fit and model complexity. The first term is a data fit term, the second term is a model complexity term and the third term is just a constant.

Picking the best model would be a trade-off between the first two terms and this is discussed in Murphy [2012], Section 15.2.4. To understand the trade-off, Murphy [2012] considers a RBF kernel in one-dimension, where the lengthscale parameter is allowed to vary, the signal variance parameter is kept constant at 1, and the noise variance σ_y^2 is constant. Let $J(l) = -\log p(\mathbf{y}|\mathbf{x}, l)$. For a short lengthscale the fit is good, thus $\mathbf{y}\mathbf{K}_y^{-1}\mathbf{y}$ is small. However, the model complexity would be high since $\mathbf{K}$ will be almost diagonal (due to very small terms inside the exponential function). Thus, the points are 'spread' apart, making the term $\log|\mathbf{K}_y|$ large. For a long lengthscale, the fit is poor. Thus, $\mathbf{K}$ is almost all 1's (the terms inside the exponential function are close to 0) i.e. the points are very 'close' together. Therefore, $\log|\mathbf{K}_y|$ is small and the model complexity term is low.

In order to infer the kernel parameters, the marginal likelihood is maximised by doing partial differentiation with respect to the kernel parameters θ_j

$$\begin{aligned}\frac{\partial \log p(\mathbf{y}|\mathbf{x})}{\partial \theta_j} &= \frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_j}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1})\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right), \text{ where } \boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y}.\end{aligned} \tag{2.43}$$

In the previous equation the following properties were used

$$\frac{\partial \mathbf{K}_y^{-1}}{\partial \theta_j} = -\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_j}\mathbf{K}_y^{-1}. \tag{2.44}$$

$$\frac{\partial \log|\mathbf{K}_y|}{\partial \theta_j} = \mathrm{tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right). \tag{2.45}$$

The computation times to calculate $\mathbf{K}_y^{-1}$ is $\mathscr{O}(N^3)$ and to calculate the gradient is $\mathscr{O}(N^2)$ per hyperparameters (kernel parameters). If there are constraints on the hyperparameters, then a transformation that satisfies the constraints can be used and the chain rule can be applied to compute the gradient. Since the log marginal likelihood and its derivative are available, the estimation of the kernel parameters can be done by using any gradient based optimiser or by using an MCMC sampling method, among other methods. However, highly-correlated kernel parameters might lead to slow convergence of the optimiser and inefficient MCMC sampling. To remedy this, whitening a parameter can make the optimisation or the MCMC sampling more efficient. This is achieved by taking the Cholesky decomposition of the prior covariances such that the whitened variable $\hat{\boldsymbol{\theta}} = \mathbf{L}^{-1}\boldsymbol{\theta}$, where $\mathbf{L}$ is the Cholesky decomposition matrix. The marginal likelihood is evaluated at $\mathbf{L}\hat{\boldsymbol{\theta}}$, then the chain rule is applied to compute the gradient and recover the initial parameters $\boldsymbol{\theta}$:

$$\frac{\partial \log p(\mathbf{y}|\mathbf{x})}{\partial \hat{\boldsymbol{\theta}}} = \frac{\partial \log p(\mathbf{y}|\mathbf{x})}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial \hat{\boldsymbol{\theta}}} = \mathbf{L}^T\frac{\partial \log p(\mathbf{y}|\mathbf{x})}{\partial \boldsymbol{\theta}}. \tag{2.46}$$

Limitations of Gaussian processes

One limitation that GPs have is the high computational complexity, as training the model, i.e. inverting the matrix $\mathbf{K}_y$ takes $\mathcal{O}(N^3)$ time with storage demands $\mathcal{O}(N^2)$ time. Another limitation is that for general likelihood models, the log marginal likelihood is intractable, hence methods such as MCMC are needed to compute the integral from Equation 2.39.

2.3 Relationship between Gaussian processes and other models

In this section I discuss GPs from different perspectives. Firstly, I look at GPs from a machine learning perspective, where I show how to get from a parametric linear model to a non-parametric model, in this case a GP [Bishop, 2006]. Secondly, I look at how a GP can be obtained using convolutions of continuous-time movement models [Hooten and Johnson, 2017]. Finally, I illustrate the link between GPs and state space models, and how a GP can be converted to a state space model and vice-versa [Särkkä et al., 2013, Särkkä and Hartikainen, 2012, Hartikainen and Särkkä, 2010]. The main advantage of this discussion is that I summarise and collect in one place all the ways GPs are represented in the literature in a clear and concise manner.

Original contributions are made in this chapter by offering more explanations on how to convert a covariance function to a state space model in Section 2.3.3. In addition to this, I show the full details on how to arrive at the stochastic differential equations when the covariance function is part of the Matérn class of kernels as these details are not shown in Särkkä et al. [2013]. In Hartikainen and Särkkä [2010], the authors illustrate a method on how to derive the stochastic differential equations when the kernel is squared exponential (RBF). I review that method, and offer more explanations when needed. Also, in Särkkä and Hartikainen [2012], in the Supplemental Material, Section 2, the authors use an identity (Equation 23) to deduce the stochastic differential equation when the kernel is squared exponential. Although, the context is different, given that I use a temporal model, not a spatio-temporal model as employed in Särkkä and Hartikainen [2012], I prove thoroughly this identity using induction (proof shown in Appendix, Section A in this thesis, but not shown in Särkkä and Hartikainen [2012]), and then use it to provide an alternative derivation of the stochastic differential equations when the covariance function is the squared exponential kernel.

2.3.1 Link between linear models and Gaussian processes

In this subsection, I show that from a parametric representation of a linear model I can arrive at a GP. This derivation is mainly based on Bishop [2006], Section 6.4.1. I consider a model such

that

$$f(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{2.47}$$

where $\mathbf{x}$ is the data vector, $\mathbf{w}$ is a weight vector and $\boldsymbol{\phi}(\mathbf{x})$ is a vector of functions of the data. A prior distribution is set over the weights $\mathbf{w}$ such that

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\ \mathbf{0}, \sigma^2\mathbf{I}), \tag{2.48}$$

where the hyperparameter σ is the standard deviation of the distribution. The probability distribution over the prior $p(\mathbf{w})$ induces a probability distribution over the functions $f(\mathbf{x})$ such that

$$\mathbf{f} = \boldsymbol{\Phi}\mathbf{w}, \tag{2.49}$$

where $\boldsymbol{\Phi}$ is the design matrix with elements $\Phi_{nk} = \phi_k(x_n)$. Since the weights $\mathbf{w}$ are normally distributed and $\mathbf{f}$ is a linear combination of the weight variables, $\mathbf{f}$ is also Gaussian. The mean and the variance can be found in the following way,

$$\mathbb{E}(\mathbf{f}) = \boldsymbol{\Phi}\mathbb{E}(\mathbf{w}) = 0. \tag{2.50}$$

$$\mathrm{Cov}(\mathbf{f}) = \mathbb{E}(\mathbf{f}\mathbf{f}^T) = \boldsymbol{\Phi}\mathbb{E}(\mathbf{w}\mathbf{w}^T)\boldsymbol{\Phi}^T = \sigma^2\ \boldsymbol{\Phi}\boldsymbol{\Phi}^T = \mathbf{K}, \tag{2.51}$$

where k is the kernel and $\mathbf{K}$ is the covariance matrix with the elements

$$\mathbf{K}_{nm} = k(x_n, x_m) = \sigma^2\boldsymbol{\phi}(x_n)^T\boldsymbol{\phi}(x_m). \tag{2.52}$$

A key fact of GPs is that they can be completely defined by their second-order statistics, the mean and the covariance function. Therefore, if a GP has a mean zero prior, defining the covariance function completely defines the process. This is equivalent to choosing the mean of the prior over the weights $\mathbf{w}$ to be 0.

I have shown that I arrive at a GP from a parametric model specified by the functions $\boldsymbol{\phi}(\mathbf{x})$. In practice, the kernel can be specified directly, rather than specifying first the functions $\boldsymbol{\phi}(\mathbf{x})$. As for the other direction, given a function f, then f can be parameterised to get to a parametric model.

2.3.2 Gaussian processes as stochastic process models

This subsection closely follows Hooten and Johnson [2017]. In this subsection I show that a GP can be obtained by using convolutions of continuous-time movement models. Brownian motion might be represented as an integral of white noise and this process can be written as a stochastic

integral equation using Ito's notation [Protter, 2004] as follows

$$\mathbf{b}(t_i) = \int_{t_0}^{t_i} d\mathbf{b}(\tau), \tag{2.53}$$

where $\mathbf{b}(t_i)$ is scaled multivariate Brownian motion (i.e. a multivariate Wiener process) at times t_i. Brownian motion is a stochastic process, therefore it can be integrated with respect to time to obtain a smoother process as follows

$$\boldsymbol{\eta}(t) = \int_{t_0}^{t} \mathbf{b}(\tau) d\tau, \tag{2.54}$$

where $\boldsymbol{\eta}(t)$ is a similar process to the integrated stochastic process by Johnson et al. [2008].

The velocity model in continuous-time has the following form

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu}(0) + \boldsymbol{\eta}(t), \tag{2.55}$$

where $\boldsymbol{\mu}(t_i)$ is the position at time t_i. Johnson et al. [2008] model the velocity directly as an OU process, then integrate it to get a smoother process and then substitute the integrated velocity process into Equation 2.55 to yield the position process. The framework used by Johnson et al. [2008] can be generalised by using convolutions in the form of

$$\boldsymbol{\eta}(t) = \int_{t_0}^{t_n} \mathbf{H}(t,\ \tau)\mathbf{b}(\tau) d\tau, \tag{2.56}$$

where t_n is the last time at which data are observed and the matrix $\mathbf{H}(t,\ \tau)$ is a 2×2 diagonal matrix with elements equal to the function

$$h(t, \tau) = \begin{cases} 1 & \text{if } t_0 < \tau \leq t. \\ 0 & \text{if } t < \tau \leq t_n. \end{cases} \tag{2.57}$$

This convolution is a Brownian motion velocity-based model and is part of a more general class of stochastic movement models named functional movement models (FMMs) [Hooten and Johnson, 2017]. Hooten and Johnson [2017] show in the Appendix A (Supplementary Material) (details are not shown here) that the FMMs from Equation 2.56 can be rewritten as

$$\boldsymbol{\eta}(t) = \int_{t_0}^{t_n} \tilde{\mathbf{H}}(t,\ \tau) d\mathbf{b}(\tau), \tag{2.58}$$

where $\tilde{\mathbf{H}}(t,\ \tau)$ is a diagonal matrix with elements

$$\tilde{h}(t,\ \tau) = \int_{\tau}^{t_n} h(t,\ \tilde{\tau}) d\tilde{\tau}. \tag{2.59}$$

The FMM representation from Equation 2.58 is regarded as a process convolution or a kernel

convolution [Hooten and Johnson, 2017, Calder, 2007]. Different kernels $h(t,\ \tau)$ lead to different movement models. After the smoothing kernel for Brownian motion is chosen, illustrated in Equation 2.56, Equation 2.59 is used to obtain the integrated kernel $\tilde{h}(t,\ \tau)$ that is convoluted with white noise. In one-dimension, the covariance function for the movement process $\eta(t)$ can be calculated [Hooten and Johnson, 2017, Paciorek and Schervish, 2006] as the convolution of kernels

$$\text{cov}(\eta(t_1),\ \eta(t_2)) = \int_{t_0}^{t_n} \sigma^2 \tilde{h}(t_1,\ \tau)\ \tilde{h}(t_2,\ \tau) d\tau, \tag{2.60}$$

for any two times t_1 and t_2. The covariance function shown in Equation 2.60 is positive definite and the proof of this is shown in Equation (4) in Paciorek and Schervish [2006] (the details are not shown here). From Equation 2.60, a GP can be obtained with the following representation [Hooten and Johnson, 2017]

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0},\ \sigma^2\ \Delta t\ \tilde{\mathbf{H}}\tilde{\mathbf{H}}^T), \tag{2.61}$$

where $(t_1,\ \ldots,\ t_n)$ is a finite subset of times, $\boldsymbol{\eta} = (\eta_1,\ \eta_2, \ldots,\ \eta_n)^T$, $\mathbf{0}$ is an $n \times 1$ vector of zeros and $\tilde{\mathbf{H}}$ is a matrix of basis functions with the i-th row equal to $\tilde{h}(t_i,\ \tau)$ for all τ. Translating this into a position process $\boldsymbol{\mu}$ at the observation times $(t_1,\ \ldots,\ t_n)$ results in

$$\boldsymbol{\mu} \sim \mathcal{N}(\mu(0)\mathbf{1},\ \sigma^2\ \Delta t\ \tilde{\mathbf{H}}\tilde{\mathbf{H}}^T). \tag{2.62}$$

This process can be generalised further. For two dimensions, the joint model can be written as

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0},\ \sigma^2\ \Delta t\ (\mathbf{I} \otimes \tilde{\mathbf{H}}\tilde{\mathbf{H}}^T)), \tag{2.63}$$

where $(t_1,\ \ldots,\ t_{2n})$ is a finite subset of times, $\boldsymbol{\eta} = (\eta_1,\ \eta_2, \ldots,\ \eta_{2n})^T$, $\mathbf{0}$ is an $2n \times 1$ vector of zeros, $\mathbf{I}$ is a 2×2 identity matrix and $\tilde{\mathbf{H}}$ is a basis functions matrix for both directions (longitude and latitude). Also, the symbol $\otimes$ refers to the tensor product of two vectors.

In conclusion, Equation 2.62 is equivalent to the definition for a GP. Thus, I showed that the convolution of continuous-time movements models leads to a GP.

2.3.3 Gaussian processes as state space models

In this subsection[2], I show that spatial, temporal and spatio-temporal GPs can be converted into infinite dimensional state space models and vice-versa. This idea was previously implemented by Lindgren et al. [2011], but this section is mainly based on Särkkä et al. [2013], Särkkä [2017], Särkkä and Hartikainen [2012]. The main issue with Gaussian processes is that the computational costs are high, $\mathcal{O}(N^3)$. Using SDE/SPDE (stochastic differential equations/stochastic partial differential equations) and state space models is a good alternative to GPs given that their

[2]In this subsection I will use the same notation as Särkkä et al. [2013], given that vector notation is more common in the literature when working with state space models. However, getting from a vector notation to a scalar notation can be easily done by using Equation 23 from Särkkä et al. [2013].

inference problem might be solved with Bayesian filters (e.g. Kalman filter) and smoothers with $\mathcal{O}(N)$ computation complexity [Grewal and Andrews, 2011, Cressie and Wikle, 2002, Hiltunen et al., 2011], where N is the number of data points. The downside of following the SDE/SPDE approach is that approximations to the spectral density are often used when it does not have a rational function form (for example, the squared exponential kernel does not have a rational form for its spectral density) and the mathematics might be difficult [Särkkä, 2017]. Before going into further details, I illustrate a few examples of representing a GP as a SDE in Equations 2.64-2.66

In the first example, consider the function $f(\mathbf{x})$, a spatial GP model with kernel $k(\mathbf{x},\mathbf{x}')$, which has the equivalent stochastic partial differential equation model [Särkkä, 2017] as follows

$$\mathscr{L} f(\mathbf{x}) = W(\mathbf{x}), \tag{2.64}$$

where $\mathscr{L}$ is an operator, $W(\mathbf{x})$ is a vector of white noise processes, $\mathbf{x} \in \mathbb{R}^d$.

Another example is when $\mathbf{f}$ is a temporal GP model with kernel $k(t,t')$, that has the equivalent state space/SDE formulation [Hartikainen and Särkkä, 2010] as follows

$$\frac{d\mathbf{f}(t)}{dt} = \mathbf{A}\mathbf{f}(t) + \mathbf{L}\mathbf{W}(t), \tag{2.65}$$

where $\mathbf{A}$, $\mathbf{L}$ are given matrices and $\mathbf{W}(t)$ is a vector of white noise processes.

Finally, the spatio-temporal GP $\mathbf{f}$, with kernel $k(\mathbf{x},t,\mathbf{x},t')$ has the corresponding stochastic evolution equation [Särkkä and Hartikainen, 2012]

$$\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial t} = \mathscr{A}_x \mathbf{f}(\mathbf{x},t) + \mathbf{L}\mathbf{W}(\mathbf{x},t), \tag{2.66}$$

where $\mathscr{A}_x$ is an operator, $\mathbf{L}$ is a given matrix and $\mathbf{W}$ is a vector of white noise processes.

Gaussian process regression reformulation

The GP regression problem can be rewritten in the following form

$$f(\mathbf{x}) \sim \mathscr{G}\mathscr{P}(m(\mathbf{x}),\ k(\mathbf{x},\mathbf{x}')). \tag{2.67}$$

$$\mathbf{y} = \mathscr{H} f(\mathbf{x}) + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathscr{N}(\mathbf{0},\ \boldsymbol{\Sigma}), \tag{2.68}$$

where $\mathbf{x}$ is a d-dimensional input vector, $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, $\mathscr{H} f(\mathbf{x}) = (f(\mathbf{x}_1),\ \ldots,\ f(\mathbf{x}_N))$ and the linear operator $\mathscr{K}$ is designed to select the elements of the function $\mathbf{f}$ that are observed. The above infinite-dimensional problem can be reformulated into a finite-dimensional version of a Bayesian linear regression problem [Särkkä et al., 2013] as follows

$$\mathbf{f} \sim \mathscr{N}(\mathbf{m},\mathbf{K}). \tag{2.69}$$

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathscr{N}(\mathbf{0},\boldsymbol{\Sigma}), \tag{2.70}$$

where $\mathbf{H}$ is a matrix and $\mathbf{f} = f(\mathbf{x})$.

Solving the infinite-dimensional problem GP model is analogous to solving a finite-dimensional Bayesian linear regression problem [Särkkä et al., 2013, Särkkä and Hartikainen, 2012]. This is consistent with the definition of a GP, where the GP is defined just at a finite number of points.

Bayesian filters and smoothers algorithms

As stated above, an alternative to GP regression is using Bayesian filters (e.g. Kalman filter) and smoothers, which significantly reduce the computation time [Grewal and Andrews, 2011, Cressie and Wikle, 2002, Hiltunen et al., 2011]. Suppose a state space model is of the form [Särkkä et al., 2013]

$$\frac{d\mathbf{f}}{dt} = \mathbf{A}\mathbf{f}(t) + \mathbf{L}\mathbf{W}(t). \tag{2.71}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{f}(t_k) + \boldsymbol{\varepsilon}_k, \tag{2.72}$$

where $k = 1, \ldots, T$ and $\mathbf{A}$, $\mathbf{L}$, $\mathbf{H}$ are given matrices and $\boldsymbol{\varepsilon}_k$ is a vector of Gaussian noise measurements and $\mathbf{W}(t)$ is a vector of Gaussian white noise processes. A Gaussian white noise process is a zero mean Gaussian random process, where the values of the process are uncorrelated. The vector function $\mathbf{f}(t)$ is a solution to a linear SDE (Equation 2.71) driven by Gaussian noise, therefore, $\mathbf{f}$ is a GP [Särkkä et al., 2013, Särkkä, 2017]. The solution of an SDE is a Markovian process, therefore the Kalman filter and Rauch-Tung-Striebel (RTS) smoother algorithms can be used to calculate the posterior distribution of an unobserved test point in linear time [Särkkä et al., 2013, Särkkä, 2017].

Spatio-temporal Gaussian processes representation as a state space model

Spatio-temporal GP regression is used with models of the following form [Särkkä et al., 2013, Särkkä, 2017]

$$f(\mathbf{x}, t) \sim \mathcal{GP}(0,\ k(\mathbf{x}, t; \mathbf{x}', t')). \tag{2.73}$$

$$\mathbf{y}_k = \mathscr{H}_k f(\mathbf{x}, t_k) + \boldsymbol{\varepsilon}_k. \tag{2.74}$$

The corresponding infinite-dimensional state space model [Särkkä et al., 2013, Särkkä, 2017] is

$$\frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial t} = \mathscr{A}_x \mathbf{f}(\mathbf{x}, t) + \mathbf{L}\mathbf{W}(\mathbf{x}, t). \tag{2.75}$$

$$\mathbf{y}_k = \mathscr{H}_k \mathbf{f}(\mathbf{x}, t_k) + \boldsymbol{\varepsilon}_k, \tag{2.76}$$

where $\mathscr{A}_x$ and $\mathscr{H}_k$ are linear operators.

This model is an infinite-dimensional Markovian model, thus, linear time inference of the

function $\mathbf{f}$ is possible by using the infinite-dimensional Kalman filter and RTS smoother [Särkkä et al., 2013, Särkkä and Hartikainen, 2012, Särkkä, 2017]. As for the spatial and temporal GPs, the representations as state space models are shown in Equations 2.64 and 2.65. Linear time inference is possible in these cases using the same algorithms by following the same procedure as above.

Gaussian processes as solutions to linear SDEs

In this subsection I show that GPs can be constructed as solutions to n-th order stochastic linear differential equations of the following form [Särkkä et al., 2013, Särkkä, 2017]

$$a_n \frac{d^n f(t)}{dt^n} + \cdots + a_1 \frac{df(t)}{dt} + a_0 f(t) = W(t), \tag{2.77}$$

where $W(t)$ is a white noise GP with mean zero. As before, the solution $f(t)$ is a GP, because $W(t)$ is a GP (the solution $f(t)$ of a linear differential equation is also Gaussian because $W(t)$ is Gaussian, as it is a linear operation on the input) [Hartikainen and Särkkä, 2010, Särkkä, 2017].

If $\mathbf{f} = \left(f,\ \frac{df}{dt},\ \ldots,\ \frac{d^{N-1}f}{dt^{N-1}}\right)$, then a space state model of the form is obtained

$$\frac{d\mathbf{f}}{dt} = \mathbf{A}\mathbf{f} + \mathbf{L}W(t). \tag{2.78}$$

$$f(t) = \mathbf{H}\mathbf{f} + \varepsilon, \tag{2.79}$$

where

$$A = \begin{pmatrix} 0 & 1 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & \ldots & -a_{n-2} & -a_{n-1} \end{pmatrix},$$

and $H = (1\,0 \ldots 0)$, $\mathbf{L} = (0\,0 \ldots 1)^T$. The vector process $\mathbf{f}(t)$ is Markovian, although $f(t)$ is generally not. Thus, a model of the same form as before is obtained and linear-time inference is performed for the function $\mathbf{f}$ using a Kalman filter or smoother algorithms [Särkkä et al., 2013, Särkkä, 2017].

Converting from state space models to covariance functions

In this subsection I show how to get from the space state model formulation to the corresponding covariance function [Särkkä et al., 2013, Särkkä, 2017]. If the Fourier transform of Equation 2.77 is taken and solved for $F(w)$, the following equation is obtained

$$F(w) = \left(\frac{1}{a_n(iw)^n + \cdots + a_1(iw) + a_0}\right) W(w) = G(iw)\, W(iw), \tag{2.80}$$

where $G(iw) = \left(\frac{1}{a_n(iw)^n + \cdots + a_1(iw) + a_0}\right)$ is the transfer function and $W(iw)$ is the Fourier transform of the white noise process. From the table of Fourier transform pairs from time domain to frequency domain I used the following transformation

$$\frac{d^n(f(t))}{dt^n} \rightarrow (iw)^n F(w). \tag{2.81}$$

The spectral density of the process can be calculated by squaring the absolute value of the Fourier transform of the process. Thus, the spectral density of the process is

$$S(w) = |W(iw)|^2 |G(iw)|^2. \tag{2.82}$$

The spectral density of the white noise process is a constant[3], hence the spectral density is of the following form

$$S(w) = \frac{\text{constant}}{\text{polynomial in } w^2}. \tag{2.83}$$

Therefore, the spectral density of Equation 2.77 is a rational function. By using the classical Wiener–Khinchin theorem, calculating the inverse Fourier transform of the spectral density gives the stationary covariance function

$$C(t) = F^{-1}[S(w)] = \frac{1}{2\pi} \int S(w) \exp(iwt) dw. \tag{2.84}$$

Finally, the corresponding covariance function is

$$k(t,t') = C(t-t'). \tag{2.85}$$

Converting from a covariance function to a state space model

A state space model can be formulated given a covariance function as follows [Särkkä et al., 2013, Hartikainen and Särkkä, 2010, Särkkä, 2017]

- Firstly, the spectral density $S(w)$ is calculated by computing the Fourier transform of the covariance function $C(t)$.
- If $S(w)$ is not a rational function, then an approximation using Taylor series expansions or Padé approximants is formed.
- Factorisation into stable and unstable parts using spectral factorisation is done such that

$$S(w) = H(iw) q_c H(-iw), \tag{2.86}$$

[3]White noise is noise that has equal intensity at different frequencies, therefore the spectral density is constant and independent of frequency. It is called white noise because of its similarities to white light, which has equal quantities of all colors [Mancini, 2003].

where q_c is the spectral density of the white noise process. The transfer function $H(iw)$, a function of iw, is stable and rational if and only if its roots (the zeros of the denominator) are in the upper half plane. The zeros of the numerator should also be in the upper plane for the transfer function to be in the minimum phase.

- Using methods from control theory [Glad and Ljung, 2000], the more general stochastic evolution equations is obtained[4]

$$d\mathbf{f}(\mathbf{x},t) = \mathbf{A}\mathbf{f}(\mathbf{x},t)dt + \mathbf{L}d\mathbf{W}(\mathbf{x},t), \tag{2.87}$$

where

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ -\mathscr{A}_0 & -\mathscr{A}_1 & \dots & -\mathscr{A}_{m-2} & -\mathscr{A}_{m-1} \end{pmatrix}$$

is a matrix of linear operators, $\mathbf{W}(\mathbf{x},t)$ is a Hilbert space valued Wiener process, and $\mathbf{L} = (0\,0\dots1)^T$. Moreover, the linear operators $\mathscr{A}_j$ are defined as

$$\begin{aligned} \mathscr{A}_0 &= \mathscr{F}_x{}^{-1}\left[a_0(iw_x)\right]. \\ \mathscr{A}_1 &= \mathscr{F}_x{}^{-1}\left[a_1(iw_x)\right]. \\ &\vdots \\ \mathscr{A}_{m-1} &= \mathscr{F}_x{}^{-1}\left[a_{m-1}(iw_x)\right], \end{aligned}$$

where $\mathscr{F}_x{}^{-1}$ is the inverse Fourier transform. Finally, $a_0,\dots a_{m-1}$ are the coefficients of the rational function form of the spectral density $S(w)$, similar to the coefficients a_i found in Equation 2.80.

More explicitly, the method called spectral factorisation is used to find the transfer function $H(iw)$. The method consists in the following steps

- Compute the roots of the numerator and denominator of $S(w)$. Given that $S(w)$ is a polynomial in w^2, i.e. it has even degree, the roots will come in pairs and be complex conjugates of one another.

- Construct $H(iw)$ from the positive-imaginary-part roots only or from the negative-real-part roots only[5].

[4] More details of how to get to the evolution equation are shown below alongside detailed examples.

[5] In Särkkä et al. [2013] there is no mention of using the negative-real-part roots, however in Särkkä and Hartikainen [2012], Example 4.1, the authors use the negative-real-part roots only. Also, in Hartikainen and Särkkä [2010], Section 4.2, negative-real-part roots were used to form the transfer function. Thus, I added this part in.

- Using the stable function $H(iw)$ construct a stable Markov process, which leads to the following frequency domain representation of the process[6]

$$(iw)^m F(w) + h_{m-1}(iw)^{m-1} F(w) + \cdots + h_0 F(w) = W(w),$$

where $W(w)$ and $F(w)$ are the formal Fourier transforms of $W(t)$ and $f(t)$, and h_0, h_1, ... , h_{m-1} are the coefficients of the polynomial in the denominator of $H(iw)$.

- The Markov representation in the time domain is

$$\frac{d^m f(t)}{dt^m} + h_{m-1}\frac{d^{m-1} f(t)}{dt^{m-1}} + \cdots + h_1 \frac{df(t)}{dt} + h_0 f(t) = W(t),$$

which is the desired form. Getting to the frequency domain representation can easily be done using Equation 2.81.

- It is important to remark that if the spectral density does not have a rational form, using approximations of $S(w)$ will only lead to approximate covariance functions.

I show multiple examples converting from a covariance function to the corresponding linear differential equations. The full details are not shown in Särkkä et al. [2013], however I derive the full details here. The class of kernels of interest is the Matérn class. The covariance function of the Matérn familiy in one-dimension is given by

$$k(t,t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{|t-t'|}{l} \right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{|t-t'|}{l} \right), \tag{2.88}$$

where $\nu, \sigma, l > 0$ are the smoothness, magnitude and lengthscale parameters, $K_\nu(\cdot)$ is the modified Bessel function of the second kind and $\Gamma(\cdot)$ is the Gamma function.

To define the spectral density I first state the Bochner's theorem [Rasmussen and Williams, 2006] as follows

Theorem 1 *A complex-valued function k on $\mathbb{R}^D$ is the covariance function of a weakly stationary mean square continuous complex-valued random process on $\mathbb{R}^D$ if and only if it can be represented as*

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}\cdot\boldsymbol{\tau}} d\mu(\mathbf{s}), \tag{2.89}$$

where μ is a positive finite measure.

From Rasmussen and Williams [2006], a process with a constant mean and with an invariant variance to translations is called weakly stationary. Moreover, a strictly stationary process has

[6]This bullet point and the next are not found in Särkkä et al. [2013], however I added them in according to Hartikainen and Särkkä [2010], due to lack of clear explanations in Särkkä et al. [2013] on how to actually derive the transfer function and the stochastic differential equation. Examples in Särkkä et al. [2013] are also very brief and skip the details.

all of its finite dimensional distributions invariant to translation. Moreover, a process X is called mean square continuous if

$$\mathbb{E}(X_t^2) < +\infty, \tag{2.90}$$

$$\lim_{s \to t} \mathbb{E}\left[|X_s - X_t|^2\right] = 0. \tag{2.91}$$

If the spectral density $S(\mathbf{s})$ exists, the Wiener-Khintchine theorem states that the covariance function and the spectral density are Fourier duals of each other [Rasmussen and Williams, 2006], Equation 4.6 such that

$$k(\boldsymbol{\tau}) = \int S(\mathbf{s}) e^{2\pi i \mathbf{s} \cdot \boldsymbol{\tau}} d\mathbf{s}. \tag{2.92}$$

$$S(\mathbf{s}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \mathbf{s} \cdot \boldsymbol{\tau}} d\boldsymbol{\tau}. \tag{2.93}$$

The spectral density of the Matérn class of kernels [Rasmussen and Williams, 2006] is

$$S(s) = \sigma^2 \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu) l^{2\nu}} \left(\frac{2\nu}{l^2} + 4\pi^2 s^2 \right)^{(-\nu + D/2)}, \tag{2.94}$$

where D is the number of dimensions. In one-dimension the above term can be simplified such that

$$S(w) = \sigma^2 \frac{2\pi^{1/2} \Gamma(\nu + 1/2)}{\Gamma(\nu)} \lambda^{2\nu} \left(\lambda^2 + w^2 \right)^{(-\nu + 1/2)}, \tag{2.95}$$

where $\lambda = \frac{\sqrt{2\nu}}{l}$ and $4\pi^2 s^2 = w^2$ (a simple transformation to angular frequency notation). From the equation above the spectral density is proportional to

$$S(w) \propto (\lambda^2 + w^2)^{-(\nu + \frac{1}{2})}. \tag{2.96}$$

Keeping in mind that $i^2 = -1$, the spectral density is factorised into

$$S(w) \propto (\lambda + iw)^{-(p+1)} (\lambda - iw)^{-(p+1)}, \tag{2.97}$$

where $\nu = p + \frac{1}{2}$ and p is a non-negative integer. This function is a rational function in w^2, therefore the transfer function $H(iw)$ of the corresponding stable Markov process exists and has the following form

$$H(iw) = (\lambda + iw)^{-(p+1)}. \tag{2.98}$$

Using Equations 2.95-2.98 and Equation 2.86, the corresponding white noise process spectral density is

$$q_c = \frac{2\sigma^2 \pi^{1/2} \lambda^{2p+1} \Gamma(p+1)}{\Gamma(p+1/2)}. \tag{2.99}$$

For integer values of p, Matérn 1/2 ($p = 0$), Matérn 3/2 ($p = 1$), Matérn 5/2 ($p = 2$) or the

squared exponential kernels ($p \to \infty$) are obtained. My goal is to deduce all the stochastic differential equations in all of these cases. I start by deriving the SDE for $p = 0$ (Matérn 1/2 kernel). Following the procedure from Section 2.3.3 and using Equation 2.97 the transfer function is

$$H(iw) = (\lambda + iw)^{-1} = \frac{1}{\lambda + iw}. \tag{2.100}$$

Then, the polynomial frequency domain representation of the process is

$$(iw)^1 F(w) + \lambda F(w) = W(w), \tag{2.101}$$

where $m = 1$ and $\lambda = h_0$. Hence, the SDE representation in the time domain is the following

$$\frac{df(t)}{dt} + \lambda f(t) = W(t). \tag{2.102}$$

The solution to this SDE is shown in Chapter 4, Section 4.2.2.

For $p = 1$, I get

$$H(iw) = (\lambda + iw)^{-2} = \frac{1}{(iw)^2 + 2\lambda(iw) + \lambda^2}. \tag{2.103}$$

Then, the polynomial frequency domain representation of the process is

$$(iw)^2 F(w) + h_1(iw)F(w) + h_0 F(w) = W(w). \tag{2.104}$$

Therefore, the SDE formulation for the Matérn 3/2 process is

$$\frac{df^2(t)}{dt^2} + 2\lambda\frac{df(t)}{dt} + \lambda^2 f(t) = W(t). \tag{2.105}$$

For $p = 2$, I have

$$H(iw) = (\lambda + iw)^{-3} = \frac{1}{(\lambda + iw)^3} = \frac{1}{(iw)^3 + 3\lambda(iw)^2 + 3\lambda^2(iw) + \lambda^3}. \tag{2.106}$$

From the previous equation the coefficients that are needed can be identified, $m = 3$, $h_2 = 3\lambda$, $h_1 = 3\lambda^2$ and $h_0 = \lambda^3$. Therefore, the SDE representation for the Matérn 5/2 kernel is

$$\frac{df^3(t)}{dt^3} + 3\lambda\frac{df^2(t)}{dt^2} + 3\lambda^2\frac{df(t)}{dt} + \lambda^3 f(t) = W(t). \tag{2.107}$$

The derivations of the Matérn 3/2 and 5/2 SDE are not explicitly shown in Rasmussen and Williams [2006], but the final solutions are shown in Rasmussen and Williams [2006], Equation 4.17. To solve the SDEs for Matérn 3/2 and 5/2 processes, Laplace transformations [Särkkä and Solin, 2019], Section 2.5 can be used. Alternatively, an easy solution to find the function f is to use Fourier transformations as in Särkkä and Solin [2019], Example 2.2.

Now, the aim is to derive the stochastic differential equation when $\nu \to \infty$ (or p $\to \infty$). In this case the kernel is called the squared exponential or RBF kernel. The one-dimensional squared exponential covariance function has the following form

$$k(t,t') = \sigma^2 \exp\left(-\frac{(t-t')^2}{2l^2}\right). \tag{2.108}$$

Performing the Fourier transform of $k(t,t')$, the spectral density is

$$\begin{aligned} S(w) = \int k(t,t+\tau)\exp(-iwt)d\tau &= \int \sigma^2 \exp\left(-\frac{\tau^2}{2l^2}\right)\exp(-iw\tau)d\tau \\ &= \sigma^2\sqrt{2\pi}l\exp\left(-\frac{l^2w^2}{2}\right), \end{aligned} \tag{2.109}$$

where the Fourier transformation table is used to get to the last equation. The spectral density $S(w)$ of the RBF kernel is not a rational function, therefore I use a Taylor series approximation to approximate it. As a reminder, the Taylor series approximation formula is

$$f(x) = f((x-a)+a) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \ldots, \tag{2.110}$$

where in general $f^{(k)}(a)$ is the k-th derivative of f evaluated at the point a. Using the Taylor Series expansion I get that

$$\exp\left(\frac{l^2w^2}{2}\right) = \exp\left(\frac{w^2}{4k}\right) \approx 1 + \frac{w^2}{4k} + \frac{1}{2!}\frac{w^4}{(4k)^2} + \cdots + \frac{1}{N!}\frac{w^{2N}}{(4k)^N}, \tag{2.111}$$

where I denoted $k = \frac{1}{2l^2}$. From the previous expression I get that[7]

$$S(w) \approx \sigma^2\sqrt{\frac{\pi}{k}}\frac{1}{1 + \frac{w^2}{4k} + \frac{1}{2!}\frac{w^4}{(4k)^2} + \cdots + \frac{1}{N!}\frac{w^{2N}}{(4k)^N}}. \tag{2.112}$$

I want the coefficient in front of the leading term w^{2N} to be 1, therefore I factorise the coefficient of w^{2N} to get

$$S(w) = \sigma^2 N!(4k)^N\sqrt{\frac{\pi}{k}}\left(\frac{1}{\sum_{n=0}^{N}\frac{N!(4k)^{N-n}}{n!}w^{2n}}\right). \tag{2.113}$$

The spectral density is now of the desired rational form. I next calculate the transfer function

[7] In Särkkä et al. [2013], in Example 2, the authors seem to have forgotten a coefficient of $\frac{1}{2^i}$ in front of the i-th term in the sum series.

$H(iw)$. For simplicity I assume N is even (so that the leading coefficient is 1). I denote

$$P(iw) = \sum_{n=0}^{N} \frac{N!(-1)^n(4k)^{N-n}}{n!} w^{2n}. \tag{2.114}$$

The transfer function $H(iw)$ is formed by following this procedure

1. Calculate numerically the roots of the polynomial $P(x)$.

2. $P(x)$ is an even degree polynomial with real coefficients, therefore the roots will be complex conjugates and will come in pairs. Let $P^-(x)$ be the polynomial formed by the solutions with negative real parts, and $P^+(x)$ the polynomial formed by the roots with positive real parts. Then,
$$P(x) = P^-(x)P^+(x).$$

3. The transfer function is then $H(iw) = \frac{1}{P^-(iw)}$ and the corresponding white noise spectral density is $q_c = \sigma^2 N!(4k)^N \sqrt{\frac{\pi}{k}}$. This procedure results in $S(w) = H(iw)q_c H(-iw)$.

The derivation so far for the RBF kernel SDE representation has been reviewed using Hartikainen and Särkkä [2010], and I added some explanations where the reader might get stuck. In Figure 2 of Hartikainen and Särkkä [2010], the authors show that for $N = 6$, the approximate spectral density and covariance functions are almost identical to the true ones. Moreover, in Hartikainen and Särkkä [2014], the authors prove rigorously that the approximate spectral converges to the true density as $N \to \infty$ and that the corresponding covariance function converges to the true covariance function, shown in Equations 6 and 7 in Hartikainen and Särkkä [2014].

There is another way to derive the stochastic differential equation of a RBF process. The following identity can be used [8] to derive the transfer function $H(iw)$ given the spectral density formula in Equations 2.112

$$\frac{1}{a_0 + a_1(iw)^2 + \cdots + (a_N)(iw)^{2N}} = \frac{1}{b_0 + b_1(iw) + \cdots + b_N(iw)^N} \times \frac{1}{b_0 + b_1(-iw) + \cdots + b_N(-iw)^N}. \tag{2.115}$$

The first term of the right hand side term is the transfer function $H(iw)$. Using the identity above, $S(w) = H(iw)q_c H(-iw)$, which is the form needed to form the SDE representation of the process. Using the coefficients of $H(iw) = \frac{1}{b_0 + b_1(iw) + \cdots + b_n(iw)^N}$, the Markov representation in the time domain is

$$b_N \frac{df^{(N)}}{dt^N} + b_{N-1} \frac{df^{(N-1)}}{dt^{N-1}} + \cdots + b_1 \frac{df^{(1)}}{dt^1} + b_0 f(t) = W(t), \tag{2.116}$$

where $W(t)$ is the white noise process.

[8] The proof is shown in the Appendix, Section A in Särkkä et al. [2013].

It is not entirely obvious why the identity in Equation 2.112 holds. I discuss here in more detail. The identity can be rewritten as

$$\begin{aligned}\frac{1}{a_0+a_1(iw)^2+\cdots+(a_N)(iw)^{2N}} &= \frac{1}{b_0+b_1(iw)+\cdots+b_N(iw)^N} \\ &\times \frac{1}{b_0+b_1(-1)^1(iw)+\cdots+b_N(-1)^N(iw)^N}.\end{aligned} \tag{2.117}$$

By doing the multiplication in the RHS term it can be easily seen why the odd powers of (iw) disappear. The formula for a coefficient a_k is

$$a_k = b_0 b_{2k} + \cdots + b_k^2(-1)^k + b_{k+1}b_{k-1}(-1)^{k-1} + \cdots + b_{2k}b_0, \tag{2.118}$$

using all the possible combinations of the terms that can be used to get to the power of $2k$.

It is known from the previous method that a large N is not needed to have an almost identical spectral density and covariance function. A value of $N = 6$ is sufficient, and hence for a value of $N = 6$ I get

$$a_0 = b_0^2. \tag{2.119}$$
$$a_1 = 2b_2b_0 - b_1^2. \tag{2.120}$$
$$a_3 = 2b_4b_0 - 2b_1b_3 + b_2^2. \tag{2.121}$$
$$a_4 = 2b_0b_6 - 2b_1b_5 + 2b_2b_4 - b_3^2. \tag{2.122}$$
$$a_5 = b_4b_6 - b_5^2. \tag{2.123}$$
$$a_6 = b_6^2. \tag{2.124}$$

Given that $a_i = \frac{N!(-1)^i(4k)^{N-i}}{i!}$, the system of equations above can be solved to get the coefficients b_i's to form the SDE representation when the covariance function is the RBF kernel. The previous approach gives a numerical way to find the transfer function $H(iw)$, given that that the roots of $P(x)$ get computed numerically, however this approach gives exact results for the coefficients b_i's.

In summary, I showed in this subsection that a GP can be represented as a state-space model [Särkkä et al., 2013, Hartikainen and Särkkä, 2010, Lindgren et al., 2011, Särkkä, 2017] in order to reduce the high computational costs to linear-time inference by using Kalman filters and smoothers algorithms [Grewal and Andrews, 2011, Cressie and Wikle, 2002, Hiltunen et al., 2011]. Also, the inverse transformation is possible and a state space model can be represented as a GP [Särkkä et al., 2013, Särkkä, 2017]. Moreover, I showed in Section 2.3.3 that GPs arise as solutions to linear stochastic differential equations. The full details are not shown in Särkkä et al. [2013], however using Särkkä et al. [2013], Hartikainen and Särkkä [2010], Särkkä and Hartikainen [2012], I illustrated the full derivation on how to get the stochastic differential

equations for the Matérn class. The advantage of representing a GPs as a state space model is that they are a computationally cheaper alternative to GPs and I showed briefly how this might be done. The downsides are that approximations are often needed to compute the spectral factorisation and the mathematics might be difficult [Särkkä, 2017].

Conclusions

In this section, I discussed GPs from three different perspectives. Firstly, from the machine learning perspective, I illustrated how a parametric model is obtained from a non-parametric model, in this case a GP. Secondly, from the ecology perspective, I demonstrated that convolutions of continuous-time movement models can yield GPs. Finally, I showed that GPs are solutions to linear stochastic differential equations and can be represented as a state space model and vice-versa. The strength and the main contribution of this chapter lie not in its distinct sections, which can be found scattered across the literature, but as a whole, connecting the dots, and explaining to a lay reader the bigger picture of GPs. I also showed the full details of how to derive the stochastic differential equations for the Matérn class of kernels and offered further explanations when I considered that it was necessary.

2.4 Non-stationary Gaussian processes

In a standard GP setting, the three key parameters: lengthscale, signal variance (amplitude) and noise variance are constant and are not input-dependent. Stationary GPs lack the flexibility to model data that presents various degrees of non-stationarity [Paciorek and Schervish, 2004, Gibbs, 1997, MacKay, 1997]. In this case a non-stationary GP, where all or a subset of the parameters: lengthscale, signal variance and noise variance vary might be more suitable for this type of data. In such a scenario, the analytical posterior of the GP becomes intractable [Bachelir, 1900, Tolvanen et al., 2014]. For example, non-stationarities in the GP have been introduced in Paciorek and Schervish [2004], Gibbs [1997], where the lengthscale parameter is input-dependent, or in the signal variance and/or in the noise parameter in Kersting et al. [2007], Tolvanen et al. [2014].

Non-stationarity in a GP model can be introduced through the use of valid non-stationary kernels (examples can be found in Rasmussen and Williams [2006], Chapter 4, one particular example is a neural network kernel). Another approach is to use a hierarchical GP model [Heinonen et al., 2016, Tolvanen et al., 2014], where all or a subset of the GP parameters are modelled by other GPs. The latter has multiple advantages over the former, including increased flexibility, as the covariance kernels of the second-layer of the GPs model determine the smoothness and structure of the first-layer parameters. Another advantage is that given the hierarchical structure and the ability to choose the GP priors on all the layers, the modeller's control

is increased, thus increasing the applicability potential to real life applications and the interpretability of the GP parameters. A recent example is Torney et al. [2021], where the authors use a hierarchical GP model to learn time-varying movement parameters with periodic (seasonal and diurnal) structure. The covariance function of the first layer of the GPs needs to be a valid non-stationary kernel, and in this regard, the technique that can transform any valid stationary covariance function into a valid non-stationary covariance function developed by Paciorek and Schervish [2004] can be used with great effectiveness.

Another remark is that the covariance function of a non-stationary GP will depend on the actual datapoints. For example, suppose that the covariance function is non-stationary in the lengthscale parameter l, but stationary for the other parameters: the noise variance ω^2, the signal variance σ^2, and that there are n datapoints x_i. Therefore, there are n l_i's for each value of x_i, while the other parameters are kept at constant value.

Similar to the hierarchical model developed by Heinonen et al. [2016] and used in this thesis, in Chapters 5 and 6, is the deep Gaussian process (DGP) model, a multi-layer generalisation of a GP, where the prior is defined recursively on multiple stochastic functions [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017, Wang et al., 2016]. The difference between the hierarchical model introduced by Heinonen et al. [2016] and the DGP model is that in the latter, independent GP priors are set on each stochastic function, not on the parameters of the GP parameters. Mathematically, a DGP model resembles a composition of multivariate functions.

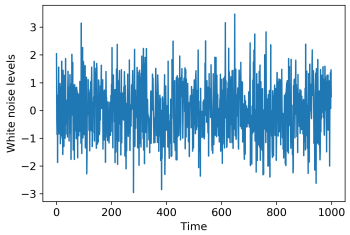
Across the literature there are other attempts to account for the non-stationarity in the data such as Tresp [2001], where the author uses a mixture of GPs, called a MGP model. This approach has the advantage that it uses arbitrary local GP kernels, however it does not guarantee function continuity over GP kernel transitions. Contrary to this, Paciorek and Schervish [2004] develop a non-stationary GP model which guarantees function continuity at the borders, however the local stationary kernels belong to the same family of kernels.

From an ecology perspective, in order to define more realistic animal movement models, allowing for the possibility for the animals to switch between several behavioural states is necessary. For example, these behavioural states might include 'encamped', where you have short step-lengths and low directional persistence or 'exploratory', where you have long step-lengths and high directional persistence [Morales et al., 2004]. To address this, in Chapter 5, we employ a non-stationary hierarchical GP model [Heinonen et al., 2016] that allows for the inference of continuous latent behavioural states. However, in situations where there are sharp transitions between states, a MGP model can be a better alternative [Tresp, 2001] than the hierarchical model. In contrast, the latter model incorporates smooth transitions between states more effectively than the former model.

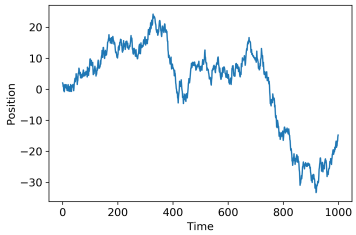
2.5 Ornstein-Uhlenbeck process

Most of the movements of animals models are derived from simple random walk processes. The studies of the irregular motion of individual particles by the botanist Brown can be considered the foundation of random walk theory [Brown, 1828]. This irregular motion of particles is now known as Brownian motion and has smoother trajectories than white noise because it is an integrated quantity (integral of a white noise GP). Brownian motion is not a flexible model for movement since it lacks drift and attraction components, however it is often used as a basic model for animal movement in continuous time [Turchin, 1998]. The Ornstein–Uhlenbeck process (named after Leonard Ornstein and George Eugene Uhlenbeck), is a stochastic process that describes the velocity of a massive Brownian particle under the influence of friction [Uhlenbeck and Ornstein, 1930]. The process is a modification of the random walk in continuous time (Wiener process) i.e. the OU process is a Brownian motion process that has attraction to a point (the mean). The process is called mean-reverting in the sense that over time, the process tends to drift towards its long-term mean with a greater attraction when the position of the particle is further away from the center.

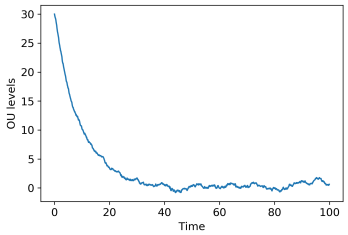
The OU model is a basic mean reversion model that has applications in areas such as biology or finance [Oksendal, 1998, Shreve, 2004, Lande, 1976, Wiens et al., 2010]. The OU process is widely used in biology modelling neuronal responses [Cain et al., 2013], whereas in mathematical finance it can be used to model the change in the interest rates and in the asset prices [Barndorff-Nielsen and Shephard, 2001a, Kluppelberg et al., 2007, Vasicek, 1977]. For example, the Vasicek model is an Ornstein-Uhlenbeck process that has been used to capture the dynamics of the short-term interest rate in the market [Vasicek, 1977]. It is worth noting that not all continuous-time models are based on OU processes i.e. there are continuous-time random walk models that do not have the mean-reversion property. Examples include GPs with different kernels than the OU kernel (Matérn 1/2 kernel), such as SE kernel or RBF kernel. Other examples include models based on potential functions [Morales et al., 2004, Brillinger, 2010]. I illustrate the Brownian motion process, white noise GP and the OU process in Figures 2.2b, 2.2a and 2.2c.



(a) Plot of a white noise process.



(b) Plot of Brownian motion.



(c) Plot of an OU process with an attraction point at 0. The initial value is 30. The mean reversion rate a is 0.1 and $\sigma = 0.3$.

Figure 2.2: White noise, Brownian motion and OU processes plots comparison.

Simulating an OU process

The stochastic differential equation (SDE) for an OU process is

$$dx_t = a(b - x_t)dt + \sigma dW_t, \tag{2.125}$$

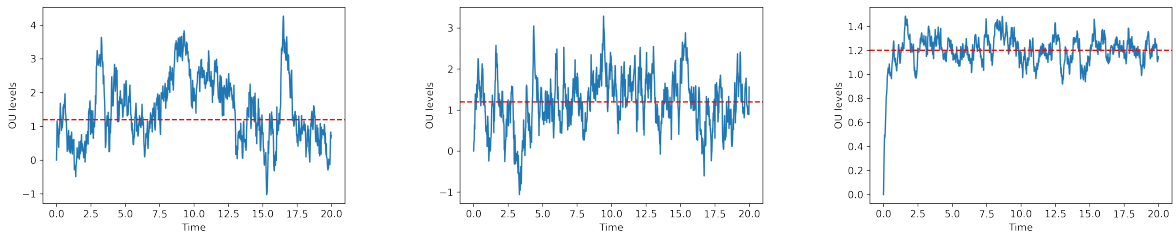
where W_t is a Wiener process, a is the rate at which the process mean reverts (a larger number results in a faster mean reverting process), b is the run average and σ is the volatility of the process. This SDE can be discretised and approximated using Euler–Maruyama method as follows

$$x_{n+1} = x_n + a\,(b - x_n)\,\Delta t + \sigma \Delta W_t, \tag{2.126}$$

where ΔW_t are independent and identically distributed Wiener increments, i.e. normal variables with zero mean and variance Δt. Thus

$$W_{t_{n+1}} - W_{t_n} = \Delta W_n \sim \mathcal{N}(0, \Delta t) = \sqrt{\Delta t}\ \mathcal{N}(0,1). \tag{2.127}$$

In Figure 2.3, I simulate multiple OU processes, where $\Delta t = 0.02$, $t = [0, 20]$, $b = 1.2$ and the initial starting point $x_0 = 0$.



(a) Plot of an OU process with $a = 1$ and $\sigma = 2$.

(b) Plot of an OU process with $a = 5$ and $\sigma = 2$.

(c) Plot of an OU process with $a = 5$ and $\sigma = 0.3$.

Figure 2.3: Multiple OU processes with various coefficients where the mean $b = 1.2$ (red dashed line).

Figures 2.3a and 2.3b have the same value for σ, but in the latter the OU levels revert faster due to higher rate a. By comparing Figures 2.3b and 2.3c, the effects of increasing σ in the first plot are noticeable due to increase in vertical variation. Since the initial starting point is close to the mean b, the noise around the mean b is more visible in Figure 2.3 than Figure 2.2c.

Link between the parameters of the OU process and the parameters of the Matérn 1/2 kernel

In one-dimension the Matérn 1/2 kernel and the OU covariance function (derivation of the OU kernel will be given in Chapter 4) have the following form

$$k(x_s, x_t) = \text{kernel variance} \times \exp\left\{\frac{-|s-t|}{\text{kernel lengthscale}}\right\}. \tag{2.128}$$

$$\text{Cov}(x_s, x_t) = \frac{\sigma^2}{2a}\exp(-a|s-t|). \tag{2.129}$$

Equating these last two equation gives

$$\text{kernel variance} = \frac{\sigma^2}{2a}. \tag{2.130}$$

$$\text{kernel lengthscale} = \frac{1}{a}, \tag{2.131}$$

where a is mean-reversion rate and σ is the volatility of the OU process. The last two equations establish a relationship between the parameters of the OU process and the parameters of the Matérn 1/2 kernel.

2.6 Bayesian inference and algorithms

In this section I only review the relevant methods and tools used in this thesis, however the cited literature in the corresponding section should provide wider information background.

2.6.1 Sampling methods

Introduction to Markov Chain Monte Carlo

The main sources for this section are: Chapter 24 from Murphy [2012], Section 11.5 from Bishop [2006], Chapter 10 from Gelman et al. [2013] and Neal [1992].

In a Bayesian setting usually the main interest is the computation of the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ and the computation of the posterior predictive distribution $p(\mathbf{y}^*|\mathbf{y})$, where $\boldsymbol{\theta}$ is the vector parameter of interest, $\mathbf{y}$ is the observed data and $\mathbf{y}^*$ is a predictive dataset. The posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ in some simple cases can be computed analytically in closed form and if it is a known distribution one can sample from it, however for more complicated, unknown models or for high dimensional models more complex methods are needed to sample from the posterior distribution [Gelman et al., 2013].

Numerical integration methods, also referred as 'quadrature' methods are a group methods that compute an integral over continuous functions at a finite set of points [Gelman et al., 2013]. This class of methods can be divided into two branches, one being the stochastic approach which makes use of methods such as Monte Carlo methods (rejection sampling and importance sampling) and the other one being the deterministic approach where quadrature rule methods can be used [Gelman et al., 2013].

Suppose the aim is to sample from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$, which is the target distribution. One possible way to sample from this distribution is to use Markov Chain Monte Carlo methods (MCMC). MCMC methods are simulation (stochastic) methods, that are based on obtaining random draws $\boldsymbol{\theta}^s$ from the target distribution $p(\boldsymbol{\theta}|\mathbf{y})$, and then calculating the integral (expectation of any function $h(\boldsymbol{\theta})$)

$$\mathbb{E}(h(\boldsymbol{\theta})) = \int h(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \approx \frac{1}{S} \sum_{s=1}^{S} h(\boldsymbol{\theta}^s). \tag{2.132}$$

The accuracy of the simulation can be improved by drawing more samples from the target distribution. While it is easy to sample from known distributions, more complex methods need to be constructed in order to sample from unknown posterior distributions. A Markov chain is constructed by sampling step by step with the distribution of the sampled draws depending only on the last value drawn. The approximate conditional distribution needs to reach the equilibrium distribution and converge to the stationary distribution (our target distribution). Once convergence occurs, the samples drawn can be considered as actual samples from the desired target distribution [Gelman et al., 2013]. Checking for convergence is essential and this is done either by visual inspection (traceplots) or by formal calculations of convergence diagnostics such as Gelman-Rubin statistic [Gelman and Rubin, 1992] or Geweke diagnostic [Geweke, 1992].

It is worth noting that other sampling algorithms such as rejection sampling or importance sampling have limitations especially in higher dimensions, such as a high rejection rate. How-

ever, MCMC methods perform much better in a more general framework (higher dimensions), allowing sampling from a large class of distributions [Gelman et al., 2013]. In the next sections I present two of the algorithms used in this thesis: the Metropolis-Hastings algorithm and the Hamiltonian Monte Carlo (HMC) algorithm. I also review some convergence diagnostics and in the last section I discuss an alternative to MCMC, that is variational inference methods.

Metropolis-Hastings algorithm

This subsection is mainly based on Gelman et al. [2013], Section 11.2. The Metropolis-Hastings (MH) algorithm is the basic MCMC method and it can be used to draw samples from a probability distribution that might be difficult to draw samples from directly. The algorithm generates iterative samples that are correlated, each sample being dependent only on the previous one (hence generating a Markov chain). As more and more samples are generated, eventually the distribution of sample values will be close to the target distribution. The algorithm proceeds as follows

1. Choose an arbitrary point x_0 to be the first draw and choose a candidate or proposal distribution $Q(x)$.

2. For t = 1, 2, ...

   - Sample a candidate or a proposal value θ_* from the proposal distribution $Q_t(\theta_*|\theta_{t-1})$. This means a candidate is proposed for the next sample value θ_* given the previous value of θ_{t-1}.
   - Calculate the ratio
   $$r = \frac{p(\theta_*|y)/Q_t(\theta_*|\theta_{t-1})}{p(\theta_{t-1}|y)/Q_t(\theta_{t-1}|\theta_*)}.$$
   - Accept the proposed sample with the probability $\min(1,r)$, otherwise reject θ_* and remain at the current value θ_{t-1} and this still counts as an iteration in the algorithm.

A sufficient but not necessary condition for our Markov chain $P(x)$ to reach the target distribution $\pi(x)$ is detailed balance i.e. $\pi(x)P(x'|x) = \pi(x')P(x|x')$. I prove that for the MH algorithm detailed balance holds. I have that

$$\begin{aligned}
\pi(x)P(x'|x) &= \pi(x)Q(x'|x)A(x',x) \\
&= \min\left[\pi(x)Q(x'|x), \pi(x')Q(x|x')\right] \\
&= \min\left[\pi(x')Q(x|x'), \pi(x)Q(x'|x)\right] \\
&= \pi(x')Q(x|x')A(x,x') \\
&= \pi(x')P(x|x'),
\end{aligned} \tag{2.133}$$

where $Q(x'|x)$ is the conditional probability of proposing a state x' given x and the acceptance probability $A(x',x)$ is the probability to accept x', $A(x',x) = \min(1,r)$, $r = \frac{\pi(x')Q(x|x')}{\pi(x)Q(x'|x)}$.

The Hybrid Monte Carlo algorithm

This subsection closely follows Bishop [2006], Section 11.5. The Hybrid or Hamiltonian Monte Carlo (HMC) algorithm is a variation of the Metropolis algorithm that makes use of a 'momentum' parameter that allows to explore the parameter space better, thus improving the mixing, especially in high dimensions. Due to the random walk behaviour, the Metropolis-Hastings sampler might move slowly through the target distribution, HMC tackles this issue by borrowing an idea from physics. By making use of both simulating and deterministic methods, the Hamiltonian Monte Carlo is also called hybrid Monte Carlo. To explain the algorithm I make use of the framework of Hamiltonian dynamics. The Hamiltonian equations are given by

$$\begin{aligned} \frac{d\theta_i}{d\tau} &= \frac{\partial H}{\partial \phi_i}, \\ \frac{d\phi_i}{d\tau} &= -\frac{\partial H}{\partial \theta_i}, \end{aligned} \tag{2.134}$$

where the θ_i's are position variables, ϕ_i's are the 'momentum' variables, evolving in continuous time τ and H is the Hamiltonian function. The joint space of position and momentum variables is called the phase space.

Two important properties of Hamiltonian dynamical systems is the preservation of H and preservation of volume in phase space under the evolution of time. H is preserved as τ evolves since

$$\begin{aligned} \frac{dH}{d\tau} &= \sum_i \left(\frac{\partial H}{\partial \theta_i}\frac{d\theta_i}{d\tau} + \frac{\partial H}{\partial \phi_i}\frac{d\phi_i}{d\tau} \right) \\ &= \sum_i \left(\frac{\partial H}{\partial \theta_i}\frac{\partial H}{\partial \phi_i} - \frac{\partial H}{\partial \phi_i}\frac{\partial H}{\partial \theta_i} \right) = 0. \end{aligned} \tag{2.135}$$

The preservation of volume in a Hamiltonian dynamical system, otherwise known as Liouville's Theorem, means that while a region might change shape, its volume remains unchanged. This can be proven by observing that the flow field (rate of change of location in phase space) is given by

$$\mathbf{V} = \left(\frac{d\boldsymbol{\theta}}{d\tau}, \frac{d\boldsymbol{\phi}}{d\tau} \right). \tag{2.136}$$

The divergence of this field is

$$\begin{aligned} \text{div } \mathbf{V} &= \sum_i \left(\frac{\partial}{\partial \theta_i}\frac{d\theta_i}{d\tau} + \frac{\partial}{\partial \phi_i}\frac{d\phi_i}{d\tau} \right) \\ &= \sum_i \left(-\frac{\partial}{\partial \theta_i}\frac{\partial H}{\partial \phi_i} + \frac{\partial}{\partial \phi_i}\frac{\partial H}{\partial \theta_i} \right) = 0. \end{aligned} \tag{2.137}$$

The joint distribution (a Boltzmann distribution) over the phase space whose total energy is the Hamiltonian is given by

$$p(\boldsymbol{\theta},\boldsymbol{\phi}) = \frac{1}{Z_H}\exp(-H(\boldsymbol{\theta},\boldsymbol{\phi})). \tag{2.138}$$

It can be concluded that $p(\boldsymbol{\theta},\boldsymbol{\phi})$ is invariant due to the fact H is preserved over time and that the volume remains constant as well. Even though H remains constant, the position variables $\boldsymbol{\theta}$ and 'momentum' variables $\boldsymbol{\phi}$ can have multiple values. One way to have ergodic samples from $p(\boldsymbol{\theta},\boldsymbol{\phi})$ is to simply replace the value of $\boldsymbol{\phi}$ with one drawn from its distribution conditioned on $\boldsymbol{\theta}$. This approach does not change the fact that $p(\boldsymbol{\theta},\boldsymbol{\phi})$ is invariant.

A suitable numerical integration scheme of the Hamiltonian equations that minimises numerical errors is called the leapfrog discretisation scheme. This scheme consists of alternating between a series of leapfrog updates and a resampling of the 'momentum' variables from their marginal distribution. More explicitly, the 'momentum' variables are updated using a half-step, followed by a full-step update of the position variables, and again followed by a second half-step update of the 'momentum' variables. The details of the leapfrog scheme are shown below, where $\hat{\theta}$ and $\hat{\phi}$ are discrete-time approximations to the position and momentum variables

$$\hat{\phi}_i\left(\tau+\frac{\varepsilon}{2}\right) = \hat{\phi}_i(\tau) - \frac{\varepsilon}{2}\frac{\partial E}{\partial \theta_i}\left(\hat{\boldsymbol{\theta}}(\tau)\right). \tag{2.139}$$

$$\hat{\theta}_i(\tau+\varepsilon) = \hat{\theta}_i(\tau) + \varepsilon\hat{\phi}_i\left(\tau+\frac{\varepsilon}{2}\right). \tag{2.140}$$

$$\hat{\phi}_i(\tau+\varepsilon) = \hat{\phi}_i\left(\tau+\frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2}\frac{\partial E}{\partial \theta_i}\left(\hat{\boldsymbol{\theta}}(\tau+\varepsilon)\right), \tag{2.141}$$

where $E(\theta)$ is interpreted as the potential energy of the system at state θ. The Hamiltonian function $H(\boldsymbol{\theta},\boldsymbol{\phi})$ contains both potential and kinetic energy such that the following relationship holds

$$H(\boldsymbol{\theta},\boldsymbol{\phi}) = E(\boldsymbol{\theta}) + \frac{1}{2}|\boldsymbol{\phi}|^2. \tag{2.142}$$

The leapfrog discretisation scheme is time-reversible, such that a negative step $-\varepsilon$ will reverse the effect of integration with a positive step ε. The backward or forward integration in time can happen with equal probability ($\frac{1}{2}$). Moreover, it is important to note that during the leapfrog process, unlike the Metropolis-Hastings algorithm the gradients of the log probability distribution are used. If $(\boldsymbol{\phi},\boldsymbol{\theta})$ is the initial state and $(\boldsymbol{\phi}_*,\boldsymbol{\theta}_*)$ is the state after the leapfrog integration, then this proposed state is accepted with probability

$$a = \min(1,\exp\left(H(\boldsymbol{\phi},\boldsymbol{\theta}) - H(\boldsymbol{\phi}_*,\boldsymbol{\theta}_*)\right)). \tag{2.143}$$

If the numerical integration was without numerical errors, then every proposed state would be accepted due to the fact that H is invariant. Thus, there is a need to check that the resulting sample values are drawn from the target distribution. To do this one needs to verify if the

detailed balance condition holds. Firstly, note that the leapfrog scheme is time-reversible, so that a forward trajectory with a positive step-size is inverse to a backward trajectory with a negative step-size. Secondly, another important aspect of this scheme is that it still preserves phase space volume exactly and this is true because the leapfrog scheme updates either variable by an amount that is a function only of the other variable.

Previous results can be used to prove detailed balance. Consider a small region of volume δV around the point $A = (\boldsymbol{\phi}_A, \boldsymbol{\theta}_A)$. A leapfrog scheme is performed and the region around the point $B = (\boldsymbol{\phi}_B, \boldsymbol{\theta}_B)$ is reached. Since the leapfrog scheme preserves volume the regions around points A and B will have the same volume. Due to time reversibility property going back from B to A is possible. The detailed balance condition with respect to the regions around A and B can now be written as

$$\begin{aligned} p(\boldsymbol{\phi}_A, \boldsymbol{\theta}_A)\delta V \times \frac{1}{2} \times \min\left(1, \exp(H(\boldsymbol{\phi}_A, \boldsymbol{\theta}_A) - H(\boldsymbol{\phi}_B, \boldsymbol{\theta}_B))\right) \\ = p(\boldsymbol{\phi}_B, \boldsymbol{\theta}_B)\delta V \times \frac{1}{2} \times \min\left(1, \exp(H(\boldsymbol{\phi}_B, \boldsymbol{\theta}_B) - H(\boldsymbol{\phi}_A, \boldsymbol{\theta}_A))\right). \end{aligned} \tag{2.144}$$

The left hand side of the equation is the probability of moving from the region around A to the region around B. The first factor is the Boltzmann probability for being in the region around A at the start, the second factor ($\frac{1}{2}$) is the probability of selecting a positive step-size for the trajectory rather a negative one, and the third factor is the probability that this trajectory will be accepted. Similarly, the right hand side is the probability of moving from the region around B to the region around A. Using Equations 2.138 and 2.143 the detailed balance condition holds.

To summarise, for each parameter θ_j in the target space, HMC adds an auxiliary variable ϕ_j in order to move fast through the parameter space. Both θ_j and ϕ_j are updated together using a Metropolis step. Consider a joint distribution $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{y}) = p(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\phi})$, from which simulation is done and only the samples of $\boldsymbol{\theta}$ are kept. HMC also requires the gradient of the log-posterior density, which needs to be computed numerically. The momentum distribution $p(\boldsymbol{\phi})$ is usually multivariate normally distributed with mean $\mathbf{0}$ and covariance matrix $\mathbf{M}$, such that the following relationship is satisfied

$$H(\boldsymbol{\theta}, \boldsymbol{\phi}) = E(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\phi}^T M^{-1}\boldsymbol{\phi}. \tag{2.145}$$

From Gelman et al. [2013], Section 12.4, the HMC algorithm proceeds as follows

1. Sample $\boldsymbol{\phi}$ from its prior distribution $\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$.

2. Jointly update $(\boldsymbol{\phi},\ \boldsymbol{\theta})$. Repeat the following L times, where L is the number of leapfrog steps and ε is the step-size (each leapfrong step is scaled by a factor of ε)

   (a) Update firstly $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\varepsilon \frac{d \log p(\mathbf{y}|\boldsymbol{\theta})}{d\boldsymbol{\theta}}$.

   (b) Use the 'momentum' vector $\boldsymbol{\phi}$ to update the 'position' vector $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \varepsilon \mathbf{M}^{-1}\boldsymbol{\phi}.$$

(c) Update again $\boldsymbol{\phi}$

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\varepsilon \frac{d \log p(\mathbf{y}|\boldsymbol{\theta})}{d\boldsymbol{\theta}}.$$

3. Let θ_{t-1} and ϕ_{t-1} be the values of the parameters before the leapfrog process and let θ_* and ϕ_* the values after the leapfrog process. Compute

$$r = \frac{p(\theta_*|\mathbf{y})p(\phi_*)}{p(\theta_{t-1}|\mathbf{y})p(\phi_{t-1})}.$$

4. Accept θ_* with probability $\min(1, r)$. Otherwise, reject θ_* and remain at the current value θ_{t-1} and this still counts as an iteration in the algorithm.

The performance of the HMC is highly dependable on choosing suitable values for ε and L. The number of leapfrog steps L should be large enough to travel trough the posterior space, but a L that is too large would result in a high rejection rate, thus wasting computational resources. If L is too small, then consecutive samples will be close together, thus the samples will exhibit random walk behaviour and it would mix slowly (for $L = 1$, the Metropolis-adjusted Langevin algorithm is obtained).

2.6.2 Convergence diagnostics

In this thesis I regularly make use of convergence diagnostics to test if the Markov chains have converged or not. I assess the convergence of the MCMC chain by looking at two aspects: mixing and stationarity. Convergence is never guaranteed, but there are several tests, both visual and statistic, to assess whether the chain appears to have converged. The visual tests consist of traceplots and the statistics that can be used are Gelman-Rubin statistic [Gelman and Rubin, 1992] or Geweke diagnostic [Geweke, 1992].

The Gelman-Rubin statistic Gelman and Rubin [1992] $\hat{R}$ is a ratio between the variance within the chains to the variance across chains. Multiple MCMC chains for each parameter and from different starting positions need to be run in order to calculate it. When $\hat{R}$ is high (greater than 1.2), then the chains should be run for longer to improve convergence to the stationary distribution. More specifically, following Murphy [2012], Section 24.4.3.1, assume there are S samples (after burn-in) drawn from each of C chains of D variables, x_{isc}, $i = 1 : D$, $s = 1 : S$, $c = 1 : C$. Let y_{sc} be a scalar quantity of interest derived from $\mathbf{x}_{1:D,s,c}$, such as $y_{sc} = x_{isc}$, for some i. Then, the within-sequence mean and the overall mean are defined as

$$\bar{y}_{.c} = \frac{1}{S}\sum_{s=1}^{S} y_{sc}. \tag{2.146}$$

$$\bar{y}_{..} = \frac{1}{C}\sum_{c=1}^{C} \bar{y}_{.c}. \tag{2.147}$$

Also, the between-sequence and within-sequence variance are defined as

$$B = \frac{S}{C-1}\sum_{c=1}^{C}(\bar{y}_{.c} - \bar{y}_{..})^2. \tag{2.148}$$

$$W = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{S-1}\sum_{s=1}^{S}(y_{sc} - \bar{y}_{.c})^2\right). \tag{2.149}$$

Therefore, the scale reduction factor is defined as

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}}, \tag{2.150}$$

where $\hat{V} = \frac{S-1}{S}W + \frac{1}{S}B$. Values of $\hat{R}$ smaller than 1.2 means that probably there is no need to run the chain for a longer time.

Another test to determine if convergence has not occurred is the Geweke test [Geweke, 1992]. The Geweke diagnostic consists in comparing the mean of the first 10% to the last 90% (other values can be taken) of the series. The chain is divided into a number of segments and this difference is computed. It can be said that the chain does not show any signs of non-convergence if the diagnostic varies between -1 and 1. The Geweke score for the chain x is computed by

$$\frac{\mathbb{E}(x_f) - \mathbb{E}(x_l)}{\sqrt{\text{Var}(x_f) + \text{Var}(x_l)}}, \tag{2.151}$$

where $\mathbb{E}$ is the mean, Var is the variance of the chain, x_f is a section at the start of the chain and x_l a section at the end of the chain.

2.6.3 Variational inference methods

GP regression is computationally challenging with the complexity generated by the inversion of the covariance matrix being $\mathcal{O}(N^3)$, where N is the dataset size. To counter this, we employ alternative methods to MCMC sampling-based inference methods such as the variational methods that formulate inference as an optimisation problem and are capable of computing the posterior distribution in a general context [Blei et al., 2017]. Variational inference methods are applied in situations where MCMC methods are difficult and costly to implement and the main idea is centered around a construction of a variational distribution that is approximate to the true posterior distribution. This is done by maximising a lower bound on the marginal likelihood, which is equivalent to minimising a Kullback-Leibler divergence between the approximate variational distribution and the exact posterior distribution. Variational inference methods will be applied in Chapter 6 in a hierarchical non-stationary GP framework to both synthetic and real datasets.

Variational inference for a Gaussian process

In this subsection I show how variational inference techniques are applied when the model is a GP. The main sources used are Titsias [2009], Campioni et al. [2021], Saul et al. [2016], Hensman et al. [2013]. In standard GP regression, assume that some outputs (data) y_i are observed at the input (training) points x_i such that $y_i = f(x_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and f is a latent (unobserved) function. A prior is set on the latent function f such that

$$f(x) \sim \mathcal{GP}\left(m(x),\ \mathbf{K}\left(x_i, x_j\right)\right), \tag{2.152}$$

where $m(x)$ is a mean function, $\mathbf{K}$ is a covariance matrix and x_i, x_j are random training points.

Now consider a set of m inducing points, given by the vector $\mathbf{z}$, which reside in the same space as $\mathbf{x}$, and the set of function values at the inducing points is given by $\mathbf{u}$. The prior distribution over the inducing latent functions is a multivariate Normal distribution i.e.

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{mm}), \tag{2.153}$$

where $\mathbf{K}_{mm}$ is the covariance matrix at the m inducing points.

To derive a lower bound on the marginal likelihood $p(\mathbf{y})$ the following assumption is necessary [Titsias, 2009]

$$p(\mathbf{f}^*|\mathbf{f}, \mathbf{u}) = p(\mathbf{f}^*|\mathbf{u}), \tag{2.154}$$

where $\mathbf{f}^*$ is the function $\mathbf{f}$ evaluated at the test points $\mathbf{x}^*$. Using Equation 2.154 the predictive posterior distribution is

$$p(\mathbf{f}^*, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}^*|\mathbf{u})p(\mathbf{u}|\mathbf{y}). \tag{2.155}$$

To perform variational inference, a variational distribution $\phi(\mathbf{u})$ is introduced such that the following relationship holds

$$q(\mathbf{f}^*, \mathbf{u}) = p(\mathbf{f}^*|\mathbf{u})\phi(\mathbf{u}), \tag{2.156}$$

where $\phi(\mathbf{u})$ is a Gaussian distribution with mean $\boldsymbol{\mu}_q$ and covariance $\mathbf{K}_q$ i.e.

$$\phi(\mathbf{u}) \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{K}_q). \tag{2.157}$$

Thus, from Equations 2.155 and 2.156, the approximation to the predictive posterior distribution is given by the following relationship

$$p(\mathbf{f}^*, \mathbf{u}|\mathbf{y}) \approx p(\mathbf{f}^*|\mathbf{u})\phi(\mathbf{u}). \tag{2.158}$$

The log marginal likelihood has the following form

$$\log p(\mathbf{y}) = \log \iint p(\mathbf{y}|\mathbf{f},\mathbf{u})p(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} = \log \iint p(\mathbf{y}|\mathbf{f})p(\mathbf{f},\mathbf{u})\frac{q(\mathbf{f},\mathbf{u})}{q(\mathbf{f},\mathbf{u})}d\mathbf{f}d\mathbf{u}$$
$$\geq \iint \log\left(\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f},\mathbf{u})}{q(\mathbf{f},\mathbf{u})}\right) q(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} = \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} - \mathscr{KL}(\phi(\mathbf{u})||p(\mathbf{u})), \tag{2.159}$$

where Jensen's inequality is applied at the first inequality, $q(\mathbf{f}) = \int q(\mathbf{f},\mathbf{u})d\mathbf{u}$ and $\mathscr{KL}$ denotes the Kullback-Leibler divergence between the prior distribution $p(\mathbf{u})$ and the variational posterior $\phi(\mathbf{u})$. The $\mathscr{KL}$ divergence between two distributions P and Q is defined by

$$\mathscr{KL}(P||Q) = \int p(x)\log\left(\frac{q(x)}{p(x)}\right)dx, \tag{2.160}$$

where p and q are probability distributions at x.

I explain in more detail why the last equality occurs below

$$\begin{aligned}
\iint \log\left(\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f},\mathbf{u})}{q(\mathbf{f},\mathbf{u})}\right) q(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} &= \iint \log\left(p(\mathbf{y}|\mathbf{f})\right) q(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} \\
&\quad + \iint \log\left(\frac{p(\mathbf{f},\mathbf{u})}{q(\mathbf{f},\mathbf{u})}\right) q(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} \\
&= \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} \\
&\quad + \iint \log\left(\frac{p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{f}|\mathbf{u})\phi(\mathbf{u})}\right) p(\mathbf{f}|\mathbf{u})\phi(\mathbf{u})d\mathbf{f}d\mathbf{u} \\
&= \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} + \iint \log\left(\frac{p(\mathbf{u})}{\phi(\mathbf{u})}\right) q(\mathbf{f},\mathbf{u})d\mathbf{f}d\mathbf{u} \\
&= \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} - \int \log\left(\frac{\phi(\mathbf{u})}{p(\mathbf{u})}\right)\phi(\mathbf{u})d\mathbf{u} \\
&= \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} - \mathscr{KL}(\phi(\mathbf{u})||p(\mathbf{u})),
\end{aligned} \tag{2.161}$$

where I used the properties of the log function in the first equation, $q(\mathbf{f},\mathbf{u}) = p(\mathbf{f}|\mathbf{u})\phi(\mathbf{u})$ and $q(\mathbf{f}) = \int q(\mathbf{f},\mathbf{u})d\mathbf{u}$ in the second equation, properties of log function, $\phi(\mathbf{u}) = \int q(\mathbf{f},\mathbf{u})d\mathbf{f}$, and the definition of $\mathscr{KL}$ in the last equation.

Moreover, the likelihood factorises across the data such that

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{f}_i), \tag{2.162}$$

where N is the number of data points. Thus, after rewriting the integral as an expectation,

Equation 2.159 becomes

$$\begin{aligned}\log p(\mathbf{y}) &\geq \sum_{i=1}^{N} \int \log p(\mathbf{y}_i|\mathbf{f}_i) q(\mathbf{f}_i) d\mathbf{f}_i - \mathscr{KL}(\phi(\mathbf{u})||p(\mathbf{u})) \\ &= \mathbb{E}_{q(\mathbf{f}_i)} \sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{f}_i) - \mathscr{KL}(\phi(\mathbf{u})||p(\mathbf{u})).\end{aligned} \tag{2.163}$$

The sparse variational method can be extended to multiple latent functions, thus defining a chained or a multilatent GP. Following Saul et al. [2016], let two latent functions $\mathbf{f}$ and $\mathbf{g}$ such that the following relationship holds

$$p(\mathbf{f},\mathbf{g}|\mathbf{u}_f,\mathbf{u}_g) = p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g), \tag{2.164}$$

where $\mathbf{u}_f$, $\mathbf{u}_g$ are the latent function values $\mathbf{f}$ and $\mathbf{g}$ evaluated at the inducing points $\mathbf{z}$. Assume that the variational distributions $\phi(\mathbf{u}_f)$ and $\phi(\mathbf{u}_g)$ are normally distributed. Then, the lower bound on the marginal log likelihood is

$$\log p(\mathbf{y}) \geq \iint \log p(\mathbf{y}|\mathbf{f},\mathbf{g}) q(\mathbf{f}) q(\mathbf{g}) d\mathbf{f} d\mathbf{g} - \mathscr{KL}(\phi(\mathbf{u}_f)||p(\mathbf{u}_f)) - \mathscr{KL}(\phi(\mathbf{u}_g)||p(\mathbf{u}_g)), \tag{2.165}$$

where Jensen's inequality is applied, $q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u}_f)\phi(\mathbf{u}_f)d\mathbf{u}_f$ and $q(\mathbf{g}) = \int p(\mathbf{g}|\mathbf{u}_g)\phi(\mathbf{u}_g)d\mathbf{u}_g$.

The likelihood factorises over the data such that we have

$$p(\mathbf{y}|\mathbf{f},\mathbf{g}) = \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i), \tag{2.166}$$

where N is the number of data points. Thus, Equation 2.165 becomes

$$\begin{aligned}\log p(\mathbf{y}) &\geq \sum_{i=1}^{N} \iint \log p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i) q(\mathbf{f}_i) q(\mathbf{g}_i) d\mathbf{f}_i d\mathbf{g}_i - \mathscr{KL}(\phi(\mathbf{u}_f)||p(\mathbf{u}_f)) - \mathscr{KL}(\phi(\mathbf{u}_g)||p(\mathbf{u}_g)) \\ &= \mathbb{E}_{q(\mathbf{f}_i,\mathbf{g}_i)} \sum_{i=1}^{N} p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i) - \mathscr{KL}(\phi(\mathbf{u}_f)||p(\mathbf{u}_f)) - \mathscr{KL}(\phi(\mathbf{u}_g)||p(\mathbf{u}_g)).\end{aligned} \tag{2.167}$$

Equation 2.167 can further be generalised to multiple latent functions. In Equation 2.167, the $\mathscr{KL}$ divergence terms can be calculated analytically since the distributions inside the terms are multivariate Normal distribution. The term difficult to calculate is the expectation term. In a general case, the integral (the expected log likelihood) has a closed formula if the likelihood is Gaussian. In other cases, where the integral is intractable, methods such as Gauss-Hermite quadrature [Hensman et al., 2015] or Monte Carlo sampling [Salimbeni and Deisenroth, 2017, Bonilla et al., 2018, Saul et al., 2016] can generally be used to calculate the expectation. The

formed method increases the computational complexity of the model by introducing a number of n nodes to calculate the integral and the approximation is exact for polynomials of degree less than $2 \times n - 1$. Finally, stochastic optimisation [Hensman et al., 2013] can then be used to maximise the lower bound and perform inference for the parameters of interest.

Chapter 3

A study on discrete-time movement models

Understanding animal movement is an important challenge in ecology, with improvement in tagging technology permitting the collection of data on an increasingly wide range of species. Consequently, methodologies for statistical analysis of such data have received considerable attention in recent years. Discrete-time random walks are the foundation of the movement data models. The advantages of the discrete-time movement models are that they are intuitive and easily implemented, however the specification of the discretisation step is often problematic and must be done in advance. Misspecification of the discretisation step might lead to a model mismatch and thus choosing an appropriate test statistic to capture the model mismatch is essential.

Authors' statement: This Chapter is based on the paper 'A study on discrete-time movement models', that has been published and presented as a conference paper at ICSTA '19, Lisbon [Paun et al., 2019]. Colin Torney, Dirk Husmeier and Ionut Paun designed the study, Ionut Paun performed the analysis and Ionut Paun wrote the manuscript. I confirm that my contribution to each section of the paper is more than 50%.

3.1 Introduction

In Chapter 2, Section 2.1 we introduced the discrete-time CRW model and its extensions as the foundation of movement data models. In this chapter, following Turchin [1998], Morales et al. [2004], Haydon et al. [2008], Hopcraft et al. [2014] we fit statistical models for components of discrete-time random walks. Those components include the step-length and the associated observed turning angle relative to the previous step between each pair of successive observations. The distributions used are Gamma or Weibull distributions for the step-lengths, and Uniform or Wrapped Cauchy distribution for the turning angles [Morales et al., 2004]. In this chapter, in Section 3.2, we introduce several discrete-time movement models and perform inference for the parameters of interest using an MCMC algorithm, namely Metropolis-Hastings, and check for convergence using both visual tests (traceplots) and convergence diagnostics tools (Gelman-

Rubin statistic). This section provides a solid inference foundation needed for Section 3.3, where we simulate data from a CRW model and we assume the discretisation step for the original data to be $\Delta_t = 1$. Then, we change the discretisation step by interpolating with different time-steps and the data after interpolation resembles a real dataset, where the true discretisation time step is unknown. Afterwards, we fit a CRW movement model, implement the MH algorithm to infer the parameters of interest, and then finally perform model checking using multiple test statistics. The main aim of the chapter is to assess whether the different test statistics capture the lack of model fit when fitting a discrete-time model with a different time step than the original data.

3.2 Overview of discrete-time movement models

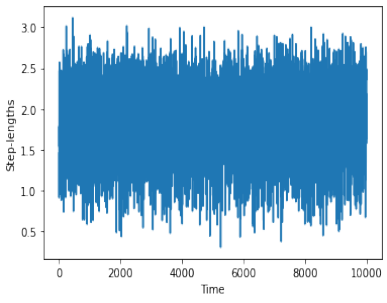
3.2.1 Data

We simulate data from three different discrete-time movement models with the data sample consisting of 10,000 observations.

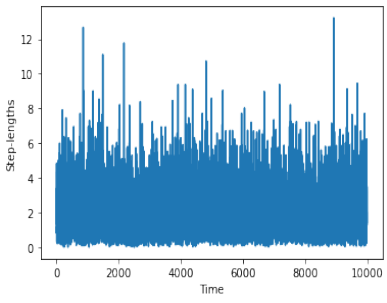
In Figure 3.1 we show plots of the data for all three models. In Table 3.1 below we show all of the relevant information for the data

Data			
Rank of the model	Simulated data sample size	step-lengths distribution	Associated turning angle distribution
First model	10,000	Weibull(5, 2)	Uniform(0, 2π)
Second model	10,000	Weibull(5, 2)	Wrapped Cauchy(0.9)
Third model	10,000	Gamma(2, 1)	Uniform(0, 2π)

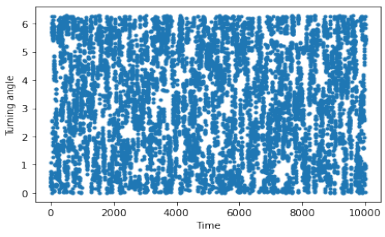
Table 3.1: Simulated data for various discrete-time movement models.



(a) Step-lengths data from a Weibull(5,2) distribution.



(b) Step-lengths data from a Gamma(2,1) distribution.



(c) Turning angles data from a Wrapped Cauchy distribution with a shape parameter value of 0.9.

Figure 3.1: Plots of the data for all three models.

3.2.2 Models

The probability density function of the Weibull distribution is

$$p(x|a,b) = \frac{a}{b}\left(\frac{x}{b}\right)^{a-1} \exp\left(-\left(\frac{x}{b}\right)^{a}\right), \tag{3.1}$$

where $x > 0$, a is the shape parameter and b is the scale parameter, both positive parameters.

The probability density function of the Gamma distribution is

$$p(x|\alpha,\beta) = \frac{\beta^{\alpha} x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)}, \tag{3.2}$$

where $x > 0$, $\alpha, \beta > 0$ and $\Gamma(\alpha)$ is the gamma function. Also, α is the shape parameter and $\beta = 1/\theta$ is the inverse-scale parameter, where θ is the scale parameter.

The probability density function of the Wrapped Cauchy distribution is

$$p(\theta|m,c) = \frac{1-c^2}{2\pi(1+c^2-2c\cos(\theta-m))}, \tag{3.3}$$

where $0 \leq \theta \leq 2\pi$, m is the location and c is the shape parameter, with $0 < c < 1$.

Let r_t represent the observed step-lengths and let θ_t represent the associated observed turning angle. The first model considered is

$$r_t \sim \text{Weibull}(a,b).$$
$$\theta_t \sim \text{Uniform}(0, 2\pi).$$

The Weibull distribution has two parameters, one parameter controlling the scale and the other controlling the shape. If the shape is one, then the Weibull distribution becomes the exponential distribution. If the shape is less than one, the Weibull has mode close to zero and has a long tail, suitable for long movements steps. If the shape is two, then the Weibull distribution is equivalent to Rayleigh distribution and describes the step-length distribution of a standard diffusion process (random walk) [Morales et al., 2004]. The turning angle follows a Uniform distribution, which means that the direction is random, thus the first model is unbiased, i.e. there is equal probability to go in any direction. The Weibull and Gamma distributions are plotted in Figures 3.2a and 3.2b.

The second model considered is

$$r_t \sim \text{Weibull}(a,b).$$
$$\theta_t \sim \text{WrapCauchy}(\theta_{t-1}, c).$$

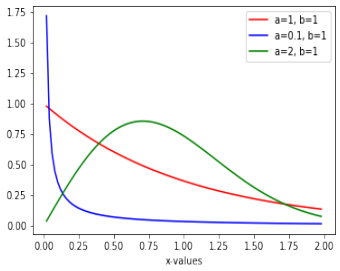
The second model is a 'persistent' random walk or a CRW model. The random walk moves unrestricted and it can go in every direction. The Wrapped Cauchy distribution is more peaked and

has heavier tails compared to the other circular distributions such as Von Mises or the Wrapped Normal distribution. We choose the Wrapped Cauchy distribution in our model, but the other distributions are suitable options as well. The Wrapped Cauchy distribution is plotted for various values of the shape parameter c in Figure 3.2c.

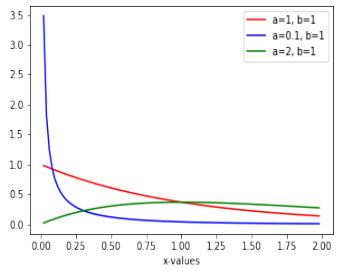
The third model considered is

$$r_t \sim \text{Gamma}(a,b).$$
$$\theta_t \sim \text{Uniform}(0,2\pi).$$

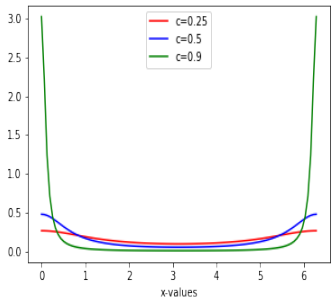
The Gamma distribution has a similar shape to the Weibull distribution as illustrated in Figures 3.2a and 3.2b. When the shape parameters of the two distributions are equal to 1, the two distributions are equivalent to the exponential distribution. Moreover, when the shape parameters are greater than 1, the Weibull distribution decreases at a larger rate than the Gamma distribution and vice versa when the shape parameters are lesser than 1.



(a) Plots of the Weibull distribution for various values of the shape (a) and scale (b) parameters.



(b) Plots of the Gamma distribution for various values of the shape (a) and scale (b) parameters.



(c) Plots of the Wrapped Cauchy distribution for various values of the shape (c) parameter.

Figure 3.2: Plots of the Weibull distribution, Gamma distribution and Wrapped Cauchy distribution for various values of the parameters.

Prior distributions

For the first two models we choose a vague improper prior for all the parameters (shape and scale parameters of the Weibull distribution, and the shape parameter of the Wrapped Cauchy distribution) with $p(x) \propto 1$, where x is one of the aforementioned parameters. For the third model, the prior chosen is Gamma(1,2) with shape parameter 1 and scale parameter 2. A Gamma prior distribution is chosen because it is a conjugate prior distribution for the Gamma distribution when the shape parameter is kept fixed. Thus, we can calculate the analytical posterior distribution. In Bayesian statistics, if the posterior distribution is in the same probability distribution family as the prior probability distribution, then the prior and posterior distributions are called conjugate distributions, and the prior is called a conjugate prior for the likelihood function [Murphy,

2012].

The equation for the prior for the third model is

$$p\left(\frac{1}{\beta}|a,b\right) = \frac{\beta^{1-a}\exp\left(-\frac{1}{b\beta}\right)}{\Gamma(a)b^a}. \tag{3.4}$$

The prior follows a Gamma distribution with shape parameter a, scale parameter b and $\frac{1}{\beta} > 0$.

3.2.3 Likelihood calculation

For the first two models, suppose the step-lengths, $r_1,\ldots,r_n$, are independent and identically draws from a Weibull distribution. For the first model, the log likelihood equation for the step-lengths is proportional to

$$l(a,b;r) \propto \sum_{i=1}^{n} \log\left(\frac{a}{b^a} r_i^{a-1}\exp\left(-\frac{r_i^a}{b^a}\right)\right). \tag{3.5}$$

Suppose the turning angles $\theta_1,\ldots,\theta_n$, are correlated draws from the Wrapped Cauchy distribution, $\theta_t \sim \text{WrapCauchy}(\theta_{t-1},c)$. Thus, for the second model, the log likelihood equation for the turning angles θ's, where c is the shape parameter is proportional to

$$l(c;\theta) \propto \sum_{i=1}^{n} \log\left(\frac{1-c^2}{2\pi(1+c^2-2c\cos(\theta_i-\theta_{i-1})}\right), \tag{3.6}$$

where θ_0 is an initial value for the turning angles, $0 \leq \theta_i \leq 2\pi$ and c is the shape parameter, with $0 < c < 1$.

It is important to note that the log likelihood for the full first model is proportional to Equation 3.5, given that the turning angles are uniformly distributed with parameters 0 and 2π. However, the full log likelihood for the second model is the sum of the log likelihoods from Equations 3.5 and 3.6, given that the models for the step-lengths and turning angles are independent. Therefore, the joint likelihood equation factorises and the step-lengths and turning angles are inferred independently.

For the third model, suppose the step-lengths, $r_1,\ldots,r_n$, are independent and identically distributed draws from a Gamma distribution where α, the shape parameter is known and $\frac{1}{\beta}$, the rate parameter (or the reciprocal of the scale parameter) is unknown. For the third model, the log likelihood function, $l(\beta;r_1,\ldots,r_n)$ is proportional to

$$l(\beta;r_1,\ldots,r_n) = \log\left(\mathscr{L}(\beta;r_1,\ldots,r_n)\right) \propto \log\left(\frac{\exp\left(\frac{-\sum_{i=1}^{n} r_i}{\beta}\right)}{\beta^{\alpha n}}\right). \tag{3.7}$$

For the third model, the log likelihood for the full model is given by Equation 3.7 keeping in

mind that the turning angles are uniformly distributed with parameters 0 and 2π. For this model, we also wish to calculate analytically the posterior distribution. The prior follows a Gamma distribution with shape a and scale b. By using Equation 3.4 for the prior defined in Section 3.2.2, the likelihood Equation 3.7 and the fact that the posterior probability is proportional to the likelihood multiplied by the prior probability we get that

$$\pi\left(\frac{1}{\beta}|a',b'\right) \propto p\left(\frac{1}{\beta}|a,b\right) \times \mathscr{L}(\beta;r_1,\ldots,r_n) = \frac{\beta^{1-a}\exp\left(-\frac{1}{b\beta}\right)}{\Gamma(a)b^a} \times \frac{\exp\left(\frac{-\sum_{i=1}^n r_i}{\beta}\right)}{\beta^{\alpha n}}. \quad (3.8)$$

After combining terms the posterior distribution, $\pi\left(\frac{1}{\beta}|a',b'\right)$ is a Gamma distribution with hyperparameters

$$a' = \alpha n + a. \quad (3.9)$$

$$b' = \frac{b}{1+b\sum_{i=1}^n x_i}. \quad (3.10)$$

3.2.4 Inference

For the first model, our goal is to infer the step-lengths, more specifically the shape and the scale parameters of the Weibull distribution. For the second model, our goal is to infer the turning angles, more specifically the shape parameter c of the Wrapped Cauchy distribution and for the third model, our goal is to infer the scale parameter b of the Gamma distribution, while keeping the shape parameter a fixed at 2.

To infer the parameters we use a MCMC method, namely the MH algorithm. MCMC is not the natural methodological choice in this case, as there are more efficient inference techniques that can be used such as directly sampling from the Cauchy or Weibull distribution, importance sampling, rejection sampling or slice sampling [Murphy, 2012]. However, these approaches would not be applicable to more complex models. MCMC, on the other hand, is a generally applicable tool. It was therefore chosen as a testbed for future, more general models.

For the third model the natural MCMC method to use is the Gibbs sampling algorithm, given that there is a conditional probability (the conjugate prior for the rate parameter $\frac{1}{\beta}$). However, the objective of this chapter is to check the convergence of a general MH sampler that we have implemented. Therefore, the MH algorithm is chosen as the inference method for the third model, as in the later chapters of the thesis we apply MCMC sampling schemes to new models that do not have conditional prior distributions, where the Gibbs sampling algorithm cannot be applied. Therefore, the conjugate prior is used to effectively compare the posterior samples against an analytically tractable posterior distribution. This is done by applying a standard hypothesis test i.e. a Kolmogorov–Smirnov (KS) test [Hodges, 1958]. If the KS test returns a small KS statistic or a high p-value, then the null hypothesis that the underlying distribution of the posterior samples is identical to the analytical posterior distribution cannot be rejected in

favour of the alternative, that is the underlying distribution of the posterior samples is different to the analytical posterior distribution.

For all models, the proposal distribution is a symmetric Normal distribution, the number of iterations are 10,000 and the burn-in is 1000 iterations for each model. Negative proposals for the shape and scale parameters of the Weibull and Gamma distributions are rejected. Similarly, proposals for the shape parameter of the Wrapped Cauchy distribution that are not between 0 and 1 are rejected. The acceptance rates are close to 40% and the step-sizes are tuned to give an acceptance probability within the desired interval (between 25%-40%) [Murphy, 2012], Section 24.3. The potential scale reduction factors are less than 1.1.

3.2.5 Results

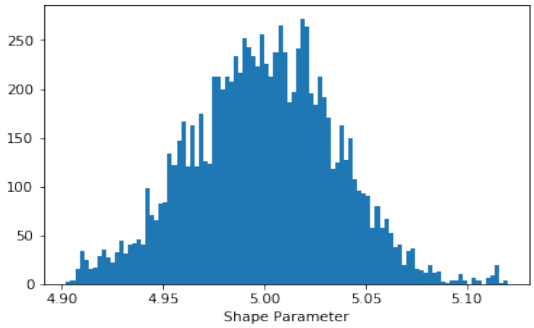
For the first model, we plot the histograms of the marginal posterior for the shape, respectively the scale parameter of the posterior distribution in Figures 3.3a and 3.3b. The histograms are noisy due to a large number of bins, therefore smoother kernel density estimation (KDE) plots using Gaussian kernels are plotted for the first model in Figures 3.3c and 3.3d. The data used to produce the plots are the posterior samples from the MCMC inference and the bandwith method used is Scott's rule [Scott, 1979]. In Figure 3.3a, the mode is close to 2, and in Figure 3.3b, it is close to 5, which is expected given that the data comes from a Weibull distribution with the shape parameter 5 and scale parameter 2.

For the second model, we plot the log likelihood of the shape parameter of the Wrapped Cauchy distribution in Figure 3.4a, which shows that the mode is located when the shape parameter is close to 0.9, as expected given that the data comes from a Wrapped Cauchy distribution with the shape parameter 0.9. Also, in Figure 3.4b, we plot a histogram of the posterior samples and the mode is close to 0.9 as expected.

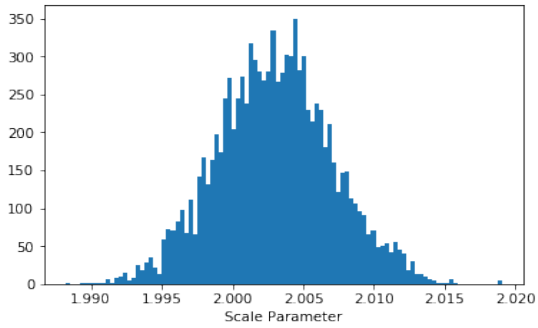
For the third model, we plot in Figure 3.4c the profile log likelihood for the scale parameter while keeping the shape parameter fixed at 2. The mode in Figure 3.4c is located around 1, which is as expected given that we simulated the step-lengths from a Gamma distribution with the scale parameter 1. In Figure 3.4d, the mode of the posterior samples for the scale parameter is located around $e^0 = 1$, which is again as expected. In Figure 3.4d, we plot together the analytical posterior pdf from Equation 3.8 and the samples obtained from the MCMC sampler, which are in agreement. This is confirmed by the KS test, which returned a value of the KS statistic of 0.0307 and a p-value of 0.1995. Since the p-value is not extreme, we cannot reject the null hypothesis that the underlying distribution of the posterior samples is identical to the analytical posterior distribution. Thus, the MH inference is successful in generating samples from the posterior. We illustrate the inference results in Table 3.2.

Results of the MCMC inference			
Weibull distribution shape parameter mean / true value	Weibull distribution scale parameter mean / true value	Wrapped Cauchy shape parameter mean / true value	Gamma distribution scale parameter mean / true value
5.011 ± 0.03 std / 5	2.001 ± 0.004 std / 2	0.89 ± 0.001 std / 0.9	0.98 ± 0.007 std / 1

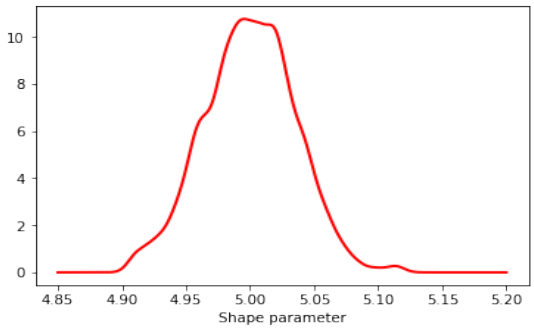
Table 3.2: Table of the inference results.



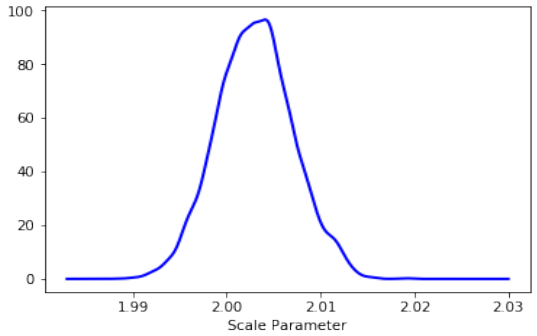
(a) Histogram of the marginal posterior for the shape parameter of the Weibull distribution for the first model.



(b) Histogram of the marginal posterior for the scale parameter of the Weibull distribution for the first model.

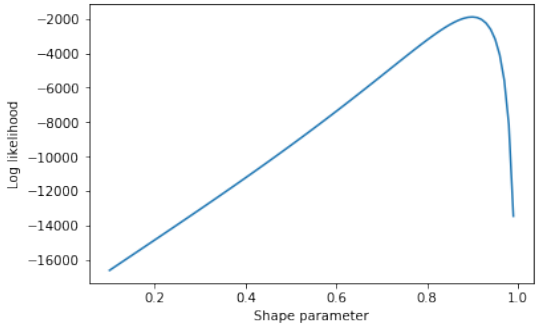


(c) Plot of the marginal posterior for the shape parameter of the Weibull distribution using kernel density estimation. The kernel is Gaussian and the bandwidth method is Scott's rule [Scott, 1979].

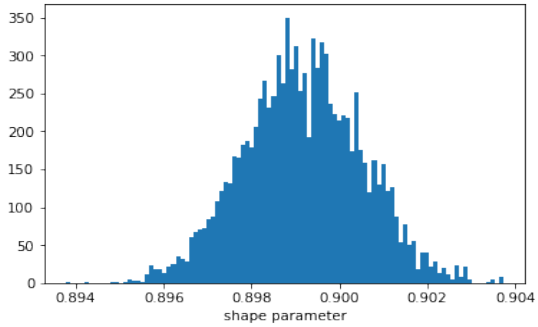


(d) Plot of the marginal posterior for the scale parameter of the Weibull distribution using kernel density estimation. The kernel is Gaussian and the bandwidth method is Scott's rule [Scott, 1979].

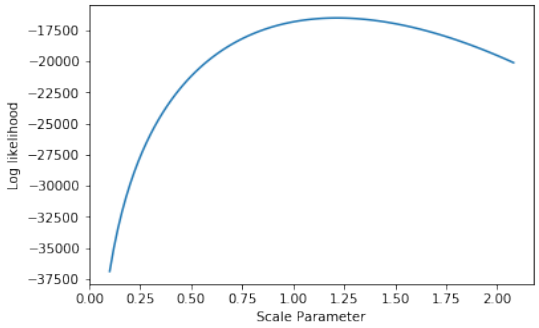
Figure 3.3: Plots of the marginal posterior distribution parameters of the Weibull distribution. The true values for the shape and scale parameters of the Weibull distribution are 5, respectively 2.



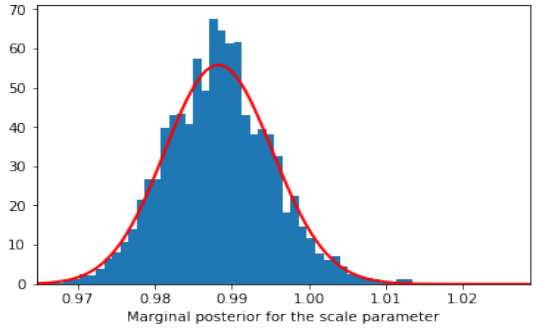
(a) Plot of the log likelihood of the shape parameter of the Wrapped Cauchy distribution for the second model.



(b) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution for the second model.



(c) Plot of the profile log likelihood for the scale parameter of the Gamma distribution while keeping the shape parameter of the Gamma distribution fixed at 2 for the third model.

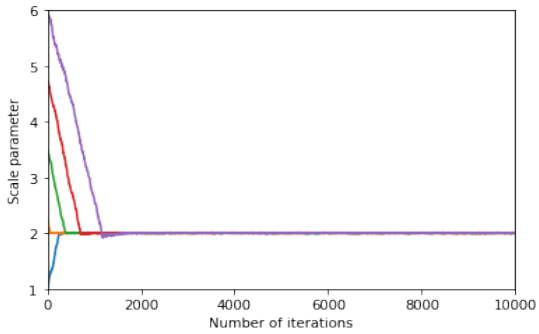


(d) Plot of the pdf of the analytical posterior distribution (a Gamma distribution) (red line) and the marginal posterior samples for the scale parameter of the Gamma distribution for the third model.

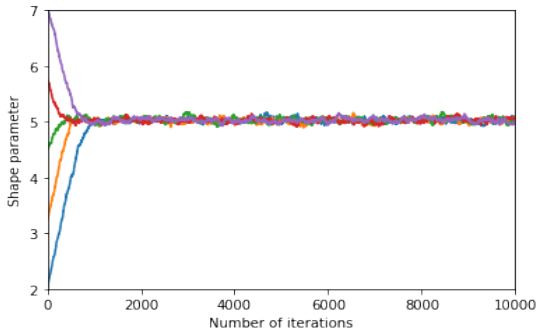
Figure 3.4: Plots of the profile log likelihoods and of the histograms of the posterior distributions for multiple parameters. The true value for the shape parameter of the Wrapped Cauchy distribution is 0.9. The true value of the shape parameter for the Gamma distribution is 1.

3.2.6 Assessing convergence

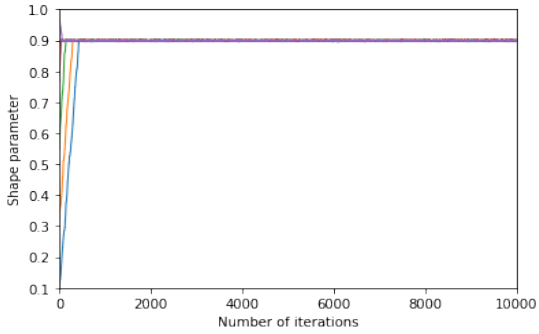
For all the models, we use both visual plots in the form of traceplots and convergence diagnostics such as Gelman-Rubin statistic [Murphy, 2012] to assess if the MCMC chains show a lack of convergence. In our case, we run 5 MCMC chains from different starting values to calculate the Gelman-Rubin statistics for each parameter. To calculate the Gelman-Rubin statistic we follow the method described in Section 2.6.2. For all models, the Gelman-Rubin statistic is very close to 1, the traceplots in Figure 3.5 show that they reached stationarity and that the mixing is good. Therefore, in all cases the MCMC chains do not show a lack of convergence.



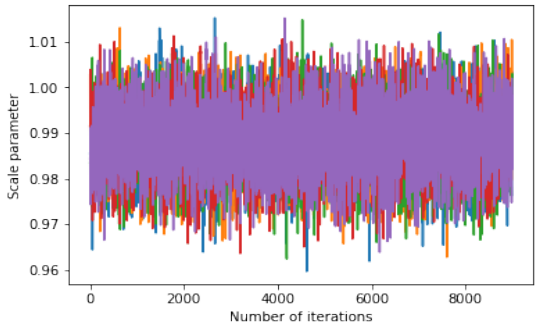
(a) Traceplots of the scale parameter of the Weibull distribution for the first model.



(b) Traceplots for the shape parameter of the Weibull distribution for the first model.



(c) Traceplots of the shape parameter of the Wrapped Cauchy distribution for the second model.



(d) Traceplots for the scale parameter of the Gamma distribution for the third model.

Figure 3.5: Traceplots for the parameters of interest. The different colours represent multiple chains starting from different initialisations.

3.3 Changing the discretisation step

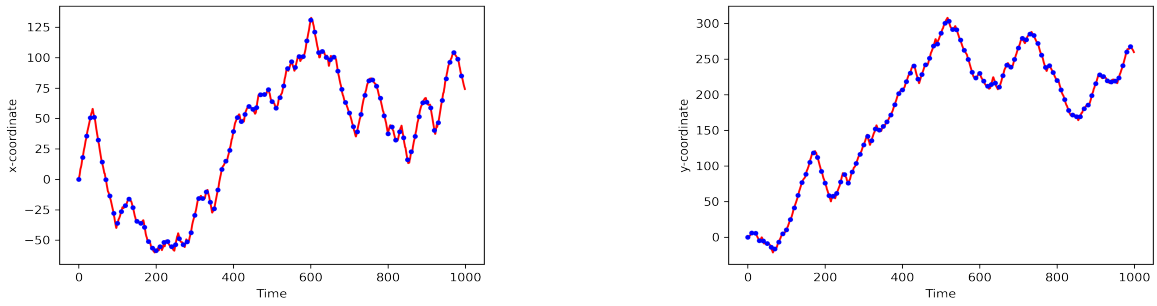
One crucial aspect when employing a discrete-time movement model is the selection in advance of a fixed discretisation step. The choice of a suitable sampling scheme will depend on the aim of the study. If the goal is to study the long-term behaviour of an animal, then a sampling frequency of months is a viable option, however if the aim of the study is the analysis of the short-term behaviour of an animal, then a high sampling frequency of minutes is a good choice. Moreover, if the discretisation step is large, the trajectory will appear more random as the correlation between points is lost [Codling and Hill, 2005]. If the discretisation step is too small, then we lose information about the long-term behaviour of an animal. Inference of animal movement in a discrete-time framework is not time scale invariant, and thus it is very important that the discretisation step is specified such that it matches with the scale at which behavioural decisions are made [McClintock et al., 2014].

In this section we simulate data from a CRW model and we assume the true discretisation step for the original data to be $\Delta_t = 1$. We change the discretisation step by interpolating with

different time steps and the data after interpolation resembles a real dataset (in a real dataset the discretion-step is unknown). Afterwards, we fit a CRW movement model, perform inference using the MH algorithm, check for convergence and then finally do model checking using multiple test statistics. The main aim of this section is to assess whether different test statistics capture the lack of model fit when fitting a discrete-time model with a different time step than the original data.

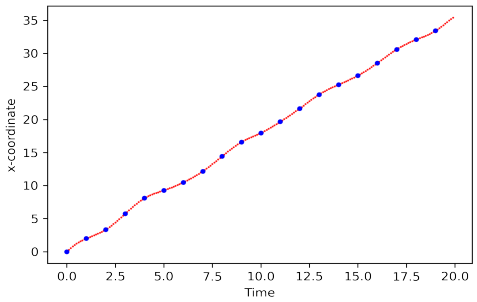
3.3.1 Data

We simulate data for the step-lengths from a Weibull distribution with the shape parameter 5 and the scale parameter 2. The associated observed turning angles data are simulated from a Wrapped Cauchy distribution with shape parameter 0.9. The sample size is 1000 and the time step for the original dataset is set to $\Delta_t = 1$ arbitrary time unit, meaning that every time unit Δ_t we observe a position of an individual animal. We denote this dataset: dataset 0. Using cubic interpolation we sample data with a new time step and we denote this dataset: dataset 1. We analyse different cases when the new time steps are 0.1, 0.2, 0.3, respectively 2 time units. For example, when the time step is 0.1 time units, this means that an observation is recorded every 0.1 time units, compared to the previous case when we recorded an observation every 1 time unit. From the x-coordinates and y-coordinates obtained from the interpolation, we calculate the step-lengths and the associated turning angles. When the new time step, $\Delta_t = 0.1$ we plot the data in Figures 3.6 and 3.7.

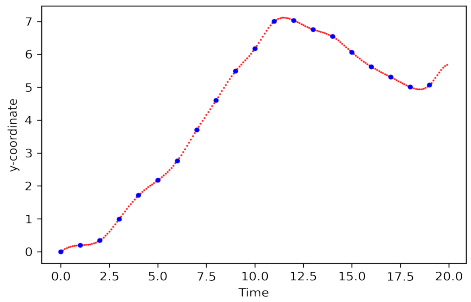


(a) (b)

Figure 3.6: Plot of the x-coordinates, respectively y-coordinates from newly obtained dataset after interpolation, dataset 1 (red line) and of every 10-th x-coordinate, respectively 10-th y-coordinate from the original dataset, dataset 0 (blue dots) for $\Delta_t = 0.1$.



(a)



(b)

Figure 3.7: Plot of the x-coordinates, respectively y-coordinates from the newly obtained dataset after interpolation, dataset 1 (red circles) and the first 20 observations from the original dataset, dataset 0 (blue dots) for $\Delta_t = 0.1$.

3.3.2 Model

Let r_t represent the observed step-length and let θ_t represent the associated observed turning angle at time t. The CRW model considered is

$$r_t \sim \text{Weibull(a,b)}.$$
$$\theta_t \sim \text{WrapCauchy}(\theta_{t-1}, c).$$

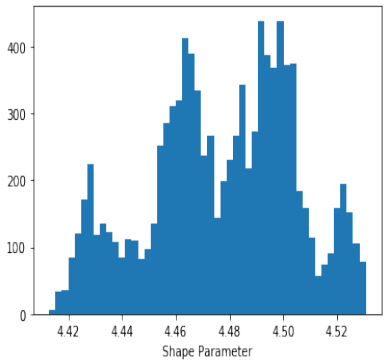
We choose a vague improper prior for all the parameters (a, b and c) such that $p(x) \propto 1$, where $x \in \{a, b, c\}$.

3.3.3 Inference

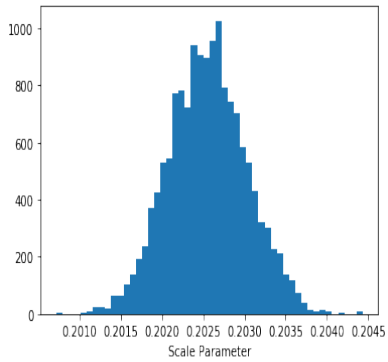
One of our goals in this section is to infer the shape and scale parameters of the Weibull distribution and the shape parameter of the Wrapped Cauchy distribution. In order to do that, we use two MCMC samplers, one to infer the shape and scale parameters from the Weibull distribution and the other one to infer the shape parameter from the Wrapped Cauchy distribution. The algorithm chosen to infer the parameters of interest is MH, a general inference tool. The proposal distribution is a symmetric Normal distribution in both cases. When the interpolating time step is 0.2, 0.3 and 2, the MCMC sample sizes are 20,000 and the burn-in used is 5000 samples for the shape and scale parameters of the Weibull distribution and 1000 samples for the shape parameter of the Wrapped Cauchy distribution. When $\Delta_t = 0.1$, we use more samples (30,000) for the first MCMC sampler and the burn-in used is 20,000 samples for the shape and scale parameters of the Weibull distribution. For $\Delta_t = 1$, the number of MCMC samples is 50,000 and the burn-in is the same as for when the time step is 0.2, 0.3 and 2. The step-sizes were tuned to give an acceptance probability within the desired interval: between 25%-40% [Murphy, 2012], Section 24.3.

3.3.4 Results

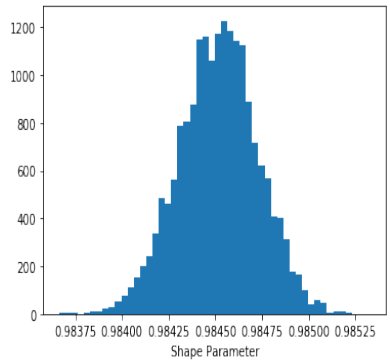
In Figures 3.8-3.12 we plot histograms of the posterior samples for the shape and scale parameters of the Weibull distribution and of the posterior samples for the shape parameter of the Wrapped Cauchy distribution for all the different time steps. We illustrate the inference results in Table 3.3. Analysing Table 3.3, the mean shape parameters do not change much as the result of the interpolation however, the scale parameter gets divided accordingly. For example, if the interpolation step is 0.1, then we have in total 10 times more observations, and the mean scale parameter is almost 10 times less (0.204) than the original scale parameter 2. For all the step-sizes we obtain similar results. For $\Delta_t = 0.1$, we plot in Figure 3.6 the new dataset (dataset 1) after interpolation.



(a) Histogram of the posterior samples of the shape parameter of the Weibull distribution.

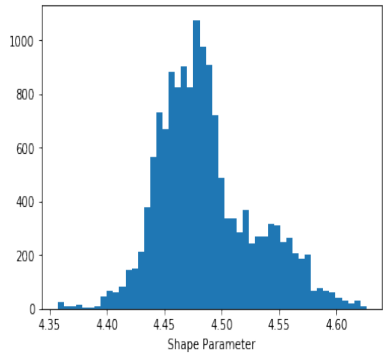


(b) Histogram of the posterior samples of the scale parameter of the Weibull distribution.

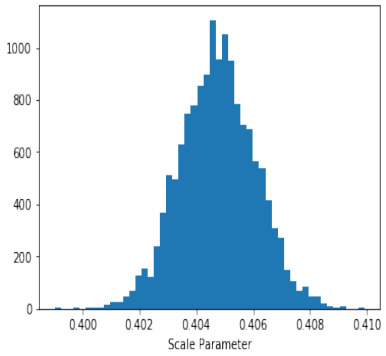


(c) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution.

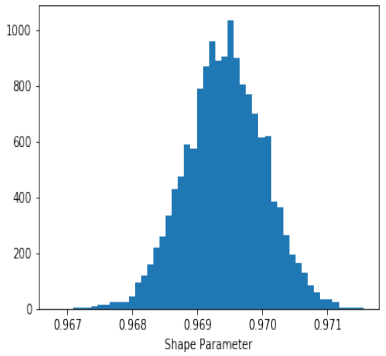
Figure 3.8: Histograms of the posterior samples for $\Delta_t = 0.1$. The true values (when $\Delta_t = 1$) for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.



(a) Histogram of the posterior samples of the shape parameter of the Weibull distribution.

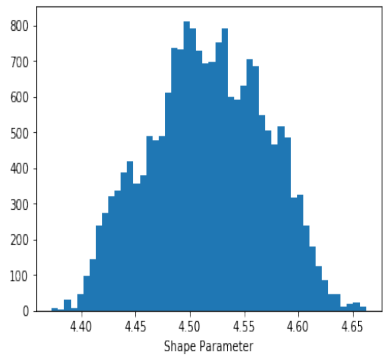


(b) Histogram of the posterior samples of the scale parameter of the Weibull distribution.

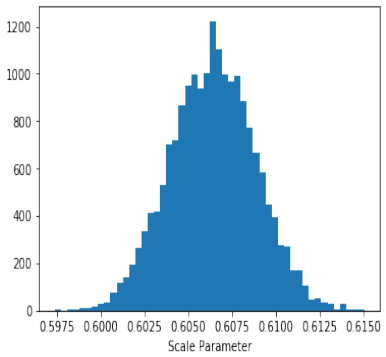


(c) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution.

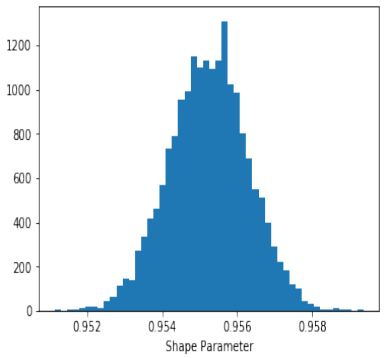
Figure 3.9: Histograms of the posterior samples for $\Delta_t = 0.2$. The true values (when $\Delta_t = 1$) for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.



(a) Histogram of the posterior samples of the shape parameter of the Weibull distribution.

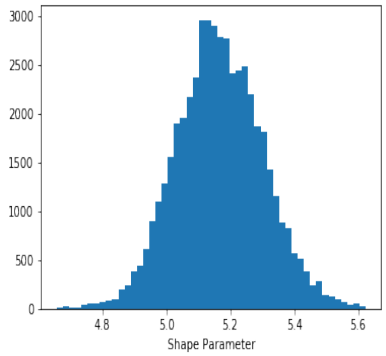


(b) Histogram of the posterior samples of the scale parameter of the Weibull distribution.

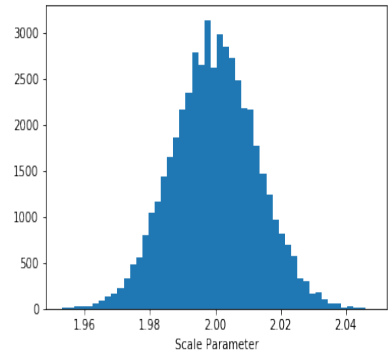


(c) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution.

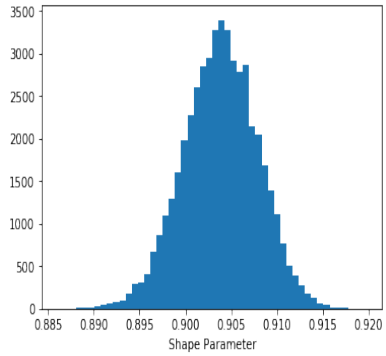
Figure 3.10: Histograms of the posterior samples for $\Delta_t = 0.3$. The true values (when $\Delta_t = 1$) for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.



(a) Histogram of the posterior samples of the shape parameter of the Weibull distribution.

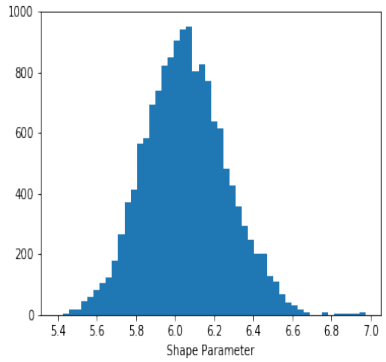


(b) Histogram of the posterior samples of the scale parameter of the Weibull distribution.

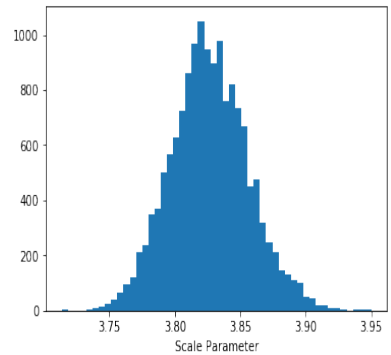


(c) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution.

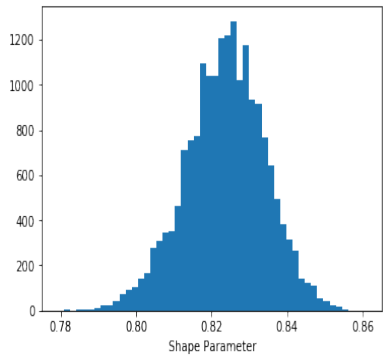
Figure 3.11: Histograms of the posterior samples for $\Delta_t = 1$. The true values (when $\Delta_t = 1$) for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.



(a) Histogram of the posterior samples of the shape parameter of the Weibull distribution.



(b) Histogram of the posterior samples of the scale parameter of the Weibull distribution.



(c) Histogram of the posterior samples of the shape parameter of the Wrapped Cauchy distribution.

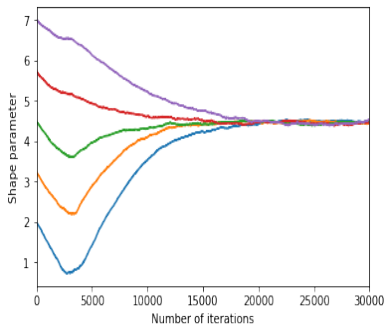
Figure 3.12: Histograms of the posterior samples for $\Delta_t = 2$. The true values (when $\Delta_t = 1$) for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.

Results of the MCMC inference				
Interpolation time step	Sample size	Weibull shape parameter mean	Weibull scale parameter mean	Wrapped Cauchy shape parameter mean
0.1	9988	4.74 ± 0.022 std	0.202 ± 0.0004 std	0.983 ± 0.0002 std
0.2	4993	4.49 ± 0.05 std	0.404 ± 0.001 std	0.969 ± 0.0005 std
0.3	3328	4.52 ± 0.05 std	0.606 ± 0.002 std	0.955 ± 0.001 std
1	997	5.16 ± 0.13 std	1.99 ± 0.013	0.903 ± 0.004 std
2	500	6.04 ± 0.21 std	3.82 ± 0.029 std	0.82 ± 0.01 std

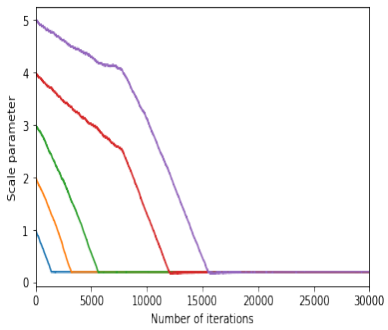
Table 3.3: Table of the inference results. The true values for the shape and scale parameters of the Weibull distribution are 5, respectively 2, and for the Wrapped Cauchy distribution the true value for the shape parameter is 0.9.

3.3.5 Convergence

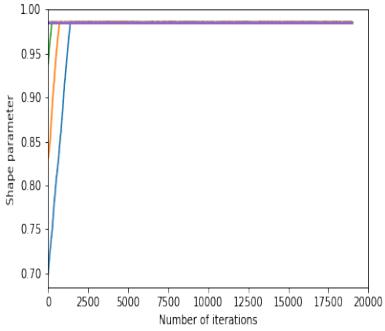
In this section we assess whether the MCMC chains show a lack of convergence by using both visual plots (traceplots) illustrated in Figures 3.13-3.16 for all the different time steps and convergence diagnostics such as Gelman-Rubin statistic. We run 5 MCMC chains from different initial positions. In all cases, the traceplots show good mixing and that the MCMC chains have reached stationarity. Also, the Gelman-Rubin statistic is very close to 1, therefore we do not have evidence of lack of convergence.



(a) Traceplots of the shape parameter of the Weibull distribution.

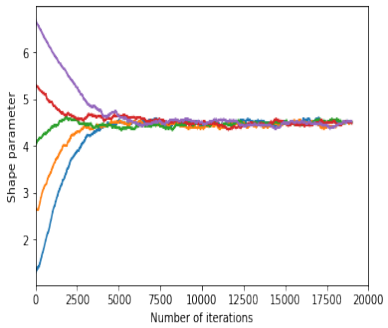


(b) Traceplots of the scale parameter of the Weibull distribution.

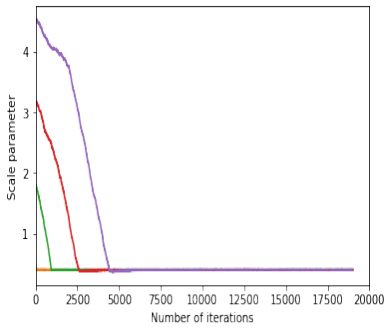


(c) Traceplots of the shape parameter of the Wrapped Cauchy distribution.

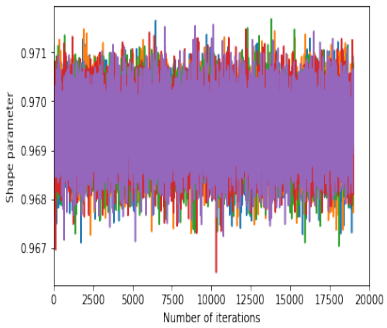
Figure 3.13: Traceplots of the parameters samples for $\Delta_t = 0.1$. The different colours represent multiple chains starting from different initialisations.



(a) Traceplots of the shape parameter of the Weibull distribution.

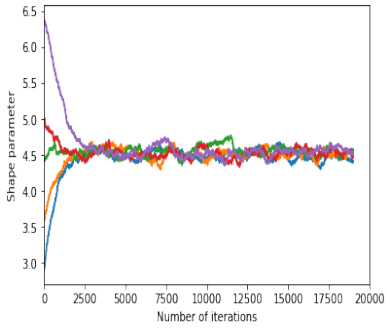


(b) Traceplots of the scale parameter of the Weibull distribution.

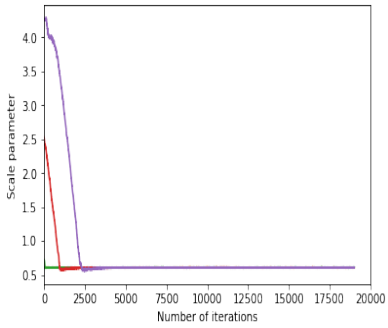


(c) Traceplots of the shape parameter of the Wrapped Cauchy distribution.

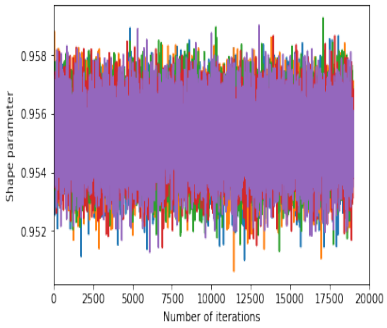
Figure 3.14: Traceplots of the parameter samples for $\Delta_t = 0.2$. The different colours represent multiple chains starting from different initialisations. The difference between the first two plots and the third plot lies in the fact that the initial starting points for the multiple chains in the first two plots are more dispersed than in the third plot, where the starting points are close to the true value.



(a) Traceplots of the shape parameter of the Weibull distribution.



(b) Traceplots of the scale parameter of the Weibull distribution.

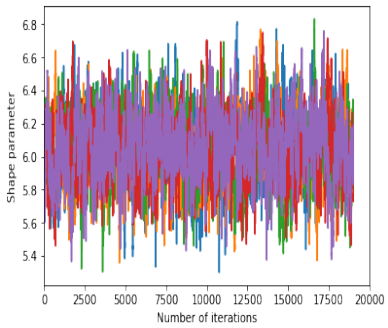


(c) Traceplots of the shape parameter of the Wrapped Cauchy distribution.

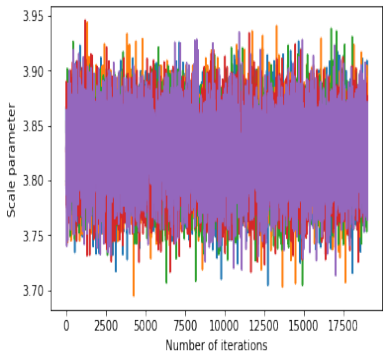
Figure 3.15: Traceplots of the parameter samples for $\Delta_t = 0.3$. The different colours represent multiple chains starting from different initialisations. The difference between the first two plots and the third plot lies in the fact that the initial starting points for the multiple chains in the first two plots are more dispersed than in the third plot, where the starting points are close to the true value.

3.3.6 Model checking

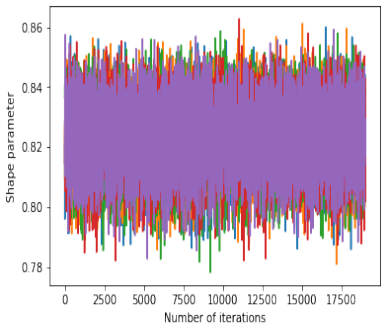
In this section we check whether our inferred model is a good fit to our data (dataset 1). Following Gelman et al. [2013], Chapter 6, we perform model checking: The most popular model checking approach consists of generating replicates samples from the posterior predictive distribution and observe the behaviour of sample summaries over repeated sampling. The goal in model checking is to calculate some statistic for which we have some idea of what an 'extreme'



(a) Traceplots of the shape parameter of the Weibull distribution.



(b) Traceplots of the scale parameter of the Weibull distribution.



(c) Traceplots of the shape parameter of the Wrapped Cauchy distribution.

Figure 3.16: Traceplots of the parameter samples for $\Delta_t = 2$. The different colours represent multiple chains starting from different initialisations.

value would be in the true dataset and compare with the same statistic calculated in predictive datasets. To conduct a posterior predictive check, we do the following

1. Come up with a suitable test statistic T that has power to diagnose violations of whatever assumption we are testing.

2. Calculate T for the observed data $\mathbf{y}$: T($\mathbf{y}$, $\boldsymbol{\phi}$), where $\boldsymbol{\phi}$ is a vector of the inferred parameters using MCMC.

3. Calculate T for each draw $\mathbf{y}$_rep from the posterior predictive distribution: this gives T($\mathbf{y}$_rep, $\boldsymbol{\phi}$).

4. Compare the posterior predictive distribution of T($\mathbf{y}$_rep, $\boldsymbol{\phi}$) to T($\mathbf{y}$, $\boldsymbol{\phi}$).

5. Calculate the Bayesian p-value or posterior predictive p-value (ppp-value) defined as the probability that the replicated samples $\mathbf{y}$_rep could be more extreme than the observed data $\mathbf{y}$, as measured by the test quantity

$$p_B = \Pr\left(T(\mathbf{y}_{\text{rep}}, \boldsymbol{\phi}) \geq T(\mathbf{y}, \boldsymbol{\phi}|\mathbf{y})\right), \tag{3.11}$$

   where the probability is calculated over the posterior distribution of $\boldsymbol{\phi}$ of $\mathbf{y}_{\text{rep}}$.

6. In practice, to compute the posterior predictive p-value we estimate the p-value by calculating the fraction of times from S simulations that T($\mathbf{y}^s_{\text{rep}}$, $\boldsymbol{\phi}^s$) $\geq T(\mathbf{y}, \boldsymbol{\phi}^s)$, for $s = 1, \ldots, S$. If the estimated posterior predictive p-value is close to 0 or 1 (say 0.05 or 0.95), then it suggests that something in our model is inadequate.

In our case we use two test-statistics, the log likelihood and the diffusion coefficient. To generate the replicate datasets we use 500 different sets of posterior samples from the two MCMC

samplers (one to infer the shape and scale parameters of the Weibull distribution and the other one to infer the shape parameter of the Wrapped Cauchy distribution). Plots of the path of the CRW model for the replicate and observed datasets when $\Delta_t = 0.1$ are illustrated in Figure 3.17 - 3.18.

Log likelihood as a test statistic

In this section we use the log likelihood as a test statistic to check whether our inferred model is a good fit to the observed data i.e. dataset 1, the data obtained after interpolation. The test statistic T($\mathbf{y}$) is the sum of the log likelihood of the Weibull distribution for the step-lengths and of the log likelihood of the Wrapped Cauchy distribution for the turning angle.

The diffusion coefficient as a test statistic

In this section we use the diffusion coefficient as a test statistic. To accomplish this, we make use of the following mathematical equation [Codling et al., 2008]

$$u^2 = 4D\tau, \tag{3.12}$$

where u is the expected distance from the origin at time step τ, and D is the diffusion coefficient. For the observed dataset, we calculate the distance from the origin at each time step, then using Equation 3.12 and linear regression we calculate the slope D.

For the replicated datasets, we can minimise the uncertainty when calculating the diffusion coefficient D by computing the average distance from the origin at each time step from multiple simulations (50 in our case), but uncertainty deriving from the MCMC inference of the parameters is still present. More specifically, suppose $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)$ be a set of parameters obtained from the MCMC inference, where ϕ_1, ϕ_2 are the shape and scale parameters of the Weibull distribution and ϕ_3 is the shape parameter of the Wrapped Cauchy distribution. Using this set of parameters we generate data multiple times, calculate the distance from the origin at each time step for every simulation, and then take the average in order to calculate the average distance from the origin vector. We can represent this mathematically. Suppose the distance from the origin vector to any position for the k-th simulated sample is $\mathbf{v}^k = (v_0^k,\ v_{\Delta_t}^k,\ v_{2\Delta_t}^k, \dots)$, where v_0^k is the distance from the origin to the origin (the first entry is always going to be 0), $v_{\Delta_t}^k$ is the distance from the origin to the position at time step Δ_t, Δ_t is the interpolation time step, and k is a natural number between 1 and 50. Thus, the average distance from the origin vector across all simulations is $\mathbf{u} = \left(\frac{\sum_{k=1}^{k=50} v_0^k}{50},\ \frac{\sum_{k=1}^{k=50} v_{\Delta_t}^k}{50},\ \frac{\sum_{k=1}^{k=50} v_{2\Delta_t}^k}{50}, \dots \right)$. We repeat the same procedure to calculate the average distance from the origin for the remaining replicated samples. It is important to stress that each replicate sample is generated using a different set of parameters, but we average across the simulations using the same set of parameters.

Model checking results

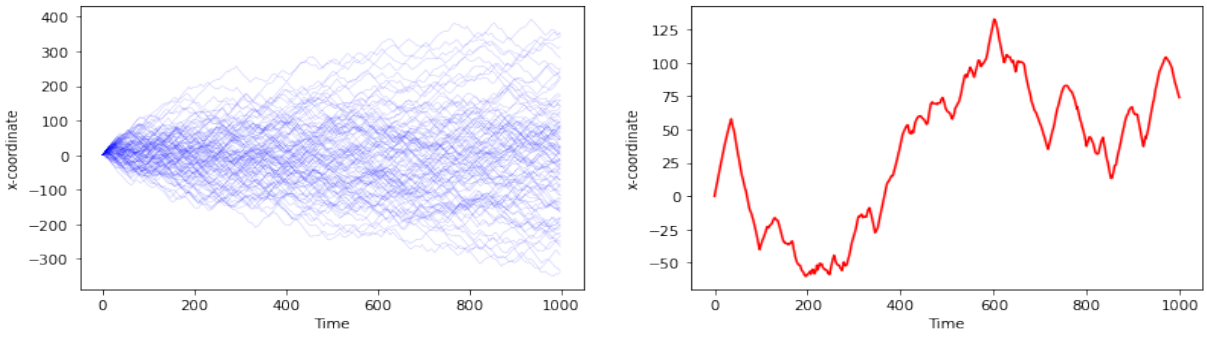
In this section we show the results of the model checking using the two test statistics. In Table 3.4 we show the results using log likelihood as a test statistic and in Figures 3.19-3.23 we plot scaled scaled histograms of the log likelihood test statistic for the replicate samples $T(\mathbf{y}_{rep})$ and for dataset 1, $T(\mathbf{y})$. In all cases the log likelihoods were re-scaled to adjust to the fact that the sample size for the replicate datasets is 1000, compared to the dataset 1 sample size. In Table 3.5 we show the results using the diffusion coefficient as a test statistic. In Figures 3.24 and 3.25 we plot histograms of the replicate samples together with the test statistics $T(\mathbf{y})$ computed using the diffusion coefficient for all the cases.

t	The dataset 1 sample size	Bayesian p-value
0.1	9988	0.004
0.2	4993	0.012
0.3	3328	0.011
1	1000	0.566
2	500	0.58

Table 3.4: Model checking results using re-scaled log likelihood per data point as a test statistic.

t	The dataset 1 sample size	Bayesian p-value
0.1	9988	0
0.2	4993	0
0.3	3328	0
1	1000	0.188
2	500	0.026

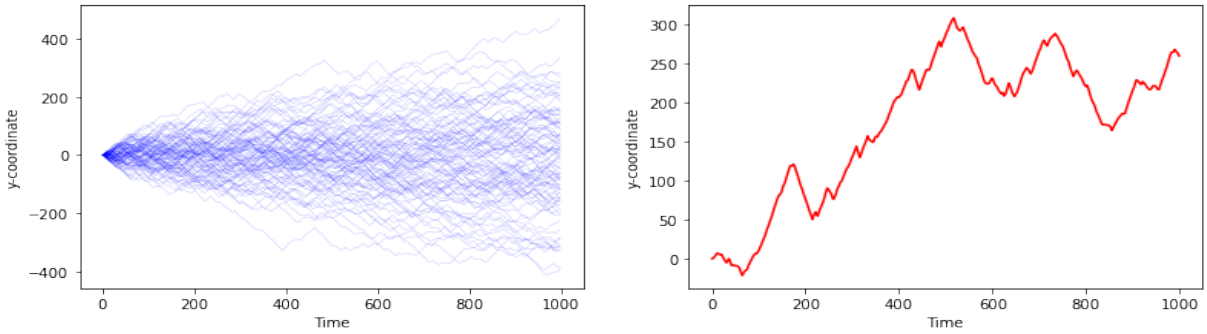
Table 3.5: Model checking results using the diffusion coefficient as a test statistic where $T(\mathbf{y}) = 17.74$.



(a) Plot of the x-coordinates for the replicated datasets for $\Delta_t = 0.1$.

(b) Plot of the x-coordinates for the observed dataset, dataset 1 for $\Delta_t = 0.1$.

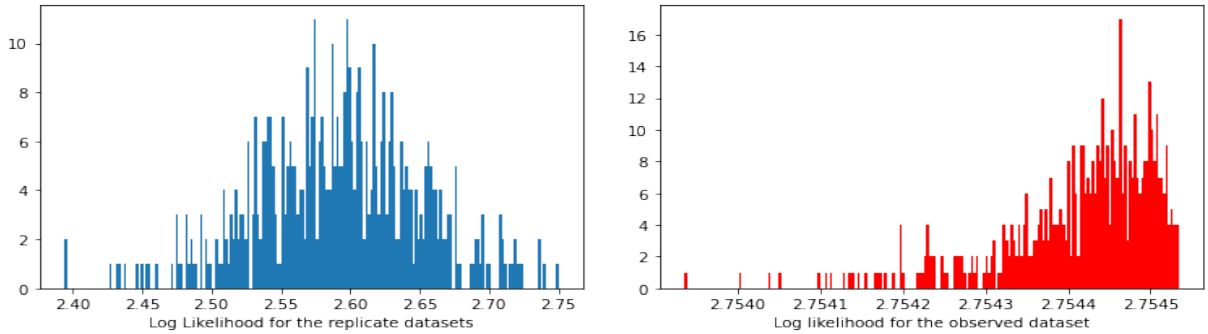
Figure 3.17: Plots of the x-coordinates for the replicated and the observed datasets for $\Delta_t = 0.1$.



(a) Plot of the y-coordinates for the replicated datasets for $\Delta_t = 0.1$.

(b) Plot of the y-coordinates for the observed dataset, dataset 1 for $\Delta_t = 0.1$.

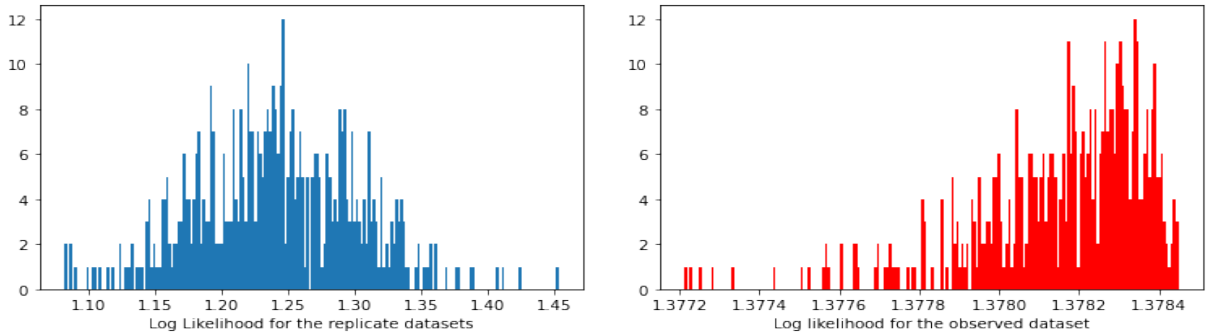
Figure 3.18: Plots of the y-coordinates for the replicated and the observed datasets for $\Delta_t = 0.1$.



(a) Scaled histogram of T(**y**_rep) for $\Delta_t = 0.1$.

(b) Scaled histogram of T(**y**) for $\Delta_t = 0.1$.

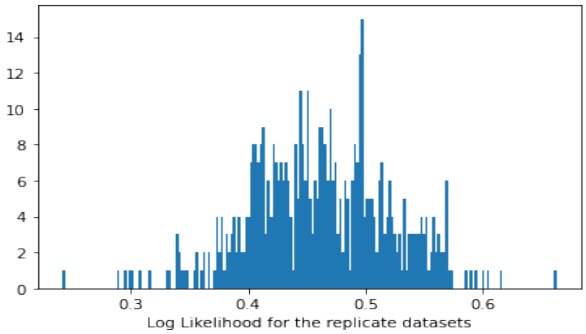
Figure 3.19: Model checking using log likelihood as a test statistic for $\Delta_t = 0.1$.



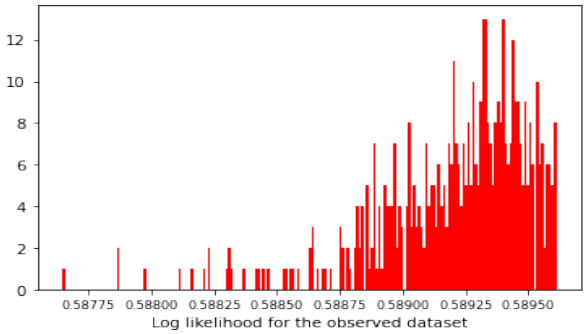
(a) Scaled histogram of T(**y**_rep) for $\Delta_t = 0.2$.

(b) Scaled histogram of T(**y**) for $\Delta_t = 0.2$.

Figure 3.20: Model checking using log likelihood as a test statistic for $\Delta_t = 0.2$.

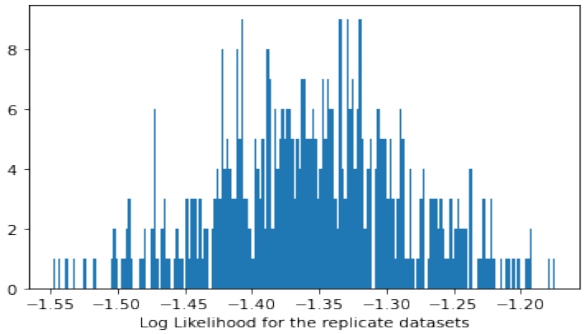


(a) Scaled histogram of T(**y**_rep) for $\Delta_t = 0.3$.

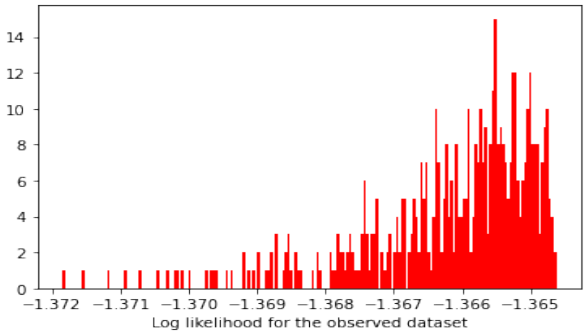


(b) Scaled histogram of T(**y**) for $\Delta_t = 0.3$.

Figure 3.21: Model checking using log likelihood as a test statistic for $\Delta_t = 0.3$.

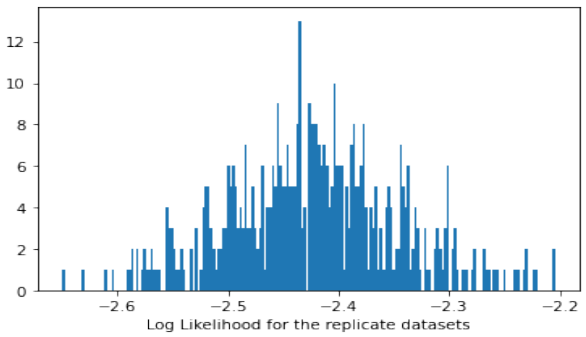


(a) Scaled histogram of T(**y**_rep) for $\Delta_t = 1$.

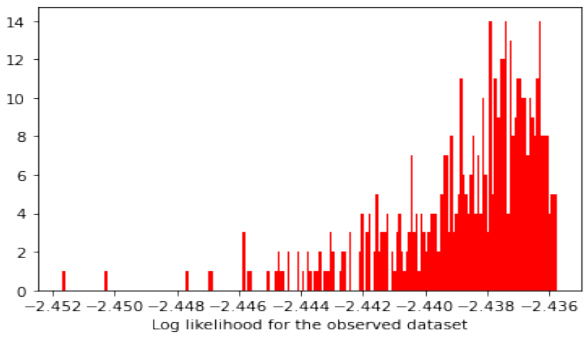


(b) Scaled histogram of T(**y**) for $\Delta_t = 1$.

Figure 3.22: Model checking using log likelihood as a test statistic for $\Delta_t = 1$.

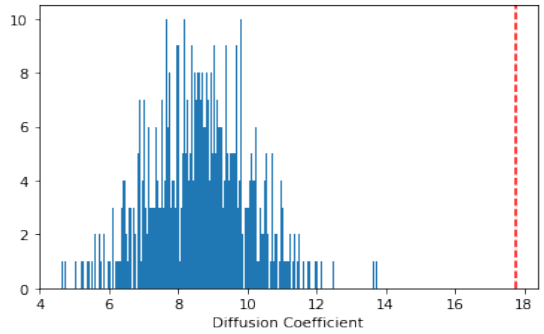


(a) Scaled histogram of T(**y**_rep) for $\Delta_t = 2$.

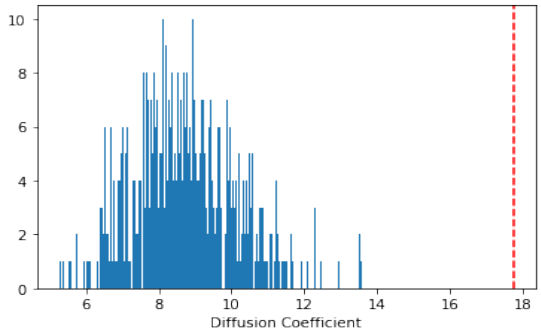


(b) Scaled histogram of T(**y**) for $\Delta_t = 2$.

Figure 3.23: Model checking using log likelihood as a test statistic for $\Delta_t = 2$.

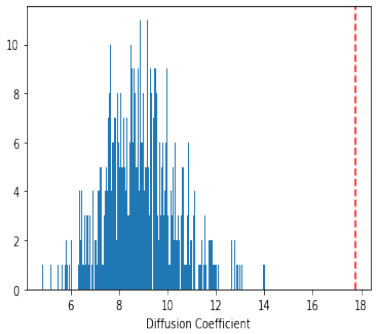


(a) Histogram of T(**y**_rep) and T(**y**) (red dashed line) for $\Delta_t = 0.1$.

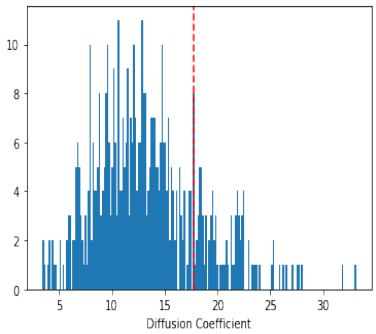


(b) Histogram of T(**y**_rep) and T(**y**) (red dashed line) for $\Delta_t = 0.2$.

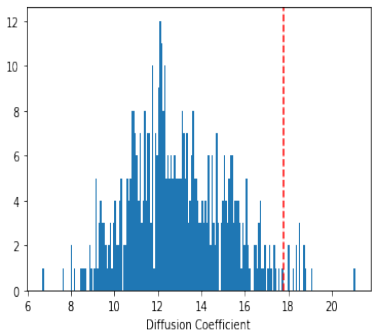
Figure 3.24: Model checking using the diffusion coefficient as a test statistic.



(a) Histogram of T(**y**_rep) and T(**y**) (red dashed line) for $\Delta_t = 0.3$.



(b) Histogram of T(**y**_rep) and T(**y**) (red dashed line) for $\Delta_t = 1.0$.



(c) Histogram of T(**y**_rep) and T(**y**) (red dashed line) for $\Delta_t = 2$.

Figure 3.25: Model checking using the diffusion coefficient as a test statistic.

In almost all of the cases we found out that there is a model mismatch between our inferred model and the observed data (dataset 1) illustrated in Tables 3.4-3.5 (extreme Bayesian p-values), in Figures 3.19-3.23 (using log likelihood), and in Figures 3.24-3.25 (using the diffusion coefficient). The only exception occurred when $\Delta_t = 2$ when using the log likelihood as test-statistic, illustrated in Table 3.4 (Bayesian p-value is 0.58) and in Figure 3.23. Also, when we fit a model with the same time step as the original data ($\Delta_t = 1$) we get that there is no evidence of a model mismatch illustrated in Tables 3.4 and 3.5 (Bayesian p-value is 0.566, respectively 0.188) and in Figures 3.22, 3.25b.

3.4 Conclusions

Mathematical modelling of animal movement is becoming ever more important in quantitative ecology, and with the improvement in GPS tagging technologies, increasing amounts of data are rapidly becoming available. This opens the door to statistical inference, to address the related challenges of parameter estimation, hypothesis testing and uncertainty quantification.

In the present chapter, we have analysed several discrete-time movement models, performed inference using a MCMC sampler and checked for convergence of the MCMC chains. From all of these models, we have focused on a model that is the centrepiece of the animal movement literature: a Markovian discrete-time random walk model. While very basic, this model is an essential building block from which many more advanced models are constructed by model extension. We have set up a Bayesian inference scheme based on sampling parameters from the posterior distribution. In order to be generalisable to more complex and advanced models, we have applied a general-purpose Markov chain Monte Carlo (MCMC) sampler. This addresses the challenges of parameter estimation (e.g. posterior means) and uncertainty quantification (credible intervals).

To address the challenge of hypothesis testing, we have generated synthetic data with a mismatch in the sampling frequency, to test procedures for detecting a systematic mismatch between the model and the data. In our study, we have evaluated a Bayesian approach to model critique based on summary statistics and posterior predictive p-values. The idea is to select an informative summary statistic and compute it from the original data. We then compare this value with the posterior distribution of the same summary statistic based on the model. This posterior distribution is obtained by drawing a sample of parameters from the posterior distribution, simulating data for each sampled parameter, and then computing the summary statistic for each simulated dataset. The posterior predictive p-value quantifies how far in the tails the summary statistic of the true data lies, with values close to 0 or 1 indicating a systematic model mismatch.

Any choice of summary statistic incurs an inevitable information loss, though, and our study has shown that the log likelihood is not sufficiently informative for consistent mismatch indication. The diffusion coefficient, on the other hand, has consistently indicated any model mismatch related to wrong sampling rates, and has thus turned out to be a more reliable model mismatch indicator. Besides testing Bayesian model critique procedures, our study has highlighted a fundamental shortcoming of discrete-time movement models, namely the need to select a sampling time interval (inverse sampling frequency) in advance. In the next chapters of the thesis (Chapters 5 and 6), we will apply Bayesian inference techniques to continuous time movement models, like Gaussian processes, which are more flexible and inherently avoid this limitation.

Chapter 4

Movement models and their covariance functions

Gaussian processes (GPs) are a powerful model that can detect patterns in the data and have been applied in various domains for decades. GPs have been applied in the field of geostatistics, where prediction is known as kriging, and have been widely implemented in the machine learning community. While being extensively used for a long time, applications of GPs are scattered across the literature. In Section 2.3, 'Relation between Gaussian processes and other models', we discussed the connections between non-parametric methods such as GPs and parametric models, convolutions of continuous-time movement models and state space models, thus connecting the many loose threads in the literature. In this chapter, we focus on representing popular continuous-time movement models such as Brownian bridge, Ornstein-Uhlenbeck (OU), Ornstein-Uhlenbeck velocity (OUV) and Ornstein-Uhlenbeck foraging (OUF) models as GPs, by deriving the appropriate and corresponding covariance function to the movement model. Thus, we gain significant advantages such as working in a non-parametric Bayesian inference framework with access to powerful machine learning libraries with already in-built inference methods such as maximum-a-posteriori (MAP), Markov Chain Monte Carlo (MCMC) or variational inference methods.

Authors' statement: Colin Torney, Dirk Husmeier and Ionut Paun designed the study, Ionut Paun performed the analysis and Ionut Paun wrote the manuscript. I confirm that my contribution to each section of the paper is more than 50%.

4.1 Introduction

In Section 2.3, 'Relation between Gaussian processes and other models', we discussed GPs and their various applications. Firstly, we looked at GPs from a machine learning perspective, where we showed that we can get from a parametric model to a non-parametric model such

as a GP. Secondly, we showed that convolutions of continuous-time movement models can be represented as a GP. Finally, we showed the advantages and disadvantages of representing a GP as a state space model and how we can convert a GP to a state space model and vice versa.

In this chapter, we expand on the background section 'Relation between Gaussian processes and other models' and show how different movement models formulated in continuous-time, such as Brownian bridge [Hooten et al., 2017], Orstein-Uhlenbeck (OU) [Uhlenbeck and Ornstein, 1930], Orstein-Uhlenbeck velocity model (OUV) [Johnson et al., 2008] and Orstein-Uhlenbeck-Foraging (OUF) [Fleming et al., 2014a] can be reformulated as GPs. Thus, the inference framework becomes non-parametric and Bayesian. Moreover, access to machine learning libraries that enable fast and efficient computational inference for large datasets is provided.

The theoretical covariance functions of the aforementioned movement models are mostly known and are found across the literature, the only exception being the OUV model, where we could not find the covariance function for the location process. However, sometimes the derivations of the covariance functions of these models are not found easily and are often incomplete or terse[1]. In this chapter we show the full derivation of these covariance functions in a simple and easy to understand manner. We first show the derivations of the covariance functions of the Brownian motion and Brownian bridge models[2], then in the case of the OUV and OUF models we offer two distinct derivations of their corresponding covariance functions for each model. For the last two models, in both cases, the first derivation makes use of the existing literature, then we expand on the existing literature to give a full derivation of their corresponding covariance functions. Furthermore, we illustrate alternative derivations of the covariance function for each model, that to the best of our knowledge are novel.

An important contribution is that we corrected the OUF covariance function formula in Fleming et al. [2014a], that had a different constant in front of the kernel and we showed thoroughly that it is a valid kernel i.e. is symmetric and positive semi-definite. Moreover, in Section 4.3 we run empirical tests to show that all our derivations for the theoretical covariance functions are correct by plotting the theoretical covariance function values against the numerical covariance function values obtained from simulating the corresponding movement model.

4.2 Movement models as Gaussian processes

It has been noted that GP regression (also known as kriging) is formally equivalent to many continuous-time movement modelling approaches [Hooten and Johnson, 2017, Fleming et al., 2014b]. Assuming that animal movement data is generated from a particular stochastic model,

[1] In the case of the Brownian motion and Brownian bridge covariance functions one might find various derivations across the literature. For the OU model, we found an accessible yet incomplete derivation on Wikipedia, however this source might not be always reliable. For the OUF model in the Supplemental Material [Fleming et al., 2014a], there is a terse derivation of the covariance function.

[2] We have never seen this derivation of the covariance function of the Brownian bridge model done before in the literature.

then inferring the parameters of that model is equivalent to placing a GP prior on the data with a particular covariance kernel. We detail here how common continuous-time movement models may be implemented within a GP framework by specifying the appropriate covariance kernel.

4.2.1 Brownian bridge movement process

A Brownian bridge process is a Brownian motion process where the starting/end times and locations are known and fixed in advance. Given that $f(t)$ is the position variable, Hooten et al. [2017] describe the Brownian bridge as multivariate normal random process such that

$$f(t) \sim \mathcal{N}\left(f(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}}(f(t_i) - f(t_{i-1})), \frac{(t - t_{i-1})(t_i - t)}{t_i - t_{i-1}}\sigma^2\right), \tag{4.1}$$

for $t_{i-1} < t < t_i$, where $f(t_{i-1})$, $f(t_i)$ are known and σ^2 is the variance.

For a start, we aim to derive the Brownian motion model's covariance function. We suppose that the position of a particle undergoing Brownian motion at time t is

$$x_t = \sigma \int_0^t dW_u, \tag{4.2}$$

where σ is the spread and W_u is the Wiener process. Likewise, at time s we have

$$x_s = \sigma \int_0^s dW_v, \tag{4.3}$$

where σ is the spread and W_v is the Wiener process at time v. Given that the mean of a Brownian motion process is 0, the covariance at the times t and s is[3]

$$\mathrm{Cov}\,(x_s, x_t) = \sigma^2 \mathbb{E}\,(x_s x_t) = \sigma^2 \mathbb{E}\left(\int_0^t dW_u \int_0^s dW_v\right). \tag{4.4}$$

Using the isometric property of the Itô integral [Protter, 2004] we get

$$\mathrm{Cov}\,(x_s, x_t) = \sigma^2 \mathbb{E}\left(\int_0^{\min(s,t)} du\right) = \sigma^2 \mathbb{E}[\min(s,t)] = \sigma^2 \min(s,t). \tag{4.5}$$

[3] In this thesis, we make use of both notations, $x(s)$ or x_s for any s interchangeably.

More explicitly, assuming $t < s$ without loss of generality, we get

$$\begin{aligned}
\mathbb{E}\left(\int_0^t dW_u \int_0^s dW_v\right) &= \mathbb{E}\left(\int_0^t dW_u \left(\int_0^t dW_v + \int_t^s dW_v\right)\right) \\
&= \mathbb{E}\left(\int_0^t dW_u \int_0^t dW_v + \int_0^t dW_u \int_t^s dW_v\right) \\
&= \mathbb{E}\left(\int_0^t dW_u \int_0^t dW_v\right) + \mathbb{E}\left(\int_0^t dW_u \int_t^s dW_v\right) \\
&= \mathbb{E}\left(\int_0^t du\right) + 0 = \mathbb{E}(t) = t = \min(s,t),
\end{aligned} \tag{4.6}$$

where the first expectation is computed using properties of the Itô integral [Protter, 2004] and the second integral is 0 due to properties of Brownian motion as there is no overlap between the two integrals.

Now we aim to prove that given the covariance function in Equation 4.5 we can arrive at the Brownian bridge movement model in Equation 4.1. We suppose that our observed data is the set $\mathbf{x} = x_1, \ldots, x_N$ and $f_i = f(x_i)$ is the function evaluated at x_i. We assume that we do not have any observation noise and we wish to predict the function values $\mathbf{f}^*$ at a new set of test points $\mathbf{x}^*$. If the observations are noiseless, then our GP will return the answer $f(\mathbf{x})$ with no uncertainty for an already seen set of observations $\mathbf{x}$.

Using the definition of the GP, the joint distribution of the GP has the following form

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}^* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}^* \\ \mathbf{K}^{*^T} & \mathbf{K}^{**} \end{pmatrix}\right), \tag{4.7}$$

where $\mathbf{K} = k(\mathbf{x},\mathbf{x})$ is $N \times N$, $\mathbf{K}^* = k(\mathbf{x},\mathbf{x}^*)$ is $N \times N^*$ and $\mathbf{K}^{**} = k(\mathbf{x}^*,\mathbf{x}^*)$ is $N^* \times N^*$ and k is a kernel. Using standard rules for conditioning Gaussians[4] [Rasmussen and Williams, 2006], the posterior has the following form

$$p(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{f}) = \mathcal{N}(\mathbf{f}^*|\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*). \tag{4.8}$$

$$\boldsymbol{\mu}^* = \mu(\mathbf{x}^*) + \mathbf{K}^{*^T}\mathbf{K}^{-1}(\mathbf{f} - \mu(\mathbf{x})). \tag{4.9}$$

$$\boldsymbol{\Sigma}^* = \mathbf{K}^{**} - \mathbf{K}^{*^T}\mathbf{K}^{-1}\mathbf{K}^*, \tag{4.10}$$

where we use the following convention regarding notation: $\boldsymbol{\mu} = \mu(\mathbf{x})$.

Suppose we have 3 observations at different time points, s, m and t, with $s < m < t$, where s and t are training points and m is a new test point. Using the formulas from Equations 4.5, 4.9 and 4.10 we calculate the mean and covariance matrix for the Brownian bridge movement

[4]Note that the definition of a GP and the standard rules for conditioning Gaussians have been introduced before in the thesis, in Section 2.2, but have been replicated here for the clarity of the derivations.

process as follows

$$\mathbf{K}^* = \sigma^2 \begin{pmatrix} \min(s,m). \\ \min(m,t) \end{pmatrix} = \sigma^2 \begin{pmatrix} s \\ m \end{pmatrix}. \tag{4.11}$$

$$\mathbf{K} = \sigma^2 \begin{pmatrix} \min(s,s) & \min(s,t) \\ \min(t,s) & \min(t,t). \end{pmatrix} = \sigma^2 \begin{pmatrix} s & s \\ s & t \end{pmatrix}. \tag{4.12}$$

$$\mathbf{K}^{**} = \sigma^2 \min(m,m) = \sigma^2 m. \tag{4.13}$$

We can assume that the means of the observations are zero. Following the formulas from Equations 4.9 and 4.10 we get that

$$\begin{aligned}
\mu^*(m) &= 0 + \begin{pmatrix} s & m \end{pmatrix} \begin{pmatrix} s & s \\ s & t \end{pmatrix}^{-1} \left(\begin{pmatrix} f(s) & f(t) \end{pmatrix}^T - \mathbf{0} \right) = \begin{pmatrix} s & m \end{pmatrix} \frac{1}{st - s^2} \begin{pmatrix} t & -s \\ -s & s \end{pmatrix} \begin{pmatrix} f(s) \\ f(t) \end{pmatrix} \\
&= \frac{1}{t-s} \begin{pmatrix} t-m & m-s \end{pmatrix} \begin{pmatrix} f(s) \\ f(t) \end{pmatrix} = \frac{1}{t-s} \left((t-m)f(s) + (m-s)f(t) \right) \\
&= \frac{1}{t-s} \left(f(s)t - mf(s) + mf(t) - sf(t) \right) = \frac{1}{t-s} m((f(t) - f(s)) + f(s)t - sf(t)) \\
&= \frac{1}{t-s} \left(m(f(t) - f(s)) + f(s)t - sf(t) + sf(s) - sf(s) \right) \\
&= \frac{1}{t-s} \left((m-s)(f(t) - f(s)) - sf(s) + tf(s) \right) = \frac{1}{t-s} \left((m-s)(f(t) - f(s)) + f(s)(t-s) \right) \\
&= f(s) + \frac{(m-s)(f(t) - f(s))}{t-s}.
\end{aligned} \tag{4.14}$$

We repeat the same process for the covariance matrix at time point m

$$\begin{aligned}
\mathbf{\Sigma}^*(m) &= \sigma^2 \left(m - \begin{pmatrix} s & m \end{pmatrix} \begin{pmatrix} s & s \\ s & t \end{pmatrix}^{-1} \begin{pmatrix} s \\ m \end{pmatrix} \right) \\
&= \sigma^2 \left(m - \begin{pmatrix} s & m \end{pmatrix} \frac{1}{st - s^2} \begin{pmatrix} t & -s \\ -s & s \end{pmatrix} \begin{pmatrix} s \\ m \end{pmatrix} \right) \\
&= \sigma^2 \left(m - \frac{1}{st - s^2} \begin{pmatrix} st - ms & -s^2 + sm \end{pmatrix} \begin{pmatrix} s \\ m \end{pmatrix} \right) \\
&= \sigma^2 \left(m - \frac{1}{t-s} (st - 2ms + m^2) \right) = \sigma^2 \left(\frac{mt - sm - st + 2ms - m^2}{t-s} \right) \\
&= \sigma^2 \frac{mt - st + ms - m^2}{t-s} = \sigma^2 \frac{t(m-s) + m(s-m)}{t-s} = \sigma^2 \frac{(t-m)(m-s)}{t-s}.
\end{aligned} \tag{4.15}$$

We have that $f(m) \sim \mathcal{N}(\mu^*(m), \mathbf{\Sigma}^*(m))$. This is equivalent to Equation 4.1 and this is the form of the Brownian bridge process that is common in the literature [Hooten et al., 2017]. The

Brownian bridge model covariance function at two different time points will be presented in the Appendix, Section C.1.

4.2.2 Ornstein-Uhlenbeck process

The stochastic differential equation (SDE) for an OU process in one-dimension is

$$dx_t = a(b - x_t)dt + \sigma dW_t, \tag{4.16}$$

where W_t is a Wiener process, a is the rate at which the process mean reverts, b is the run average and σ is the volatility of the process.

We illustrate the derivation of the covariance function for the OU model. Changing variables $f(x_t, t) = x_t e^{at}$ we get

$$df(x_t,\ t) = ax_t e^{at}dt + e^{at}dx_t = e^{at}abdt + \sigma e^{at}dW_t, \tag{4.17}$$

using the formula for the OU stochastic differential Equation 4.16 in the last equation. Integrating from 0 to t we get

$$x_t e^{at} = x_0 + \int_0^t e^{as}ab\ ds + \int_0^t \sigma e^{as}dW_s. \tag{4.18}$$

Therefore,

$$x_t = x_0 e^{-at} + b(1 - e^{-at}) + \sigma \int_0^t e^{-a(t-s)}dW_s \tag{4.19}$$

Then, from the previous equation we calculate the expectation such that

$$\mathbb{E}(x_t) = x_0 e^{-at} + b(1 - e^{-at}). \tag{4.20}$$

The term $\sigma \int_0^t e^{-a(t-s)}dW_s$ disappears because of properties of Brownian motion as the expectation $\mathbb{E}(W_t) = 0$.

Using Itô's isometry property [Protter, 2004] we have that (we assume $s < t$ withous loss of generality)

$$\begin{aligned}
\mathrm{Cov}(x_s, x_t) &= \mathbb{E}[(x_s - \mathbb{E}(x_s))(x_t - \mathbb{E}(x_t)] \\
&= \mathbb{E}\left(\int_0^s \sigma e^{a(u-s)}dW_u \int_0^t \sigma e^{a(v-t)}dW_v\right) \\
&= \sigma^2 e^{-a(s+t)}\mathbb{E}\left(\int_0^s e^{au}dW_u \int_0^t e^{av}dW_v\right) \\
&= \sigma^2 e^{-a(s+t)}\mathbb{E}\left(\int_0^s e^{au}dW_u \left(\int_0^s e^{av}dW_v + \int_s^t e^{av}dW_v\right)\right) \\
&= \sigma^2 e^{-a(s+t)}\left(\mathbb{E}\left(\int_0^s e^{au}dW_u \int_0^s e^{av}dW_v\right) + \mathbb{E}\left(\int_0^s e^{au}dW_u \int_s^t e^{av}dW_v\right)\right).
\end{aligned} \tag{4.21}$$

We compute the first expectation using properties of the Itô integral [Protter, 2004] and the second expectation is 0, due to properties of Brownian motion, as there is no overlap between the two integrals.

$$\begin{aligned}
\text{Cov}(x_s, x_t) &= \sigma^2 e^{-a(s+t)} \mathbb{E}\left(\int_0^s e^{au} dW_u \int_0^s e^{au} dW_u\right) \\
&= \sigma^2 e^{-a(s+t)} \int_0^s e^{2au} du = \sigma^2 e^{-a(s+t)} \frac{e^{2au}}{2a}\Bigg|_0^s \\
&= \frac{\sigma^2}{2a} e^{-a(s+t)} \left(e^{2as} - 1\right) = \frac{\sigma^2}{2a} e^{-a(s+t)} \left(e^{2a\min(s,t)} - 1\right) \\
&= \frac{\sigma^2}{2a} \left(e^{-a|t-s|} - e^{-a(t+s)}\right).
\end{aligned} \tag{4.22}$$

As t and s grow large $e^{-a(t+s)} \to 0$, therefore, we get that the OU covariance function is given by

$$\text{Cov}(x_s, x_t) = \frac{\sigma^2}{2a} e^{-a|t-s|}. \tag{4.23}$$

To get the final result we use the following identity: $2a\min(s,t) - a(s+t) = -a|s-t|$ derived from

$$|s-t| = \max(s,t) - \min(s,t). \tag{4.24}$$

$$s+t = \max(s,t) + \min(s,t). \tag{4.25}$$

4.2.3 Orstein-Uhlenbeck velocity model

The OUV model or the continuous-time correlated random walk (CTCRW) is a animal movement model made popular by Johnson et al. [2008]. We denote x_t the location of the animal at time t, and v_t the velocity is given the following equation in one-dimension

$$dx_t = v_t dt. \tag{4.26}$$

We then model the velocity by an OU process, given by the following equation in one-dimension such that

$$dv_t = a(b - v_t)dt + \sigma dW_t, \tag{4.27}$$

where W_t is a Wiener process, a is mean reversion rate, b is the average, σ is the volatility of the process that measures the deviation of the velocity around the mean.

We are interested in calculating the covariance function for the location process $\mathbf{x}$ at time t. We can then use this covariance function as the kernel of a GP, instead of working with stochastic differential equations. We give two approaches to calculating the covariance function of an OUV model. The first derivation is based on Michelot and Blackwell [2019] and the second derivation

of the covariance function of the OUV model is novel. We note that the first method is more general, done by solving the stochastic differential equations and the second method uses the fact that we already know the formula for the OU covariance function.

OUV model covariance derivation: First derivation

For the first derivation we use Michelot and Blackwell [2019], Appendix S1. The OUV model in one-dimension is defined as

$$dx_t = v_t dt. \tag{4.28}$$

$$dv_t = -av_t dt + \sigma dW_t, \tag{4.29}$$

where x_t is the location process, v_t is the velocity process, a is the mean-reversion rate, the average is taken to be zero, σ is the volatility of the process and W_t is a Wiener process. For simplicity, we work with the univariate case. We multiply Equation 4.29 by e^{at} such that

$$e^{at} dv_t = -ae^{at} v_t dt + e^{at} \sigma dW_t. \tag{4.30}$$

We notice that

$$d(e^{at} v_t) = ae^{at} v_t dt + e^{at} dv_t. \tag{4.31}$$

Adding Equations 4.30 and 4.31 we get that

$$d(e^{at} v_t) = e^{at} \sigma dW_t. \tag{4.32}$$

We integrate both sides between t and $t+\delta$,

$$e^{a(t+\delta)} v_{t+\delta} - e^{at} v_t = \sigma \int_{s=t}^{t+\delta} e^{as} dW_s. \tag{4.33}$$

Solving this we get the solution

$$v_{t+\delta} = e^{-a\delta} v_t + \sigma \int_{s=t}^{t+\delta} e^{-a(t+\delta-s)} dW_s. \tag{4.34}$$

To solve for x_t we integrate Equation 4.28 between t and $t+\delta$ and using Equation 4.34 we get that

$$x_{t+\delta} - x_t = \int_{s=t}^{t+\delta} v_s ds = \int_{s=t}^{t+\delta} \left(e^{-a(s-t)} v_t + \sigma \int_{u=t}^{s} e^{-a(s-u)} dW_u \right) ds. \tag{4.35}$$

Therefore,

$$\begin{aligned} x_{t+\delta} &= x_t + v_t \int_{s=t}^{t+\delta} e^{-a(s-t)} ds + \sigma \int_{s=t}^{t+\delta} \int_{u=t}^{s} e^{-a(s-u)} dW_u ds \\ &= x_t + v_t \left[-\frac{e^{-a(s-t)}}{a} \right]_{s=t}^{t+\delta} + \sigma \int_{u=t}^{t+\delta} \int_{s=u}^{t+\delta} e^{-a(s-u)} ds dW_s \\ &= x_t + \left(\frac{1-e^{-a\delta}}{a} \right) v_t + \sigma \int_{u=t}^{t+\delta} \left[-\frac{e^{-a(s-u)}}{a} \right]_{s=u}^{t+\delta} dW_u \\ &= x_t + \left(\frac{1-e^{-a\delta}}{a} \right) v_t + \frac{\sigma}{a} \int_{u=t}^{t+\delta} (1-e^{-a(t+\delta-u)}) dW_u. \end{aligned} \tag{4.36}$$

We denote the Gaussian error term as[5]

$$\xi(\delta) = \frac{\sigma}{a} \int_{u=t}^{t+\delta} (1-e^{-a(t+\delta-u)}) dW_u. \tag{4.37}$$

The derivation so far has been reproduced from Michelot and Blackwell [2019]. Now we aim to use the Gaussian error term from Equation 4.37 to calculate the covariance function for the location process x_t. We do this in a similar manner to the calculation of the covariance of the location process and the velocity process found in Michelot and Blackwell [2019], page 15. For simplicity, we choose the times to be 0 and δ. The Gaussian error term for the location process of the velocity model between the times 0 and δ, respectively δ', where a and σ are parameters of the CTCRW model is

$$\xi(\delta) = \frac{\sigma}{a} \int_{0}^{\delta} (1-e^{-a(\delta-u')}) dW_{u'}. \tag{4.38}$$

$$\xi(\delta') = \frac{\sigma}{a} \int_{0}^{\delta'} (1-e^{-a(\delta'-v')}) dW_{v'}. \tag{4.39}$$

The covariance function at times δ and δ' is

$$\begin{aligned} \text{Cov}[\xi(\delta), \xi(\delta')] &= \mathbb{E}[(\xi(\delta) - \mathbb{E}[\xi(\delta])(\xi(\delta') - \mathbb{E}[\xi(\delta')])] \\ &= \mathbb{E}[\xi(\delta)\xi(\delta')] - \mathbb{E}[\xi(\delta)]\mathbb{E}[\xi(\delta')]. \end{aligned} \tag{4.40}$$

[5]Note that in the article there is a plus sign, however there should be a minus sign, given the following equations on page 14 and 15 of Michelot and Blackwell [2019].

Using properties of the Itô integral [Protter, 2004] and the notation $\gamma = \min(\delta, \delta')$ we get that

$$\begin{aligned}
\mathbb{E}[\xi(\delta)\xi(\delta')] &= \frac{\sigma^2}{a^2}\int_0^{\gamma}(1-e^{-a(\delta-u')})(1-e^{-a(\delta'-u')})du' \\
&= \frac{\sigma^2}{a^2}\int_0^{\gamma} 1 - e^{-a(\delta-u')} - e^{-a(\delta'-u')} + e^{-a(\delta'+\delta-2u')}du' \\
&= \frac{\sigma^2}{a^2}\left(\gamma - \frac{e^{-a(\delta-\gamma)}}{a} - \frac{e^{-a(\delta'-\gamma)}}{a} + \frac{e^{-a(\delta+\delta'-2\gamma)}}{2a} - 0 + \frac{e^{-a(\delta-0)}}{a} + \frac{e^{-a(\delta'-0)}}{a}\right. \\
&\qquad \left. - \frac{e^{-a(\delta+\delta'-0)}}{2a}\right) \\
&= \frac{\sigma^2}{a^2}\left(\gamma - \frac{e^{-a(\delta-\gamma)} + e^{-a(\delta'-\gamma)} - e^{-a\delta} - e^{-a\delta'}}{a} + \frac{e^{-a(\delta+\delta'-2\gamma)} - e^{-a(\delta+\delta')}}{2a}\right) \\
&= \frac{\sigma^2}{a^2}\left(\gamma - \frac{2\left(e^{-a(\delta-\gamma)} + e^{-a(\delta'-\gamma)} - e^{-a\delta} - e^{-a\delta'}\right) - e^{-a(\delta+\delta'-2\gamma)} + e^{-a(\delta+\delta')}}{2a}\right).
\end{aligned} \tag{4.41}$$

We can simplify the above expression. We can assume without loss of generality that $\delta \leq \delta'$ i.e. $\gamma = \min(\delta, \delta') = \delta$. Therefore, Equation 4.41 becomes

$$\begin{aligned}
\mathbb{E}[\xi(\delta)\xi(\delta')] &= \frac{\sigma^2}{2a^3}\left(2a\delta - 2\left(e^0 + e^{-a(\delta'-\delta)} - e^{-a\delta} - e^{-a\delta'}\right) + e^{-a(\delta'-\delta)} - e^{-a(\delta+\delta')}\right) \\
&= \frac{\sigma^2}{2a^3}\left(2a\delta - 2 - e^{-a(\delta'-\delta)} + 2e^{-a\delta} + 2e^{-a\delta'} - e^{-a(\delta+\delta')}\right) \\
&= \frac{\sigma^2}{2a^3}\left(2e^{-a\delta} - e^{-a(\delta+\delta')} - e^{-a(\delta'-\delta)} + 2e^{-a\delta'} + 2a\delta - 2\right) \\
&= \frac{\sigma^2}{2a^3}\left(2e^{-a\delta} - e^{-a(\delta+\delta')} - e^{-a|\delta'-\delta|} + 2e^{-a\delta'} + 2a\min(\delta,\delta') - 2\right).
\end{aligned} \tag{4.42}$$

Due to symmetry we can insert modulus $|\delta - \delta'|$ in the last equation in Equation 4.42. Using the expectation formula from Michelot and Blackwell [2019], page 14, we have

$$\mathbb{E}[\xi(\delta)] = x_0 + \left(\frac{1-e^{-a\delta}}{a}\right)v_0. \tag{4.43}$$

$$\mathbb{E}[\xi(\delta')] = x_0 + \left(\frac{1-e^{-a\delta'}}{a}\right)v_0, \tag{4.44}$$

where x_t is the location process and v_t is the velocity process at time t.

Plugging everything into Equation 4.40 we get the covariance formula for the location process for a general mean. However, if we assume that the mean of the process is 0 by taking

$x_0 = v_0 = 0$, the location process' covariance function for the velocity model is Equation 4.42. Reverting to the standard notation used in this chapter ($\xi \to x$, $\delta \to t$, $\delta' \to t'$), the covariance of the location process x of the OUV model at two time points t and t' is

$$\text{Cov}[x(t), x(t')] = \frac{\sigma^2}{2a^3}\left(2e^{-at} - e^{-a(t+t')} - e^{-a|t'-t|} + 2e^{-at'} + 2a\min(t,t') - 2\right), \tag{4.45}$$

where a is the mean reversion rate, σ is the volatility of the OU process for the velocity process v.

OUV model covariance derivation: Second approach

We suppose we have two observations recorded at positions x_t and $x_{t'}$ at times t and t' drawn from an OUV model. Given that we assume that the mean of the process is 0 we have that the covariance functions is

$$\begin{aligned}\text{Cov}[x(t), x(t')] = \mathbb{E}\left[x(t)x(t')\right] &= \mathbb{E}\left(\int_0^t v(s)ds \int_0^{t'} v(r)dr\right) \\ &= \int_0^t \int_0^{t'} \mathbb{E}\left[v(s)v(r)\right] dsdr = \int_0^t \int_0^{t'} \frac{\sigma^2}{2a}\left(e^{-a|s-r|} - e^{-a(s+r)}\right) dsdr.\end{aligned} \tag{4.46}$$

In Equation 4.46 we used the fact that we can move the expectation inside the integral and the fact that the velocity process is modelled by an OU model, which has a known covariance function derived in Equation 4.22 replicated here for clarity

$$\text{Cov}[v(s), v(t)] = \frac{\sigma^2}{2a}\left(e^{-a|t-s|} - e^{-a(t+s)}\right), \tag{4.47}$$

where v is the velocity process at times t and s, σ is the volatility of the process and a is the mean inversion rate.

We have $|s-r| = s-r$ if $r \leq s$ and $|s-r| = r-s$ if $r > s$. We assume $t < t'$ without loss of

generality. Equation 4.46 becomes

$$\begin{aligned}
\mathrm{Cov}[x(t),x(t')] &= \frac{\sigma^2}{2a}\int_0^t ds\left(\int_0^s e^{-a(s-r)}-e^{-a(s+r)}dr+\int_s^{t'} e^{-a(r-s)}-e^{-a(s+r)}dr\right)\\
&= \frac{\sigma^2}{2a}\int_0^t ds\left(\frac{e^{-a(s-r)}}{a}\Big|_{r=0}^{r=s}+\frac{e^{-a(s+r)}}{a}\Big|_{r=0}^{r=s}-\frac{e^{-a(r-s)}}{a}\Big|_{r=s}^{r=t'}+\frac{e^{-a(s+r)}}{a}\Big|_{r=s}^{r=t'}\right)\\
&= \frac{\sigma^2}{2a}\int_0^t ds\Bigg(\left(\frac{1}{a}-\frac{e^{-as}}{a}\right)+\left(\frac{e^{-2as}}{a}-\frac{e^{-as}}{a}\right)-\left(\frac{e^{-a(t'-s)}}{a}-\frac{1}{a}\right)\\
&\qquad\qquad +\left(\frac{e^{-a(t'+s)}}{a}-\frac{e^{-2as}}{a}\right)\Bigg)\\
&= \frac{\sigma^2}{2a^2}\int_0^t\left(-2e^{-as}+e^{-a(t'+s)}-e^{-a(t'-s)}+2\right)ds\\
&= \frac{\sigma^2}{2a^2}\left(2\frac{e^{-as}}{a}\Big|_{s=0}^{s=t}-\frac{e^{-a(t'+s)}}{a}\Big|_{s=0}^{s=t}-\frac{e^{-a(t'-s)}}{a}\Big|_{s=0}^{s=t}+2t\right)\\
&= \frac{\sigma^2}{2a^3}\left(2\left(e^{-at}-1\right)-\left(e^{-a(t+t')}-e^{-at'}\right)-\left(e^{-a(t'-t)}-e^{-at'}\right)+2ta\right)\\
&= \frac{\sigma^2}{2a^3}\left(2e^{-at}-2-e^{-a(t+t')}-e^{-a(t'-t)}+2e^{-at'}+2ta\right)\\
&= \frac{\sigma^2}{2a^3}\left(2e^{-at}-e^{-a(t+t')}-e^{-a|t'-t|}+2e^{-at'}+2a\min(t,t')-2\right),
\end{aligned} \tag{4.48}$$

where due to symmetry we can insert the modulus $|t-t'|$ in the last equation. In conclusion, the two approaches (Equations 4.45 and 4.48) of determining the covariance function of the location process x at two times t and t' for the velocity model give the same result

$$\mathrm{Cov}[x(t),x(t')] = \frac{\sigma^2}{2a^3}\left(2e^{-at}-e^{-a(t+t')}-e^{-a|t'-t|}+2e^{-at'}+2a\min(t,t')-2\right). \tag{4.49}$$

4.2.4 OU-Foraging model

The OU-Foraging model (OUF) is a generalisation of the OU model derived by Fleming et al. [2014a] that adds random foraging periods to the OU model. These periods are regulated by the introduction of another time-scale parameter τ_F, that corresponds to foraging behaviour. The OUF corresponding covariance function at times t and t' is given by the following formula

$$k(t,t') = \sigma_H\frac{\tau_H e^{-\frac{|t-t|'}{\tau_H}}-\tau_F e^{-\frac{|t-t'|}{\tau_F}}}{\tau_H-\tau_F}, \tag{4.50}$$

where σ_H is the position variance parameter and τ_H, τ_F denote the time-scales parameters.

First of all, in order to use this model, the OUF covariance function given by Equation 4.50

must be positive-semidefinite. While the OUF kernel is a popular kernel, in Fleming et al. [2014a] the explanation of why it is a positive-semidefinite and its derivation are terse. Firstly, we thoroughly prove the positive-semidefiniteness of the OUF kernel and then we illustrate a novel and complete derivation of the OUF model's covariance function.

The covariance matrix of a random vector $\mathbf{x} \in \mathbf{R}^n$ with mean vector $\mathbf{m}$ is defined as

$$\mathbf{\Sigma} = \mathbb{E}\left[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^T\right]. \tag{4.51}$$

The $(i,j)^{\text{th}}$ element of the covariance matrix $\mathbf{\Sigma}$ is given by

$$\mathbf{\Sigma}_{ij} = \mathbb{E}\left[(x_i - m_i)\left(x_j - m_j\right)\right] = k_{ij}. \tag{4.52}$$

Any covariance matrix has various properties, including the fact that it is symmetric and that it is positive-semidefinite. Therefore, to prove that the OUF kernel gives a positive-semidefinite covariance matrix we prove that starting from the definition given by Equations 4.51 or 4.52 we obtain the OUF kernel formula.

Following the Supplementary Material from Fleming et al. [2014a], Section C.3, and using the authors' notation for simplicity, we can derive the OUF covariance function from the Langevin equations

$$\frac{d}{dt}x(t) = -\frac{1}{\tau_H}(x(t)-\mu) + u(t). \tag{4.53}$$

$$\frac{d}{dt}u(t) = -\frac{1}{\tau_F}u(t) + a(t), \tag{4.54}$$

where the x-movement is driven by the process $u(t)$, which itself is driven by the white noise process $a(t)$. We can solve Equation 4.53 and construct the covariance matrix by using the OU derivation presented in the Supplementary Material by Fleming et al. [2014a], Section C.2. Equation C.21 from Fleming et al. [2014a] (Equation 4.55), illustrates the covariance matrix for the OUF kernel

$$\mathbf{\Sigma}(t,t') = \int_{-\infty}^{t} ds \int_{-\infty}^{t'} ds' e^{-\frac{t-s}{\tau_H}} e^{-\frac{t'-s'}{\tau_H}} \mathbb{E}\left[u(s)u(s')\right]. \tag{4.55}$$

Equation 4.55 is obtained by using Equation 4.52, the definition of the expectation and Equation C.20 [Fleming et al., 2014a], replicated below for clarity

$$\lim_{t_0 \to -\infty} x(t) - \mu = \int_{-\infty}^{t} dt' e^{-\frac{t-t'}{\tau_H}} u(t'), \tag{4.56}$$

where we set the arbitrary initial conditions $x(t_0) = x(0)$. Using Equation 4.54 and Equations C.23, C.28 [Fleming et al., 2014a] we obtain Equation C.29 [Fleming et al., 2014a] (Equation

4.57)

$$\mathbf{\Sigma}(t,t') = \int_{-\infty}^{t} ds \int_{-\infty}^{t'} ds' e^{-\frac{t-s}{\tau_H}} e^{-\frac{t'-s'}{\tau_H}} \frac{\sigma_a \tau_F}{2} e^{-\frac{|s-s'|}{\tau_F}}, \tag{4.57}$$

where Equations C.23 and C.28 have the following formulas

$$\mathbf{\Sigma}_{\mathrm{OU}}(t,t') = \frac{\sigma_u \tau_H}{2} e^{-\frac{|t-t'|}{\tau_H}}. \tag{4.58}$$

$$\mathbb{E}\left[a(t)a(t')\right] = \sigma_a \delta(t-t') \ (\delta(t)\text{is the Dirac delta distribution}). \tag{4.59}$$

We aim to prove that from Equation 4.57 we get to Equation C.30 [Fleming et al., 2014a] (Equation 4.60), which is Fleming et al. [2014a]'s final formula for the OUF covariance function

$$\mathbf{\Sigma}(t,t') = \frac{\sigma_a \tau_H \tau_F}{2} \frac{\tau_H e^{-\frac{-|t-t'|}{\tau_H}} - \tau_F e^{-\frac{|t-t'|}{\tau_F}}}{\tau_H - \tau_F}. \tag{4.60}$$

We explain how to get to this result in greater detail. For simplification we denote $\tau_H = H$, $\tau_F = F$ and $\sigma_a = a$. We can assume $s < t'$ without loss of generality, and as the integral in Equation 4.57 is symmetric we have that $t < t'$ as well. We have that

$$\int_{-\infty}^{t'} e^{-\frac{t'-s'}{H}} e^{-\frac{|s-s'|}{F}} ds' = \int_{-\infty}^{s} e^{-\frac{t'-s'}{H}} e^{-\frac{s-s'}{F}} ds' + \int_{s}^{t'} e^{-\frac{t'-s'}{H}} e^{-\frac{s'-s}{F}} ds'. \tag{4.61}$$

We denote

$$\begin{aligned} A &= \int_{-\infty}^{s} e^{\frac{s'-t'}{H}} e^{\frac{s'-s}{F}} ds' = \int_{-\infty}^{s} e^{\frac{(H+F)s'-Ft'-Hs}{HF}} ds' \\ &= \frac{HF}{H+F} e^{\frac{(H+F)s'-Ft'-Hs}{HF}} \Bigg|_{-\infty}^{s} = \frac{HF}{H+F} e^{\frac{(H+F)s-Ft'-Hs}{HF}} \\ &= \frac{HF}{H+F} e^{\frac{s-t'}{H}}. \end{aligned} \tag{4.62}$$

Also, let

$$\begin{aligned} B &= \int_{s}^{t'} e^{\frac{s'-t'}{H}} e^{\frac{s-s'}{F}} ds' = \int_{s}^{t'} e^{\frac{(F-H)s'-Ft'+Hs}{HF}} ds' \\ &= \frac{HF}{F-H} e^{\frac{(F-H)s'-Ft'+Hs}{HF}} \Bigg|_{s}^{t'} = \frac{HF}{F-H} \left(e^{\frac{(F-H)t'-Ft'+Hs}{HF}} - e^{\frac{(F-H)s-Ft'+Hs}{HF}} \right) \\ &= \frac{HF}{F-H} \left(e^{\frac{s-t'}{F}} - e^{\frac{s-t'}{H}} \right). \end{aligned} \tag{4.63}$$

Equation 4.57 becomes

$$
\begin{aligned}
\mathbf{\Sigma}(t,t') &= \int_{-\infty}^{t} \frac{aF}{2} e^{\frac{s-t}{H}} \left(\frac{HF}{F+H} e^{\frac{s-t'}{H}} + \frac{HF}{F-H} \left(e^{\frac{s-t'}{F}} - e^{\frac{s-t'}{H}} \right) \right) ds \\
&= \frac{aF}{2} \int_{-\infty}^{t} \frac{HF}{H+F} e^{\frac{2s-t-t'}{H}} + \frac{HF}{F-H} \left(e^{\frac{s-t'}{F}+\frac{s-t}{H}} - e^{\frac{s-t'}{H}+\frac{s-t}{H}} \right) ds \\
&= \frac{aF}{2} \int_{-\infty}^{t} \frac{HF}{H+F} e^{\frac{2s-t-t'}{H}} + \frac{HF}{F-H} \left(e^{\frac{(H+F)s-Ht'-Ft}{FH}} - e^{\frac{2s-t-t'}{H}} \right) ds \\
&= \frac{aF}{2} \left(\int_{-\infty}^{t} \frac{HF}{H+F} e^{\frac{2s-t-t'}{H}} ds + \int_{-\infty}^{t} \frac{HF}{F-H} \left(e^{\frac{(H+F)s-Ht'-Ft}{FH}} - e^{\frac{2s-t-t'}{H}} \right) ds \right) \\
&= \frac{aF}{2} \left(\frac{HF}{H+F} \frac{H}{2} e^{\frac{2s-t-t'}{H}} \Big|_{-\infty}^{t} + \frac{HF}{F-H} \left(\frac{HF}{H+F} e^{\frac{(H+F)s-Ht'-Ft}{FH}} - \frac{H}{2} e^{\frac{2s-t-t'}{H}} \right) \Big|_{-\infty}^{t} \right) \\
&= \frac{aF}{2} \left(\frac{H^2F}{2(F+H)} e^{\frac{t-t'}{H}} + \frac{HF}{F-H} \left(\frac{FH}{H+F} e^{\frac{t-t'}{F}} - \frac{H}{2} e^{\frac{t-t'}{H}} \right) \right) \\
&= \frac{aHF}{2} \left(\frac{HF}{2(F+H)} e^{\frac{t-t'}{H}} + \frac{F^2H}{(F-H)(F+H)} e^{\frac{t-t'}{F}} - \frac{HF}{2(F-H)} e^{\frac{t-t'}{H}} \right) \\
&= \frac{aHF}{2} \left(\frac{HF(F-H) - HF(H+F)}{2(F+H)(F-H)} e^{\frac{t-t'}{H}} + \frac{F^2H}{(F-H)(H+F)} e^{\frac{t-t'}{F}} \right) \\
&= \frac{aHF}{2} \left(\frac{-H^2F}{(F+H)(F-H)} e^{\frac{t-t'}{H}} + \frac{F^2H}{(F+H)(F-H)} e^{\frac{t-t'}{F}} \right) \\
&= \frac{aHF}{2} \left(\frac{-H}{F-H} e^{\frac{t-t'}{H}} + \frac{F}{F-H} e^{\frac{t-t'}{F}} \right) \frac{HF}{F+H} \\
&= \frac{aHF}{2} \frac{HF}{F+H} \left(\frac{He^{\frac{t-t'}{H}} - Fe^{\frac{t-t'}{F}}}{H-F} \right) \\
&= \frac{aHF}{2} \frac{HF}{F+H} \left(\frac{He^{\frac{-|t-t'|}{H}} - Fe^{\frac{-|t-t'|}{F}}}{H-F} \right).
\end{aligned}
\tag{4.64}
$$

Due to the symmetry we can add modulus in the previous equation to make the kernel stationary. Reverting to the original notation we get that

$$
\begin{aligned}
\mathbf{\Sigma}(t,t') &= \frac{\sigma_a \tau_H^2 \tau_F^2}{2(\tau_F+\tau_H)} \left(\frac{\tau_H e^{\frac{-|t-t'|}{\tau_H}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}}}{\tau_H - \tau_F} \right) \\
&= \sigma_H \left(\frac{\tau_H e^{\frac{-|t-t'|}{\tau_H}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}}}{\tau_H - \tau_F} \right),
\end{aligned}
\tag{4.65}
$$

which is the same form as Equation 4.50.

We arrived at the same form as Equation C.30 [Fleming et al., 2014a] (Equation 4.60), except the constants in front of the kernel. Therefore, we proved rigorously that the OUF kernel is positive-semidefinite, given that we arrived at the OUF covariance function starting from the

general definition of a covariance matrix given in Equation 4.51, and we know that any covariance matrix is positive-semidefinite.

OU-Foraging model and its covariance function: Different derivation

We present a novel approach to derive the OUF covariance function. This new derivation is based on Särkkä et al. [2013]'s approach to convert a state space model to a covariance function. The methods of converting a state space model to a covariance function and vice-versa have been reviewed in Section 2.3.3.

Differentiating Equation 4.53 with respect to t one more time, we get that

$$\begin{aligned}\frac{d^2}{dt^2}x(t) &= -\frac{1}{\tau_H}\frac{d}{dt}x(t)+\frac{d}{dt}u(t)\\ &= -\frac{1}{\tau_H}\frac{d}{dt}x(t)+a(t)-\frac{1}{\tau_F}u(t) \text{ (by using Equation 4.54).}\end{aligned} \tag{4.66}$$

Rearranging terms we get

$$\frac{d^2}{dt^2}x(t)+\frac{1}{\tau_H}\frac{d}{dt}x(t)+\frac{1}{\tau_F}u(t)=a(t). \tag{4.67}$$

However, from Equation 4.53 we get that

$$u(t)=\frac{d}{dt}x(t)+\frac{1}{\tau_H}[x(t)-\mu]. \tag{4.68}$$

From Equation 4.68 we get that

$$\frac{1}{\tau_F}u(t)=\frac{1}{\tau_F}\frac{d}{dt}x(t)+\frac{1}{\tau_F\tau_H}[x(t)-\mu]. \tag{4.69}$$

Therefore,

$$\begin{aligned}&\frac{d^2}{dt^2}x(t)+\frac{1}{\tau_H}\frac{d}{dt}x(t)+\frac{1}{\tau_F}\frac{d}{dt}x(t)+\frac{1}{\tau_F\tau_H}x(t)-\frac{\mu}{\tau_F\tau_H}=a(t)\\ &=\frac{d^2}{dt^2}x(t)+\left(\frac{1}{\tau_H}+\frac{1}{\tau_F}\right)\frac{d}{dt}x(t)+\frac{1}{\tau_F\tau_H}x(t)-\frac{\mu}{\tau_F\tau_H}.\end{aligned} \tag{4.70}$$

We can denote the coefficients in front of the derivatives of $x(t)$: $a_2=1$, $a_1=\frac{1}{\tau_H}+\frac{1}{\tau_F}$, $a_0=\frac{1}{\tau_H\tau_F}$. We ignore the extra constant (as we can choose $\mu=0$). Following the procedure outlined in Särkkä et al. [2013] we get that the spectral factorisation of the OUF process is

$$S(a)=q_c|G(ia)|^2, \tag{4.71}$$

where q_c is the spectral factorisation of the white noise process $a(t)$ and $G(ia)=\frac{1}{a_2(ia)^2+a_1(ia)+a_0}$.

From Equation 4.71, we get that

$$\begin{aligned} S(a) &= \frac{q_c}{|-a_2a^2+(a_1i)a+a_0)|^2} = \frac{q_c}{\left(-a_2a^2+a_0\right)^2+(a_1a)^2} \\ &= \frac{q_c}{a_2^2a^4+\left(a_1^2-2a_2a_0\right)a^2+a_0^2}. \end{aligned} \tag{4.72}$$

The stationary covariance function of the process is given by the inverse Fourier transform of the spectral density

$$C(t) = \frac{1}{2\pi}\int S(a)\exp(iat)da. \tag{4.73}$$

From Equation 4.72, we denote $f(a) = a_2^2a^4+\left(a_1^2-2a_2a_0\right)a^2+a_0^2$. We denote $a^2 = w^2$, $A = {a_2}^2 = 1$, $B = {a_1}^2-2a_2a_0$, and $C = a_0^2$ to get to a standard quadratic reduced equation form. We then get the following equation

$$f(w) = A\left(w^2+\frac{B}{2A}\right)^2-\frac{B^2-4AC}{4A^2}. \tag{4.74}$$

The Equation 4.74 is of the following form: $(w^2+\text{constant}_1)^2-\text{constant}_2$. Thus, Equation 4.72 is of the following form

$$S(w) = \frac{\text{constant}_3}{(w^2+\text{constant}_4)^2+\text{constant}_5}. \tag{4.75}$$

Since Equation 4.75 does not have an inverse Fourier transformation, in order to use inverse Fourier transformations to solve Equation 4.73, we use partial fractions. The solutions to Equation 4.74 are of the form $w = \pm\alpha_1$ and $w = \pm\alpha_2$. Therefore,

$$\begin{aligned} f(w) &= A(w-\alpha_1)(w+\alpha_1)(w-\alpha_2)(w+\alpha_2) \\ &= A(w^2-\alpha_1^2)(w^2-\alpha_2^2) \\ &= (w^2-\alpha_1^2)(w^2-\alpha_2^2). \end{aligned} \tag{4.76}$$

Thus, Equation 4.75 has the form

$$\begin{aligned} S(w) &= \frac{\text{constant}_6}{w-\alpha_1}+\frac{\text{constant}_7}{w+\alpha_1}+\frac{\text{constant}_8}{w-\alpha_2}+\frac{\text{constant}_9}{w+\alpha_2} \\ &= \frac{\text{constant}_{10}}{(w^2-\alpha_1^2)(w^2-\alpha_2^2)} \\ &= \frac{\text{constant}_{11}}{w^2-\alpha_1^2}+\frac{\text{constant}_{12}}{w^2-\alpha_2^2}. \end{aligned} \tag{4.77}$$

We can replace the constants in Equation 4.74 with $A = a_2^2 = 1$, $B = a_1^2-2a_2a_0$ and $C = a_0^2$, where $a_1 = \left(\frac{1}{\tau_H}+\frac{1}{\tau_F}\right)$ and $a_0 = \frac{1}{\tau_H\tau_F}$. From Equation 4.74, we start calculating the two terms

in the RHS of the equation. Firstly, we have

$$\begin{aligned}\frac{B}{2A} &= \frac{1}{2}\left[\left(\frac{1}{\tau_H}+\frac{1}{\tau_F}\right)^2-\frac{2}{\tau_H\tau_F}\right]=\frac{1}{2}\left[\frac{(\tau_H+\tau_F)^2}{(\tau_H\tau_F)^2}-\frac{2\tau_H\tau_F}{(\tau_H\tau_F)^2}\right]\\ &= \frac{1}{2}\left[\frac{\tau_H^2+\tau_F^2}{(\tau_H\tau_F)^2}\right]=\frac{1}{2}\left(\frac{1}{\tau_H^2}+\frac{1}{\tau_F^2}\right).\end{aligned} \tag{4.78}$$

Secondly, we calculate the term

$$\begin{aligned}\frac{B^2-4AC}{4A^2} &= \frac{B^2-4C}{4}=\frac{\left(a_1^2-2a_0\right)^2-4a_0^2}{4}\\ &= \frac{1}{4}\left[\left(\frac{1}{\tau_H}+\frac{1}{\tau_F}\right)^2-\frac{2}{\tau_H\tau_F}\right]^2-\frac{1}{(\tau_H\tau_F)^2}\\ &= \frac{1}{4}\left[\frac{(\tau_H+\tau_F)^2}{(\tau_H\tau_F)^2}-\frac{2\tau_H\tau_F}{(\tau_H\tau_F)^2}\right]^2-\frac{1}{(\tau_H\tau_F)^2}\\ &= \frac{1}{4}\left[\frac{\tau_H^2+\tau_F^2}{(\tau_H\tau_F)^2}\right]^2-\frac{1}{(\tau_H\tau_F)^2}\\ &= \frac{1}{4}\left(\frac{1}{\tau_H^2}+\frac{1}{\tau_F^2}\right)^2-\frac{1}{(\tau_H\tau_F)^2}\\ &= \frac{\frac{1}{\tau_H^4}+\frac{1}{\tau_F^4}+\frac{2}{\tau_H^2\tau_F^2}-\frac{4}{(\tau_H\tau_F)^2}}{4}\\ &= \frac{\frac{1}{\tau_H^4}+\frac{1}{\tau_F^4}-\frac{2}{\tau_H^2\tau_F^2}}{4}\\ &= \left(\frac{\frac{1}{\tau_H^2}-\frac{1}{\tau_{F^2}}}{2}\right)^2.\end{aligned} \tag{4.79}$$

Therefore, solving Equation 4.74 gives

$$w^2+\frac{\left(\frac{1}{\tau_H^2}+\frac{1}{\tau_F^2}\right)}{2}=\pm\frac{\frac{1}{\tau_H^2}-\frac{1}{\tau_F^2}}{2}. \tag{4.80}$$

Solving Equation 4.80 we get that

$$\alpha_1^2=\frac{-1}{\tau_H^2}. \tag{4.81}$$

$$\alpha_2^2=\frac{-1}{\tau_F^2}. \tag{4.82}$$

Plugging the roots into Equation 4.77 and using Equation 4.72 we get

$$S(w) = \frac{q_c}{\left(w^2 + \frac{1}{\tau_H^2}\right)\left(w^2 + \frac{1}{\tau_F^2}\right)} = q_c \left(\frac{D}{w^2 + \frac{1}{\tau_H^2}} + \frac{E}{w^2 + \frac{1}{\tau_F^2}} \right), \tag{4.83}$$

where D and E are constants to be determined. Multiplying Equation 4.83 by $w^2 + \frac{1}{\tau_H^2}$ and equating $w^2 = -\frac{1}{\tau_H^2}$ gives us that $D = \frac{1}{-\frac{1}{\tau_H^2} + \frac{1}{\tau_F^2}} = \frac{1}{\frac{-\tau_F^2}{\tau_F^2\tau_H^2} + \frac{\tau_H^2}{\tau_F^2\tau_H^2}} = \frac{\tau_H^2\tau_F^2}{(\tau_H - \tau_F)(\tau_H + \tau_F)}$. Similarly, $E = \frac{\tau_H^2\tau_F^2}{(\tau_F - \tau_H)(\tau_H + \tau_F)}$. Therefore, from Equation 4.83 we get

$$S(w) = \frac{q_c\left(\tau_H^2\tau_F^2\right)}{(\tau_H - \tau_F)(\tau_H + \tau_F)} \left(\frac{1}{w^2 + \frac{1}{\tau_H^2}} - \frac{1}{w^2 + \frac{1}{\tau_F^2}} \right). \tag{4.84}$$

The inverse Fourier transform of $S(w) = \frac{2\alpha}{\alpha^2 + w^2}$ is $e^{-\alpha|t|}$, where α is a constant. The constant α can be a complex number, with $\mathrm{Re}(\alpha) > 0$. Adding in Equation 4.84 the constants needed to perform the Fourier inverse transformations we get that

$$\begin{aligned} C(t) &= \frac{1}{2}\frac{q_c\left(\tau_H^2\tau_F^2\right)}{(\tau_H - \tau_F)(\tau_H + \tau_F)} \left(\tau_H e^{\frac{-1}{\tau_H}|t|} - \tau_F e^{\frac{-1}{\tau_F}|t|} \right). \\ &= \Gamma \frac{\tau_H e^{\frac{-1}{\tau_H}|t|} - \tau_F e^{\frac{-1}{\tau_F}|t|}}{\tau_H - \tau_F}, \end{aligned} \tag{4.85}$$

where Γ is a constant. The corresponding covariance function is $k(t,t') = C(t-t')$. From the previous relation and Equation 4.85 we obtain the OUF covariance function found in Equation 4.65, where $\sigma_a = q_c$.

In conclusion, we have discovered a small error in the final formula of Fleming et al. [2014a] covariance function found in Equation 4.60 compared to Equations 4.65 and 4.85, the constant in front of the kernel being the only difference. The formulas in Equations 4.65 and 4.85 have been derived using two different approaches, and this provides evidence that the results obtained in the aforementioned equations are derived correctly.

4.3 Numerical covariance and theoretical covariance of the movement models comparison

In this section we test whether our theoretical covariance functions for all the different movement models are correct by plotting the theoretical covariance values against the numerical covariance values obtained from simulating the respective model.

4.3.1 Brownian motion numerical covariance and theoretical covariance comparison

In Figure 4.1 we plot the numerical covariance function values computed from multiple simulations and the theoretical covariance function values obtained from Equation 4.5 for the Brownian motion. The two functions are in very close agreement.

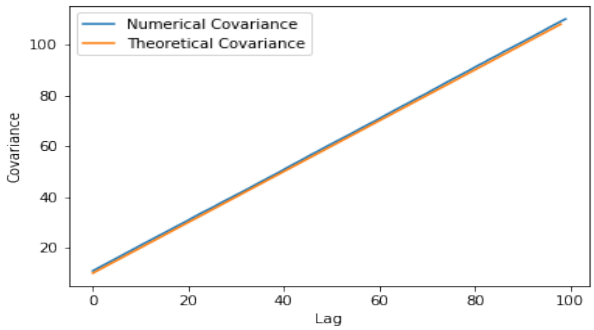


Figure 4.1: Brownian motion covariance plots, computed numerically from 100,000 simulations and theoretically from Equation 4.5. The covariance functions were computed from the pairs with indices 10 to 110 and 50 to 150 respectively.

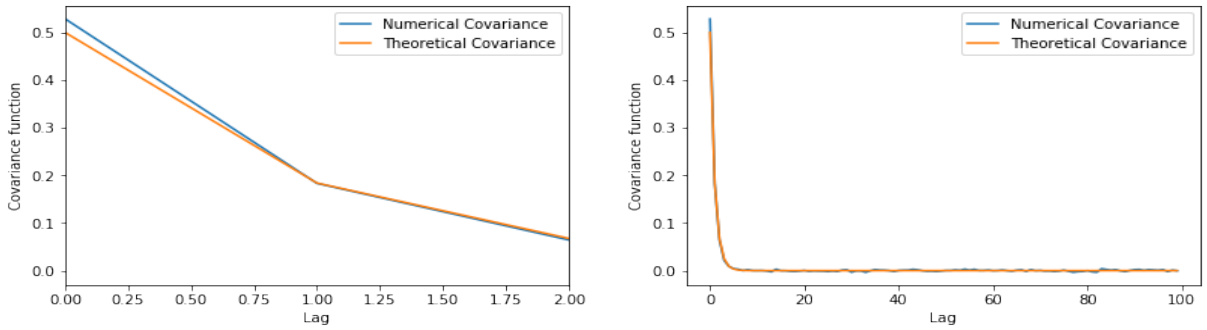
The numerical covariance function was computed by using the following formula

$$\text{Cov}(x_s, x_t) = \mathbb{E}[(x_s - \mathbb{E}(x_s))(x_t - \mathbb{E}(x_t))] = \mathbb{E}(x_s x_t), \tag{4.86}$$

given that the mean of the Brownian motion particle at position x at any time t is 0. Therefore, we simulate Brownian motion at positions x's for randomly chosen time indices. To calculate the expectation, we simulate the process multiple times, calculate the product $x_s x_t$ each time, for t and s, the indices of interest, and then take the average.

4.3.2 OU numerical covariance and theoretical covariance comparison

We simulate an OU process using the Euler-Maruyama method with a time step $\Delta_t = 0.1$, $\sigma = a = 1$. The time step was chosen such that the algorithm converges to the true solution. At a computational cost, the time step can be chosen to be smaller in order to ensure convergence of the algorithm. The numerical OU covariance function was calculated as in the previous section and the theoretical covariance function was computed from the last equation in Equation 4.22. In Figure 4.2 we plot the numerical and theoretical OU covariance function values, which agree almost perfectly.



(a) (b)

Figure 4.2: OU covariance plots, computed numerically from 100,000 simulations and theoretically from the last equation in Equation 4.22. The covariance functions were computed from the pairs with indices 10 to 110 and 10 i.e. $\text{Cov}((x[10],\ldots,x[110]), x[10])$. Figure (a) focuses more on the first few pairs, while Figure (b) shows the covariance functions plotted for all pairs mentioned above.

4.3.3 OUV model numerical covariance and theoretical covariance comparison

We simulate the OUV model by using the Euler-Maruyama method with a time step $\Delta_t = 0.01$ (smaller value than in the previous subsection, chosen such that the algorithm converges to the true solution) and $\sigma = a = 1$. The numerical covariance function was computed in the same manner as in the previous sections and the theoretical covariance function was computed from Equation 4.49. In Figure 4.3 we plot the numerical against the theoretical covariance functions for different time points and the fit is very good.

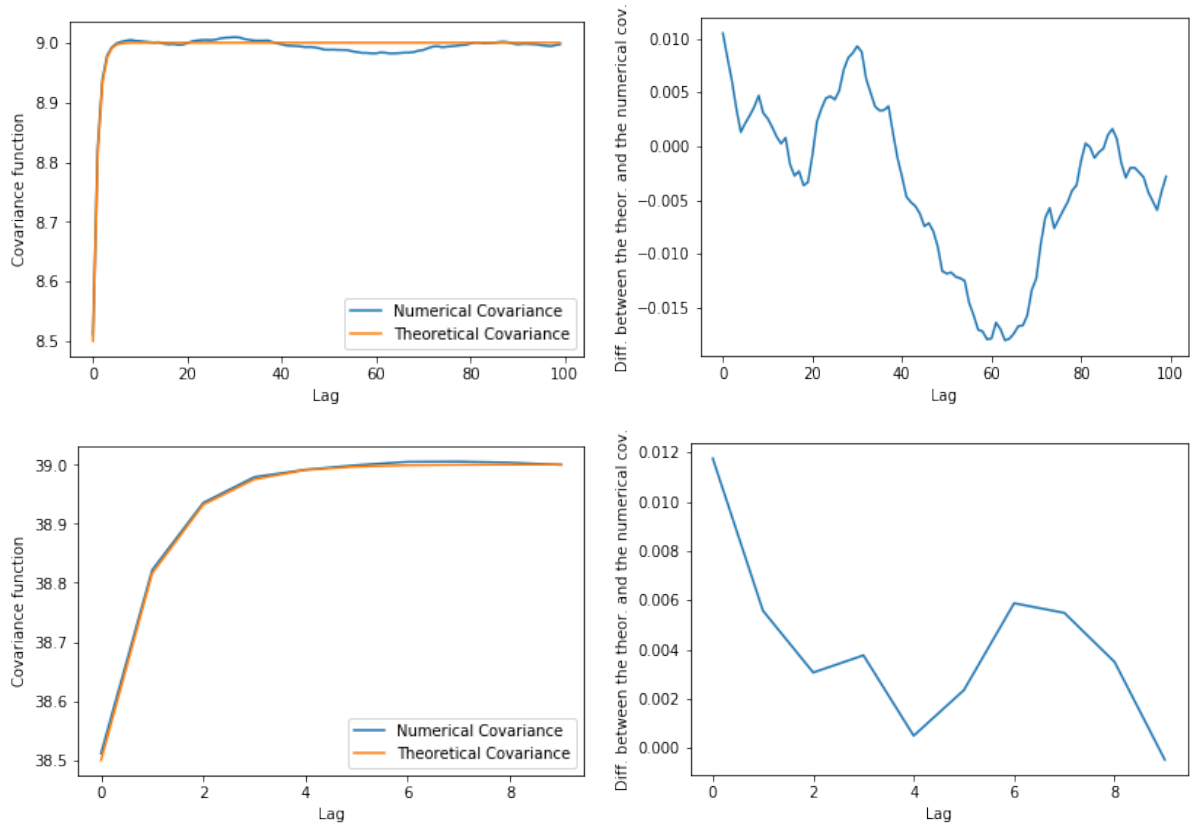


Figure 4.3: On the first column: OUV model covariance plots, computed numerically from 2,000,000 simulations (with the time step $\Delta_t = 0.01$) and theoretically from Equation 4.49. The covariance functions on the first row were computed from the pairs with indices 10 to 110 and 10 i.e. $\text{Cov}((x[10],\ldots,x[110]), x[10])$, while the covariance functions on the second row were computed from the pairs with indices 40 to 50 and 40 i.e. $\text{Cov}((x[40],\ldots,x[50]), x[40])$. On the second column: difference between the OUV model covariance plots obtained from the same pairs.

4.3.4 OU-Foraging model theoretical and numerical covariance comparison

We simulate the OUF model by using the Euler-Maruyama method with a time step $\Delta_t = 0.1$, chosen such that the algorithm converges, and by using the Equations 4.53 and 4.54. The numerical covariance function was computed in a similar manner to previous sections and the theoretical covariance function was computed in two ways using Equation 4.60 (Fleming et al. [2014a]'s OUF formula) and Equation 4.65 (the formula derived in this thesis). We plot the theoretical covariance functions against the numerical covariance function at different time points using different parameter values in Figure 4.4. There is a noticeable discrepancy between the plots, as Equation 4.60 does not have the appropriate factor in front of the equation. As the fit using the formula derived in this thesis is very good in all cases, we can now be confident that the true theoretical covariance function for the OUF model is given by Equation 4.65, not

Equation 4.60 and there is an error in latter formula.

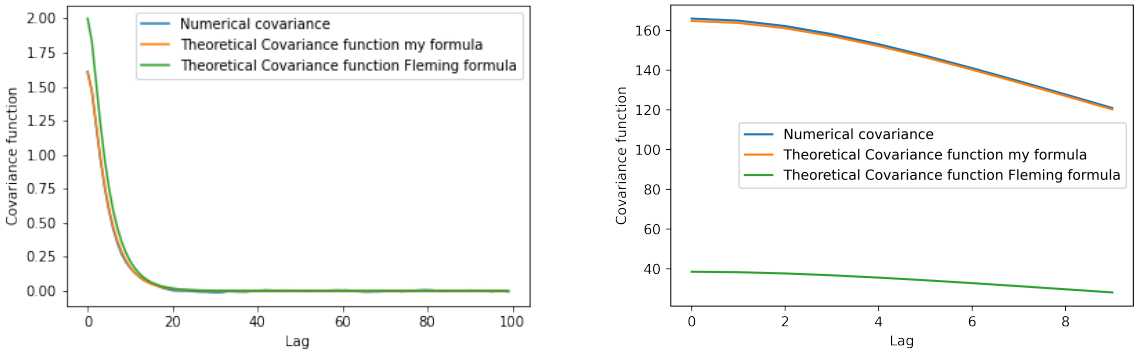


Figure 4.4: On the first column: OUF covariance functions plotted, computed numerically from 100,000 simulations (with the time step $\Delta_t = 0.1$), the theoretical covariance computed using Equation 4.65 and the theoretical covariance function using Fleming's covariance formula, Equation 4.60. The indices 10 to 110 and 10 i.e. Cov($(x[10],\ldots,x[110])$, $x[10]$) and parameter values of $\tau_H = 4$, $\tau_F = 1$ and $\sigma_a = 1$ have been used to produce the covariance functions. Similarly, on the second column: OUF covariance functions plotted, computed numerically from 200,000 simulations (with the time step $\Delta_t = 0.1$), the theoretical covariance computed using Equation 4.65 and the theoretical covariance function using Fleming's covariance formula 4.60. The indices 80 to 90 and 80 i.e. Cov($(x[80],\ldots,x[90])$, $x[80]$) and parameter values of $\tau_H = 11$, $\tau_F = 7$ and $\sigma_a = 1$ have been used to produce the covariance functions using Equation 4.65.

4.4 Conclusions

In this chapter we fully derived the theoretical covariance functions for known movement models such as OU, OUV and OUF models. We also proved that these covariance functions are correct by comparing them to the numerical covariance functions values obtained from simulating their corresponding movement model. Reformulating the continuous movement models as a GP grants us the benefits of working with a non-parametric probabilistic, flexible and powerful model that can detect multiscale patterns and trends in the data. Moreover, we can easily perform inference and quantify uncertainty for the parameters of interest using methods such as MCMC and variational inference in a quick manner using machine learning libraries such as TensorFlow in a Bayesian inference framework.

Chapter 5

Spatial latent field inference using a hierarchical GP

Understanding the spatial dynamics of animal movement is an essential component of maintaining ecological connectivity, conserving key habitats, and mitigating the impacts of anthropogenic disturbance. Altered movement and migratory patterns are often an early warning sign of the effects of environmental disturbance, and a precursor to population declines. Here, we present a hierarchical Bayesian framework based on Gaussian processes for analysing the spatial characteristics of animal movement. At the heart of our approach is a novel covariance kernel that links the spatially-varying parameters of a continuous-time velocity model with GPS locations from multiple individuals. We demonstrate the effectiveness of our framework by first applying it to a synthetic dataset, then by analysing telemetry data from the Serengeti wildebeest migration. Through the application of our approach, we are able to identify the key pathways of the wildebeest migration as well as revealing the impacts of human presence on movement behaviour.

Note: This chapter is based on the paper 'Inferring spatially-varying animal movement characteristics using a hierarchical continuous-time velocity model' submitted to the journal 'Ecology Letters' and has been accepted for publication. The paper is a collaboration formed by Ionut Paun (first author), Colin J. Torney, Dirk Husmeier and J. Grant C. Hopcraft. Colin Torney, Dirk Husmeier and Ionut Paun designed the study, Ionut Paun performed the analysis, Ionut Paun and Colin Torney wrote the manuscript with inputs from Dirk Husmeier and J. Grant C. Hopcraft, and J. Grant C. Hopcraft collected the data. I confirm that my contribution to each section of the chapter is more than 50%.

5.1 Introduction

Increasingly, animals are moving through human-altered landscapes [Tucker et al., 2018]. Infrastructure, growing human populations, and artificial boundaries, such as fences or roads, are disrupting animal movement patterns [Wittemyer et al., 2008a, Løvschal et al., 2017, Doherty et al., 2021] and consequently, many far-ranging or migratory species are in decline [Wilcove and Wikelski, 2008, Harris et al., 2009, Campbell et al., 2021, Studds et al., 2017]. In order to effectively protect these species it is essential to understand how animals respond to environmental disturbances and further, to identify the areas, such as migratory corridors, stop-over sites, or foraging grounds, that are vital for the survival of a species.

In recent years, there has been a rapid advance in our ability to collect fine scale data on the movement and behaviour of animals [Brown et al., 2013, Kays et al., 2015, Wilmers et al., 2015]. Allied with the increase in data availability has been the development of statistical models [Hooten et al., 2017] that are able to infer key characteristics of movement and identify the drivers of observed movement patterns, one of the core aims of movement ecology [Nathan et al., 2008]. A fundamental component in the statistical analysis of movement has been the random walk model [Fagan and Calabrese, 2014, Codling et al., 2008, Kareiva and Shigesada, 1983]. Using both continuous and discrete time formulations, this approach has been employed to detect different behavioural modes, such as encamped or exploratory, in movement data [Morales et al., 2004], to refine home range estimates based on the autocorrelation present in trajectories [Fleming et al., 2015], to detect spatially or temporally shifting migration routes [Gurarie et al., 2017], and to evaluate the role of social interactions in driving movement decisions [Torney et al., 2018b, Haydon et al., 2008].

Examining the landscape level drivers of movement has typically employed parametric functions of environmental covariates via HMMs [Langrock et al., 2012], or step selection functions [Thurfjell et al., 2014, Avgar et al., 2016]. Multistate random walks can be used to represent different behavioural modes with exploratory, transit states characterised by large step-lengths and high directional persistence, while encamped or foraging states display shorter step-lengths and greater tortuosity. Incorporating covariates into these models may be achieved by specifying state transition rates as functions of the environment [Morales et al., 2004, Patterson et al., 2009] or linking the random walk distributions themselves to the covariates [Hopcraft et al., 2014]. More recently flexible non-parametric approaches have been proposed that allow continuous and dynamic movement parameters to be incorporated into models [Torney et al., 2021, Michelot et al., 2021], opening the way for the development of hierarchical models of movement that are driven by latent spatial fields.

Within statistical ecology, Gaussian random fields (equivalently GPs) are a popular tool for the modelling and analysis of spatial data [Banerjee et al., 2004, Rue et al., 2009]. As opposed to semi-parametric approaches, such as splines or radial basis functions, a random field models a two-dimensional surface (representing a latent field or spatially correlated residuals)

as a realization of a stochastic process [Gelfand and Schliep, 2016]. If every finite collection of random variables that form this stochastic process has a multivariate Normal distribution, then the random field is a Gaussian random field, or a GP.

As all linear SDEs can be expressed as GPs with an appropriate covariance structure [Särkkä et al., 2013], all random walk movement models that can be formulated as a linear SDE are also equivalent to GPs [Hooten and Johnson, 2017, Torney et al., 2021]. Hence, linking spatial Gaussian random fields with a continuous-time movement model involves linking one GP with another, and is an example of multi-layered GP regression. Inference with multi-layer GPs is an active area of research in the machine learning community and several different approaches have been employed. If the output from one GP forms the input of another, then this forms a deeper GP [Damianou and Lawrence, 2013] and including multiple layers of GPs may be considered analogous to a deep neural network. Multiple GPs may also be combined within the likelihood function [Saul et al., 2016], for example if both the location and scale parameters of a distribution were allowed to vary over time or space. Finally, the outputs from multiple low level GPs can be used to define the covariance structure of a high-level GP [Heinonen et al., 2016] leading to a non-stationary stochastic process at the highest level. This final approach can be used to model data that has characteristics, such as autocorrelation or variance, that vary over time or space. It is this final approach that we adopt in this work to learn multiple latent spatial fields that define the parameters of a continuous-time velocity model of animal movement [Johnson et al., 2008]. The spatial fields are therefore the lower level GPs which provide the parameters of a covariance function of a higher level GP. These parameters have a clear ecological interpretability, representing the directional persistence and average speed of individuals at each location of the landscape.

In what follows, we introduce a non-stationary covariance matrix that allows us to link the spatial Gaussian random fields defining the parameters of the movement model with observed GPS locations. The covariance matrix we derive enables us to infer the spatially-varying parameters of a velocity model using irregularly sampled positional data with observation noise. We next describe the computational inference methodology we employ to fit the model to data and provide two example studies. In the first, we generate a synthetic dataset with known properties that we infer with our framework. In our second case study, we apply the framework to telemetry data collected over a period of 6 years from a long-term study of the Serengeti wildebeest migration.

5.2 Methods

5.2.1 A covariance matrix for non-stationary correlated velocity models

The bedrock of all movement models is the discrete-time CRW [Morales et al., 2004, McClintock et al., 2012] and was discussed and applied in Chapter 3. The standard OUV model is the closest continuous-time equivalent to the discrete time CRW, and unlike the latter model it can fit irregularly sampled data [Johnson et al., 2008]. The OUV model is also called the correlated velocity model or the integrated OU model and was discussed in Chapter 4. Given that we are dealing with non-stationary data we wish to derive a non-stationary version of the correlated velocity model, that is, we wish to derive a covariance matrix that represents the correlation structure in positional observations of an animal following an autocorrelated continuous-time random walk with varying parameters. Our starting point is therefore an assumed movement model for the animal that is a non-stationary OUV model described by the following equations,

$$\begin{aligned} d\mathbf{x} &= \mathbf{v}dt, \\ d\mathbf{v} &= -a(t)\mathbf{v}dt + b(t)d\mathbf{W}_t, \end{aligned} \tag{5.1}$$

where $\mathbf{x}$ is the true location of the animal, $\mathbf{v}$ is its velocity, $\mathbf{W}_t$ is a Wiener process, and $a(t)$ and $b(t)$ are time-varying coefficients that determine the mean-reversion rate and volatility of the OU process respectively.

While our movement model is a two-dimensional model we will present the derivation of the covariance matrix in the one-dimensional case to simplify notation and calculations. In the case of constant parameters of the movement process, i.e. $a(t) = a$ and $b(t) = b$, the covariance function of the OU process was derived in Chapter 4 and is equivalent to the exponential covariance function after relaxation of transients terms,

$$\mathrm{Cov}\,(v_t, v_s) = \frac{b^2}{2a}\exp[(-a|t-s|)]. \tag{5.2}$$

To relate the covariance of the velocity process to the covariance of the positions, we note that for a zero-mean position process

$$\mathrm{Cov}\,(x_t, x_s) = \mathbb{E}\,(x_t x_s) = \mathbb{E}\left(\int_0^t v_u du \int_0^s v_r dr\right). \tag{5.3}$$

(The zero-mean assumption can always be satisfied by a change of coordinates so that the initial location is at the origin). Through changing the order of integration and application of Fubini's theorem, Equation 5.3 leads to

$$\mathrm{Cov}\,(x_t, x_s) = \int_0^t \int_0^s \mathrm{Cov}\,(v_u, v_r)\, dudr. \tag{5.4}$$

Hence, the covariance of the position process can be found by performing the double integration of the covariance of the velocity process. For constant parameters of the velocity process the covariance function defined by Equation 5.2 may be substituted into Equation 5.4 and the integral is tractable.

In this work, we are interested in time-varying velocity characteristics and we therefore employ a non-stationary version of Equation 5.2 proposed in Paciorek and Schervish [2004] as

$$\mathrm{Cov}\left(v_t, v_s\right) = \sigma_{st}^2 \exp\left[\left(-\frac{|t-s|}{l_{st}}\right)\right], \tag{5.5}$$

where

$$\begin{aligned} \sigma_{st}^2 &= \sigma(s)\sigma(t)\sqrt{\frac{2l(s)l(t)}{l(s)^2+l(t)^2}}, \\ l_{st} &= \sqrt{\frac{l(s)^2+l(t)^2}{2}}, \end{aligned} \tag{5.6}$$

and $l(t), \sigma(t)$ are the values at time t of the time-varying kernel lengthscale and amplitude parameters respectively. Note that the parameterisation used here differs from that of Equation 5.2, but there is a direct correspondence between the two.

Substituting Equation 5.5 into Equation 5.4 gives

$$\mathrm{Cov}\left(x_t, x_s\right) = \int_0^t \int_0^s \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] dudr, \tag{5.7}$$

which contains an intractable integral due to the non-constant nature of l and σ. To approximate a numerical solution to the double integral we make the following assumption. As we have observations of the animal trajectory at discrete, known time points, we assume that between two successive observations the parameters of the movement process are constant and this will provide good approximations if the data is sufficiently dense and relatively uniform. This assumption means that if we have n observations, the non-stationary OU process will be split into $n-1$ piecewise OU processes, with each process having constant parameters. The advantage of this approach is that we can break down the integrals of Equation 5.4 into segments corresponding to the intervals between observations. Each segment has constant l and σ values, therefore the integral can be solved. We then sum over segments to obtain the full integral.

In more detail, given observations at discrete time points t_1, t_2,..., t_n, where n is the total number of observations, we have

$$\mathrm{Cov}\left(x_i, x_j\right) = \int_{t_1}^{t_i} \int_{t_1}^{t_j} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] dudr. \tag{5.8}$$

The inner integral can be written as a sum of integrals with limits corresponding to observation

times,

$$\int_{t_1}^{t_2} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] du + \int_{t_2}^{t_3} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] du \\ \cdots + \int_{t_{j-1}}^{t_j} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] du, \tag{5.9}$$

with a similar decomposition employed for the outer integral. Combined this leads to

$$\text{Cov}\left(x_i, x_j\right) = \sum_{q=1}^{i-1} \sum_{p=1}^{j-1} \int_{t_q}^{t_{q+1}} \int_{t_p}^{t_{p+1}} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] du dr. \tag{5.10}$$

As each term of the summation corresponds to a pair of between-observation intervals (p,q), the parameters of the movement process are assumed to be constant. Within interval p, corresponding to the interval between t_p and t_{p+1}, we take the mean at the endpoints as the constant parameter value so that

$$l_p = \frac{1}{2}\left[l\left(t_p\right) + l\left(t_{p+1}\right)\right], \\ \sigma_p = \frac{1}{2}\left[\sigma\left(t_p\right) + \sigma\left(t_{p+1}\right)\right]. \tag{5.11}$$

To obtain the parameters required for the non-stationary covariance kernel, we combine Equation 5.11 with Equation 5.6 to define,

$$\sigma_{pq}^2 = \sigma_p \sigma_q \sqrt{\frac{2 l_p l_q}{l_p^2 + l_q^2}}, \\ l_{pq} = \sqrt{\frac{l_p^2 + l_q^2}{2}}. \tag{5.12}$$

Finally, we end up with a covariance matrix defined as a summation over a sequence of tractable integrals,

$$\text{Cov}\left(x_i, x_j\right) = \sum_{q=1}^{i-1} \sum_{p=1}^{j-1} \int_{t_q}^{t_{q+1}} \int_{t_p}^{t_{p+1}} \sigma_{ru}^2 \exp\left[\left(-\frac{|r-u|}{l_{ru}}\right)\right] du dr, \tag{5.13}$$

where the parameters σ_{pq} and l_{pq} are constant within the limits of integration.

To solve this equation, we now consider three cases: $p = q$, $p > q$ and $p < q$. This is done due to the modulus inside the double integral, as we need to consider the relationship between u and r, and consequently the relationship between p and q. Due to symmetry, the latter two cases will be analogous. We denote the double integral term inside Equation 5.13 as I for simplicity.

Firstly, if $p = q$, then we have within the integration interval a region where $u < r$ and a region where $u > r$. To account for the modulus term $|r-u|$, we split the inner integral into two integrals with appropriate integral bounds such that the modulus disappears. We then calculate

the integrals individually. This gives,

$$
\begin{aligned}
I &= \sigma_{pp}^2 \int_{t_p}^{t_{p+1}} \int_{t_p}^{t_{p+1}} \exp\left(-\frac{|u-r|}{l_{pp}}\right) du dr \\
&= \sigma_{pp}^2 \int_{t_p}^{t_{p+1}} \left[\int_{t_p}^{r} \exp\left(-\frac{(r-u)}{l_{pp}}\right) du + \int_{r}^{t_{p+1}} \exp\left(-\frac{(u-r)}{l_{pp}}\right) du \right] dr \\
&= \sigma_{pp}^2 \int_{t_p}^{t_{p+1}} \left[l_{pp} \exp\left(-\frac{(r-u)}{l_{pp}}\right) \Big|_{t_p}^{r} - l_{pp} \exp\left(-\frac{(u-r)}{l_{pp}}\right) \Big|_{r}^{t_{p+1}} \right] dr \\
&= \sigma_{pp}^2 l_{pp} \int_{t_p}^{t_{p+1}} \left[1 - \exp\left(-\frac{(r-t_p)}{l_{pp}}\right) - \exp\left(-\frac{(t_{p+1}-r)}{l_{pp}}\right) + 1 \right] dr \\
&= \sigma_{pp}^2 l_{pp} \left[2r + l_{pp} \exp\left(-\frac{(r-t_p)}{l_{pp}}\right) - l_{pp} \exp\left(-\frac{(t_{p+1}-r)}{l_{pp}}\right) \right]_{t_p}^{t_{p+1}} \\
&= \sigma_{pp}^2 l_{pp} \left[2(t_{p+1}-t_p) + l_{pp} \exp\left(-\frac{(t_{p+1}-t_p)}{l_{pp}}\right) - l_{pp} - l_{pp} + l_{pp} \exp\left(-\frac{(t_{p+1}-t_p)}{l_{pp}}\right) \right] \\
&= 2\sigma_{pp}^2 l_{pp}^2 \left[\frac{(t_{p+1}-t_p)}{l_{pp}} + \exp\left(-\frac{(t_{p+1}-t_p)}{l_{pp}}\right) - 1 \right].
\end{aligned}
\tag{5.14}
$$

In the second case, when $p > q$, we have that $t_p \geq t_{q+1}$ and hence $u > r$ throughout the interval and the modulus term can be replaced with $u - r$. We calculate each integral in turn to give,

$$
\begin{aligned}
I &= \sigma_{pq}^2 \int_{t_q}^{t_{q+1}} \int_{t_p}^{t_{p+1}} \exp\left(-\frac{(u-r)}{l_{pq}}\right) du dr = \sigma_{pq}^2 \int_{t_q}^{t_{q+1}} \left[-l_{pq} \exp\left(-\frac{(u-r)}{l_{pq}}\right) \Big|_{t_p}^{t_{p+1}} dr \right] \\
&= \sigma_{pq}^2 l_{pq} \int_{t_q}^{t_{q+1}} \exp\left(-\frac{(t_p-r)}{l_{pq}}\right) - \exp\left(-\frac{(t_{p+1}-r)}{l_{pq}}\right) dr \\
&= \sigma_{pq}^2 l_{pq}^2 \left[\exp\left(-\frac{(t_p-r)}{l_{pq}}\right) - \exp\left(-\frac{(t_{p+1}-r)}{l_{pq}}\right) \right]_{t_q}^{t_{q+1}} \\
&= \sigma_{pq}^2 l_{pq}^2 \left[\exp\left(-\frac{(t_p-t_{q+1})}{l_{pq}}\right) - \exp\left(-\frac{(t_{p+1}-t_{q+1})}{l_{pq}}\right) - \exp\left(-\frac{(t_p-t_q)}{l_{pq}}\right) \right. \\
&\qquad \left. + \exp\left(-\frac{(t_{p+1}-t_q)}{l_{pq}}\right) \right].
\end{aligned}
\tag{5.15}
$$

Keeping in mind the symmetry in p and q we can rewrite the last equation as

$$I = \sigma_{pq}^2 l_{pq}^2 \left[\exp\left(-\frac{|t_p - t_{q+1}|}{l_{pq}}\right) - \exp\left(-\frac{|t_{p+1} - t_{q+1}|}{l_{pq}}\right) - \exp\left(-\frac{|t_p - t_q|}{l_{pq}}\right) + \exp\left(-\frac{|t_{p+1} - t_q|}{l_{pq}}\right) \right]. \tag{5.16}$$

Therefore, the covariance matrix of the non-stationary integrated OU process for the positions is

$$\mathrm{Cov}\left(x_i, x_j\right) = \sum_{q=0}^{i-1} \sum_{p=0}^{j-1} \sigma_{pq}^2 l_{pq}^2 \left[2\delta_{pq} \frac{\left(t_{p+1} - t_q\right)}{l_{pq}} + \exp\left(-\frac{|t_p - t_{q+1}|}{l_{pq}}\right) - \exp\left(-\frac{|t_{p+1} - t_{q+1}|}{l_{pq}}\right) - \exp\left(-\frac{|t_p - t_q|}{l_{pq}}\right) + \exp\left(-\frac{|t_{p+1} - t_q|}{l_{pq}}\right) \right], \tag{5.17}$$

where $\sigma_{pq}^2 = \sigma_p \sigma_q \sqrt{\frac{2 l_p l_q}{l_p^2 + l_q^2}}$, $l_{pq} = \sqrt{\frac{l_p^2 + l_q^2}{2}}$ and $\delta_{pq} = 1$ when $p = q$, and 0 otherwise, δ_{pq} is the Kronecker delta term.

5.2.2 Model formulation

In this work, we develop a two-layer hierarchical GP model using the non-stationary integrated OU kernel matrix derived above. In this formulation the lengthscale parameter (corresponding to directional persistence) and the variance parameter (corresponding to speed) are modelled by GPs, however we assume that the measurement error is homogeneous, since we use the same kind of telemetry equipment to record all the observations. The spatial location $\mathbf{x} = x(\mathbf{t})$ is a 2-dimensional matrix composed of latitude and longitude coordinates, while our observations consist of a vector $\mathbf{y} = y(\mathbf{t})$ that is an $n \times 2$ matrix of locations at times $\mathbf{t}$. We assume a regression model for the top layer of our GP hierarchy as,

$$\mathbf{y} = x(\mathbf{t}) + \boldsymbol{\varepsilon}, \tag{5.18}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \omega^2 \mathbf{I}\right)$ is a random observation noise vector term that follows a Normal distribution with variance ω^2. The latent function x corresponds to the (unknown) true location of the animal and we place a GP prior on this vector-valued function,

$$x(\mathbf{t}) \sim \mathcal{GP}\left(\mathbf{y}_0, k_{NS}\left(\mathbf{t}, \mathbf{t}'\right)\right), \tag{5.19}$$

where $\mathbf{y}_0$ is the location of the animal at the first time point and $k_{NS}(\mathbf{t}, \mathbf{t}')$ is the integrated non-stationary kernel defined by Equation 5.17. We refer to this as the first (or top) layer of our

hierarchy and this corresponds to assuming that the animal is following an unbiased correlated random walk [Johnson et al., 2008] with varying characteristics.

Implicit in the specification of the GP prior is the dependence of $k_{NS}(\mathbf{t},\mathbf{t}')$ on two lower-level GPs. This is the second layer of our hierarchy. The lengthscale and variance parameters are modelled as latent functions dependent on the dummy variables $\mathbf{h}$ and we place separate GP priors on these functions,

$$\begin{aligned}\tilde{l}(\mathbf{h}) &\sim \mathcal{GP}\left(\mu_l, k_l\left(\mathbf{h},\mathbf{h}'\right)\right),\\ \tilde{\sigma}(\mathbf{h}) &\sim \mathcal{GP}\left(\mu_\sigma, k_\sigma\left(\mathbf{h},\mathbf{h}'\right)\right).\end{aligned} \tag{5.20}$$

To link the two layers, we make use of the fact that the latent functions $\tilde{l}$ and $\tilde{\sigma}$ have a first-order dependence on the variables $\mathbf{h}$ and a second-order dependence on time $\mathbf{t}$. Thus, we first pass the latent functions through an exponential transform to ensure positivity, then we translate the dependence on $\mathbf{h}$ of the latent functions to the temporal dependence of the non-stationary kernel via a composition of functions, i.e.

$$\begin{aligned}l(\mathbf{t}) &= \exp\left\{\left[\tilde{l}\left(h(\mathbf{t})\right)\right]\right\},\\ \sigma(\mathbf{t}) &= \exp\{[\tilde{\sigma}\left(h(\mathbf{t})\right)]\}.\end{aligned} \tag{5.21}$$

These values are time-dependent, but mediated by the variables $\mathbf{h}$ at time $\mathbf{t}$. They enter the first layer GP via Equations 5.11 and 5.17 of the non-stationary kernel definition, thus linking Section 5.2.1 to Section 5.2.2.

The set of dummy variables $\mathbf{h} = h(\mathbf{t})$ introduced in Eq. 5.20 could be temperature, precipitation, or some record of the animal's time-dependent spatial preferences (e.g. elicited from the literature or collated from a wider field study). If the variables $\mathbf{h}$ are assumed independent of the data $\mathbf{y}$, then the model is methodologically accurate and probabilistically valid, and can be consistently represented by a directed acyclic graph (DAG). However, in this chapter, we have set the dummy variables $h(\mathbf{t})$ equal to $y(\mathbf{t})$, i.e. there is an undirected edge connecting $\mathbf{h}$ and $\mathbf{y}$. This violates the DAG constraint by creating a cyclic structure as the data $\mathbf{y}$ is the output and the input. Thus, we cannot apply the DAG factorisation rule of the joint probability distribution, i.e. the model is not a probabilistic generative model.

If the undirected edge between $\mathbf{h}$ and $\mathbf{y}$ is removed, then the resulting simplified model is an approximation, but conceptually is a proper probabilistic generative model that can be consistently represented by a DAG. This approach has been used to probabilistically model ODEs (the GP-ODE model) [Barber and Wang, 2014], however it can lead to identifiability problems when data are systematically missing [Macdonald et al., 2015]. Moreover, the pseudolikelihood method introduced by Besag [1975] as an approximation to the likelihood function in the context of inferring a Markov random spatial field corresponds to the application of the DAG factorisation rule to a cyclic structure since the model is based on a lattice with undirected edges

between nodes. This model has been widely used as an approximation for modelling probability distributions over lattices. Similarly, while the model presented in this chapter is an approximation, it is in line with other model approximations from the machine learning and statistics literature [Besag, 1975, Barber and Wang, 2014, Macdonald et al., 2015] and the benefits gained by introducing an approximate model capable to link locations with spatial GPs defining the parameters of the movement model are substantial. The hierarchical model's structure is illustrated in Figure 5.1.

To complete the model formulation it remains to specify the covariance kernels of the lower level GPs, k_l and k_σ. These kernels control the covariance structure of the latent spatial fields and we employ a standard RBF kernel [Rasmussen and Williams, 2006] for the empirical data study and a periodic kernel for the synthetic data. This latter choice is dictated by the periodic boundaries of the simulations (see Section 5.2.5 for details) and would not be an appropriate choice for our empirical data.

The total log probability density of a trajectory segment is calculated by adding the log marginal likelihood of the data (log marginal likelihood of the first layer of the GP hierarchy) with the log probability probability density of the latent functions (log probability density of the second layer of the GP hierarchy). The formula is given by

$$\mathscr{L} = \log\left(\mathscr{N}\left(\mathbf{y}|\mathbf{y}_0, k_{NS} + \omega^2\mathbf{I}\right)\right) + \log\left(\mathscr{N}\left(\tilde{\mathbf{l}}|\mu_l, k_l\right)\right) + \log\left(\mathscr{N}\left(\tilde{\boldsymbol{\sigma}}|\mu_\sigma, k_\sigma\right)\right), \tag{5.22}$$

where $\mathbf{I}$ is an $n \times n \times 2$ identity matrix and the two dimensions of the data $\mathbf{y}$ are independent.

The new covariance kernel represents a notable advance over previous research since it allows linking location data with a latent spatial field that defines the shared movement characteristics of multiple individuals. While previous works, including Torney et al. [2021], have applied hierarchical Gaussian processes to animal movement, in this chapter a novel contribution is obtained by linking a velocity-based movement model to a spatial random field via an integrated covariance kernel. Further, it avoids the noise amplification inherent in numerical differentiation that would be required for obtaining velocities from location-based movement models.

5.2.3 Model inference

To fit the model to data we implement our framework using TensorFlow Probability, a probabilistic programming library that is built on TensorFlow, an open-source deep learning platform [Abadi et al., 2016]. A key advantage of working with this library is that it provides access to TensorFlow's automatic differentiation capacity which allows us to efficiently compute gradients of the total log probability density of the model (marginal log likelihood of the data and the log probability density of the prior distributions over the latent functions) given by the Equation 5.22 with respect to model parameters. We use this capacity in two stages of inference. Firstly, we use a gradient-based optimiser to calculate MAP values for the latent functions of the model,

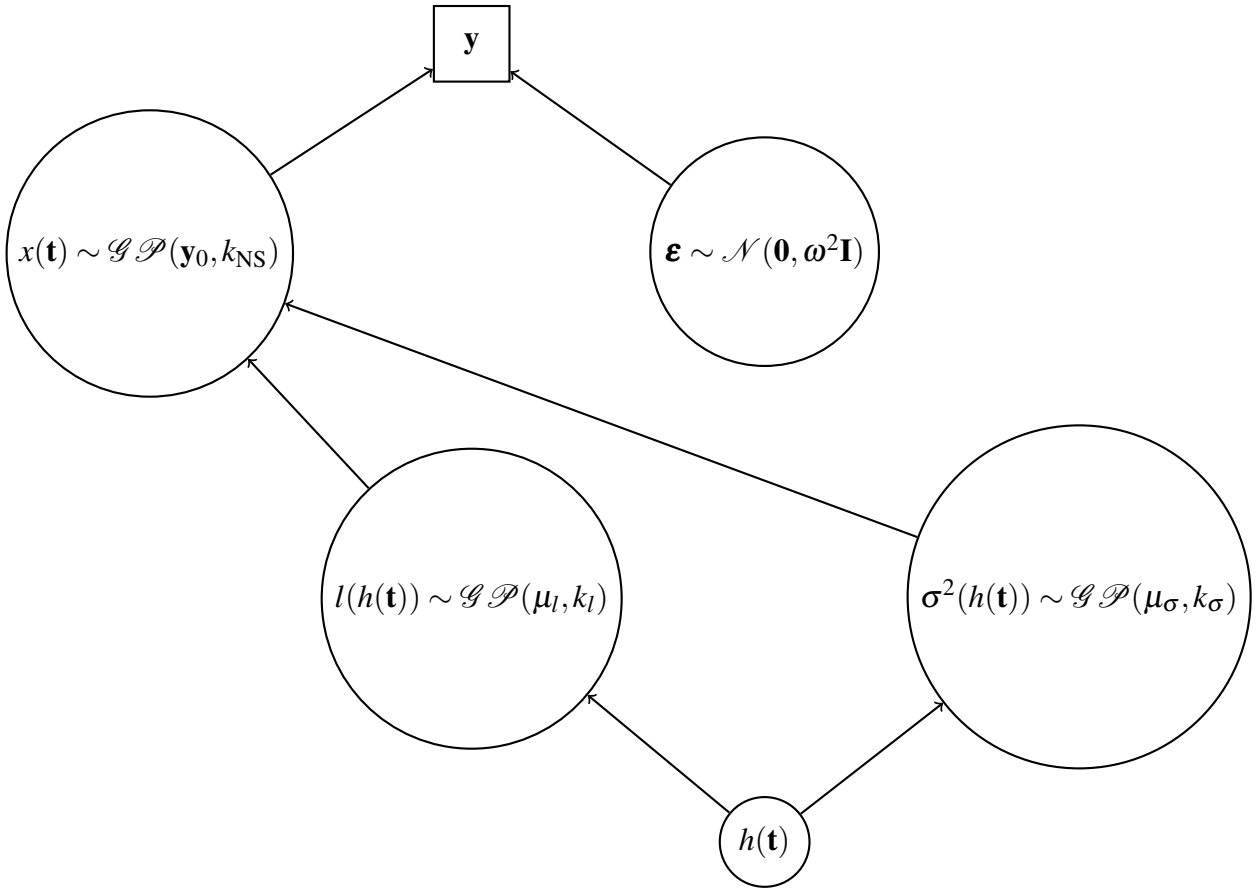


Figure 5.1: This figure shows the structure of the hierarchical Bayesian model proposed in this chapter, where we assume that the lengthscale, signal variance are also modelled by a GP. The circle nodes denote variables and the rectangle nodes denote fixed values or observations. $\mathbf{y}$ are the recorded locations at times $\mathbf{t}$, $\mathbf{h} = h(\mathbf{t})$ is a set of dummy variables that is set to $\mathbf{y}$ in this chapter.

for the low level kernel parameters and the observation noise, adopting an empirical Bayes approach to specifying these latter 'nuisance' parameters of the model. Secondly, we employ HMC [Neal, 1992] for sampling from the posterior distribution of the latent fields, while the rest of parameters are kept fixed at the MAP values. This is a gradient-based sampler that improves efficiency by biasing MCMC proposals to move in the direction of increasing likelihood.

In general, one type of problems that can be encountered while performing maximum marginal likelihood estimation for the hyperparameters of a GP are identifiability issues. While the resulting predictions are not affected, the poor estimation of the lengthscale and signal variance parameters might cause a loss of inference robustness and a lack of model interpretability [Plumlee and Joseph, 2016, Hang, 2004]. Identifiability issues can be addressed by adding more information into the model by incorporating informative priors on the hyperparameters [Brynjarsdóttir and O'Hagan, 2014]. Within the domain of statistical ecology, the appropriate choice of priors has been discussed in the literature [Wesner and Pomeranz, 2020, Lemoine, 2019, McCarthy and Masters, 2005, Banner et al., 2020, Ellison, 2004]. A good choice of a prior distribution should yield biological plausible values on the scale of the response variables.

Another limitation of GP regression is that it does not scale well to large datasets because training requires $\mathcal{O}\left(N^3\right)$ time due to the inversion of the covariance matrix. Once the inversion is complete, prediction is $\mathcal{O}\left(N\right)$ for the predictive mean and $\mathcal{O}\left(N^2\right)$ for the predictive variance per new test sample. To ensure our framework is able to run efficiently with large numbers of observations (of the order of 100,000 samples), we take certain steps in order to reduce computational complexity to manageable levels.

Firstly, we approximate the full likelihood using trajectory segmentation, where we segment individual trajectories into smaller, more computationally manageable sections. This approach extends the assumption that each GPS collar provides a trajectory that is conditionally independent of others given the latent spatial fields by further breaking trajectories from the same individual into multiple segments. For example, given a trajectory consisting of 4,000 observations spanning 2 years, we break this trajectory into 8 segments of 500 observations each spanning a 3-month period. This method, also known as a mixture of Gaussian process experts [Rasmussen and Ghahramani, 2002], has been applied successfully to movement data [Torney et al., 2021], and provides an accurate approximation to the true likelihood if the length of the trajectory segment is large compared to the autocorrelation length of the GP [Snelson and Ghahramani, 2007]. In the context of animal movement, this corresponds to selecting trajectory segments with a length greater than the maximum time scale over which directional persistence is observed.

Secondly, rather than learning a latent spatial field value for each location of a GPS fix, we define a grid of locations $\mathbf{x}_{\text{grid}}$ within a fixed domain at which we define the function values for the lower level Gaussian fields. To obtain the values of the lengthscale and variance at the location of an animal (required for Equation 5.21) we compute the conditional probabilities of the function values at that location given the grid of latent values. This approach reduces the

number of latent function values we need to infer and further provides a method for these values to be shared across trajectory segments. More specifically, we use Equations 2.29-2.31 from Chapter 2, replicated and adapted below for clarity

$$p(\mathbf{g}|\mathbf{x}, \mathbf{x}_{\text{grid}}, \mathbf{g}_{grid}) = \mathcal{N}(\mathbf{g}|\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{5.23}$$

$$\boldsymbol{\mu} = \mu(\mathbf{x}) + \mathbf{K}_{g*}^{T}\mathbf{K}_{\text{grid}}^{-1}(\mathbf{g}_{\text{grid}} - \mu(\mathbf{x}_{\text{grid}})). \tag{5.24}$$

$$\boldsymbol{\Sigma} = \mathbf{K} - \mathbf{K}_{g*}^{T}\mathbf{K}_{\text{grid}}^{-1}\mathbf{K}_{g*}, \tag{5.25}$$

where the latent function $\mathbf{g}$ is the lengthscale or the signal variance latent function, $\mathbf{g}_{\text{grid}}$ is the latent function g evaluated at the grid of locations $\mathbf{x}_{\text{grid}}$, $\mu(\mathbf{x}) = \boldsymbol{\mu}$, the mean at the actual observed locations $\mathbf{x}$, $\mathbf{K}_{\text{grid}} = \mathbf{K}(\mathbf{x}_{\text{grid}}, \mathbf{x}_{\text{grid}})$, $\mathbf{K}_{g*} = \mathbf{K}(\mathbf{x}, \mathbf{x}_{\text{grid}})$ and $\mathbf{K} = \mathbf{K}(\mathbf{x}, \mathbf{x})$. The covariance matrix $\mathbf{K}$ is calculated by using the kernels k_l or k_σ from Equations 5.20, depending whether the latent function $\mathbf{g}$ is the lengthscale, or the signal variance.

In this chapter, the focus is on inferring the movement behaviour of an individual animal while traversing a domain. Thus, we infer the lengthscale and signal variance functions as these functions can be related to the directional persistence and average speed of individual animals. Similarly, we do not predict an individual animals' future locations $\mathbf{y}^*$, but we predict the latent functions $\mathbf{l}^*$ and $\boldsymbol{\sigma}^*$ at new time points $\mathbf{t}^*$ and positions $\mathbf{x}^*$. That is, we are not interested where an individual animal will be at new time points, but how will it behave in the future at new locations.

Finally, we can model the movement of multiple animals by assigning to each individual an unique ID. Then, we group observations from the same individual into batches. Observations from different individuals will not be grouped in the same batch, but will be assigned to different batches. If observations from the same individual are not sufficient (less than than the size of a batch), then that individual is dropped.

5.2.4 Empirical data collection

GPS collars (Followit, formerly 'Televilt,' GSM or Iridium transmitters with GPS location) were deployed on 31 migratory wildebeest (*Connochaetes taurinus*) in Serengeti National park, Tanzania. Animals were immobilized by veterinarians from the Tanzania Wildlife Research Institute (TAWIRI) or the Tanzania National Parks (TANAPA) using an injectable dart containing 4-6 mg of etorphine and 80–100 mg of azaperone, fired from a veterinary rifle from a stationary vehicle near the animal. Veterinarians followed the handling and care protocols established by TAWIRI.

Collared animals were healthy reproductively active adult females (>2 years old) that were selected at random with an attempt to ensure collars were distributed throughout the main aggregations of the herds. A total of 85,000 GPS observations were obtained between June 2013 and June 2019. Collars were either collected after 2-3 years of deployment using a remote-release mechanism, or collected in the field after a mortality event. Collars were continually redeployed

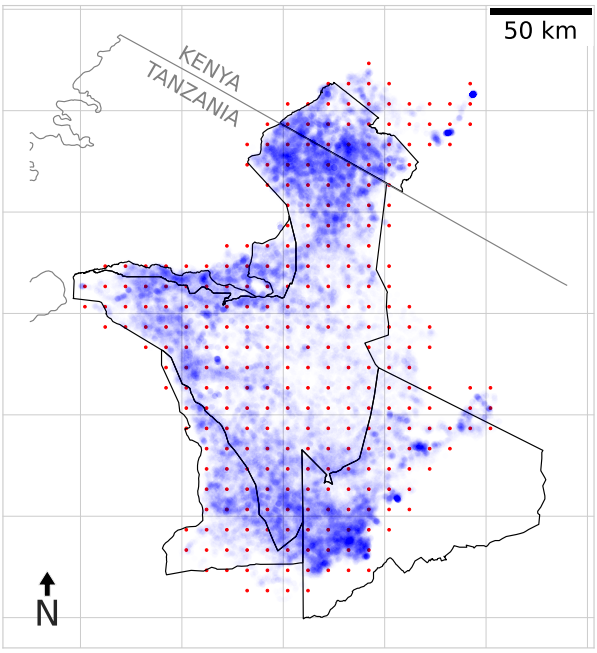


Figure 5.2: Telemetry locations and inference grid. A map of the Serengeti National Park with GPS locations shown as blue points. The red dots show the inducing grid ($\mathbf{x}_{\text{grid}}$) used for inference of the latent spatial fields.

during the study with the last deployment occurring in March 2018. Animals included in the study were tracked for periods ranging from 183 days to 1119 days. Figure 5.2 shows a map of the Serengeti National Park along with the recorded locations of wildebeest. The grid of latent function locations $\mathbf{x}_{\text{grid}}$ used for inference is also shown on the map.

5.2.5 Synthetic data generation

For the generation of the synthetic dataset, we simulate from a non-stationary correlated random walk model, where the parameters of the velocity process, mean-reversion, a, and the volatility of the OU process, b are position-dependent,

$$\begin{aligned} d\mathbf{x} &= \mathbf{v}dt, \\ d\mathbf{v} &= -a(\mathbf{x})\mathbf{v}dt + b(\mathbf{x})d\mathbf{W}_t. \end{aligned} \tag{5.26}$$

The movement process gives rise to positional observations of the animal at discrete time points that are subject to observation error, so that $\mathbf{y} = \mathbf{x} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a white noise vector term. We create the spatial fields for $a(\mathbf{x})$ and $b(\mathbf{x})$ using a two-dimensional version of the warped sine function,

$$\text{wsin}(v) = \sqrt{\frac{1+\alpha^2}{1+\alpha^2\sin^2(2\pi v)}}\sin(2\pi v), \tag{5.27}$$

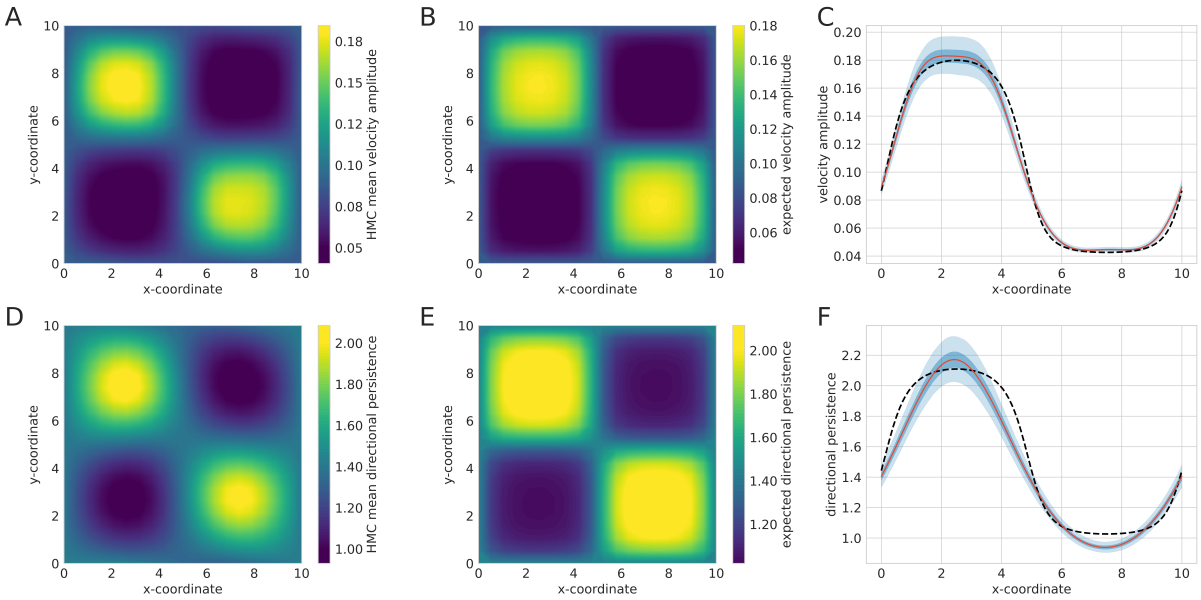


Figure 5.3: Simulation model inference. (A) and (B) show the inferred kernel variance and approximate ground truth value of the kernel variance respectively. (D) and (E) show the inferred lengthscale and the approximate ground truth kernel lengthscale respectively. (C) and (F) show a one-dimensional profile with uncertainty; the black dashed line is the approximate ground truth value of the parameters, the red line is the HMC mean and the dark blue region is the 50% CI and the light blue region is the 90% CI.

where $\alpha = 2$ gives a flattened sine wave that provides a more patch-like environment. More specifically, the spatial fields are given by

$$a(\mathbf{x}) = c_1 \log(1 + \exp(c_2 \text{wsin}(2F\pi x)\text{wsin}(2F\pi y))),$$
$$b(\mathbf{x}) = c_3 \log(1 + \exp(c_4 \text{wsin}(2F\pi x)\text{wsin}(2F\pi y))),$$

where c_i are constants, $i \in \{1,2,3,4\}$, x, y are the spatial coordinates and F is the frequency of the patches in the domain. We simulate $\mathbf{x}$ and the latent functions $a(\mathbf{x})$, $b(\mathbf{x})$ recursively, where the first location $\mathbf{x}_0$ is a randomly chosen point within the domain. The spatial field used to generate the movement trajectories are shown in Figure 5.3. The grid of latent function locations $\mathbf{x}_{\text{grid}}$ used for inference is formed of 400 equally distanced pairs of points between 0 and 10.

To account for the finite simulation domain, we introduce periodic boundary conditions for the environment. This creates an infinite domain on which the simulated animals move, but they encounter a repeating, tiled spatial field if they cross the boundaries of the environment. We simulate 200 individuals moving across an environment and collect 500 positional observations from each individual.

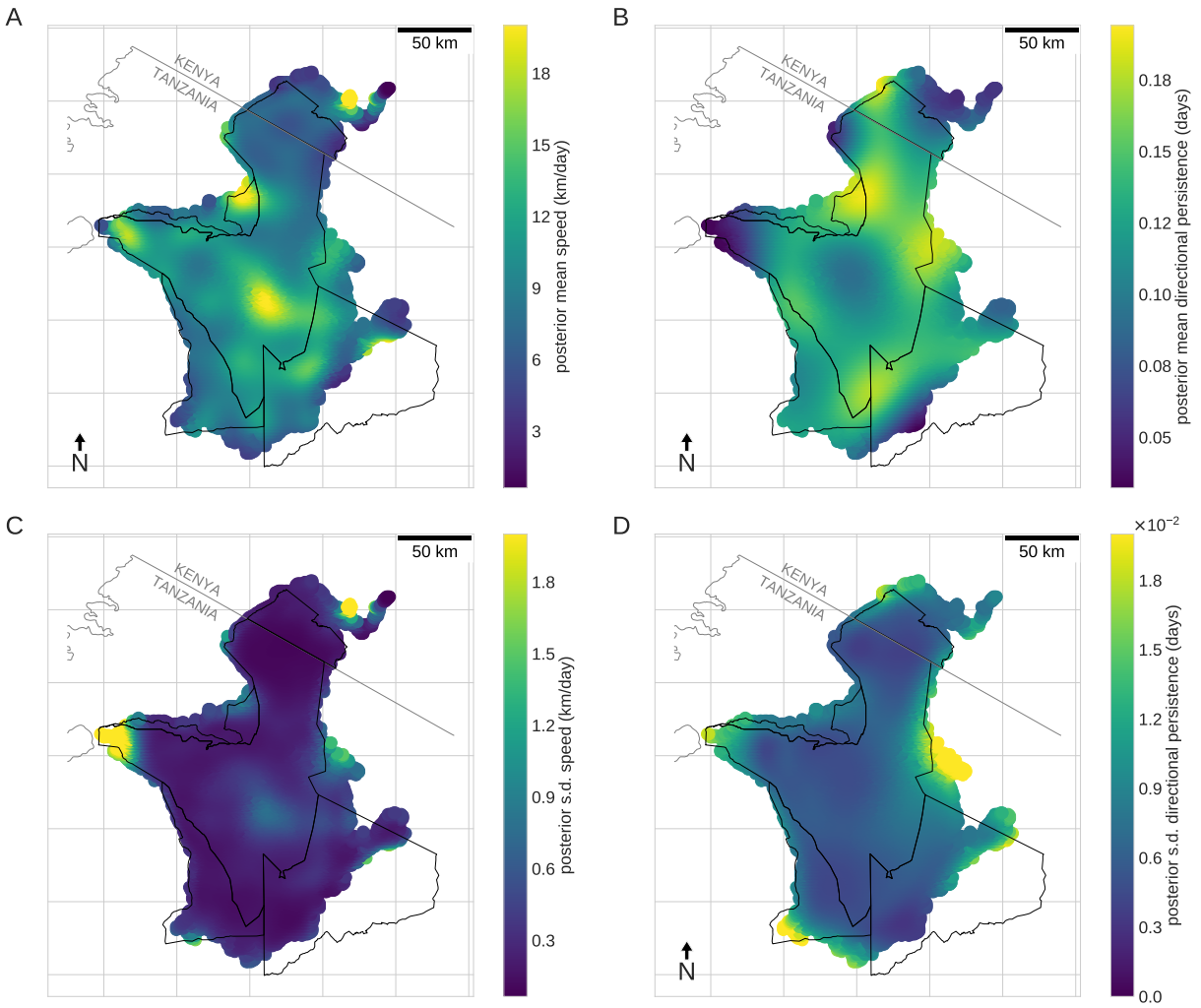


Figure 5.4: Empirical data inference. (A) Posterior mean kernel variance (speed) calculated from HMC samples. (B) Posterior mean directional persistence. (C) Kernel variance (speed) standard deviation of HMC samples. (D) Directional persistence standard deviation of HMC samples.

5.3 Results

To fit the hierarchical model to data and infer the latent spatial fields, we firstly optimise the hyperparameters of the low level GP kernels and the variance parameter of measurement error using the Adam optimiser [Kingma and Ba, 2017]. We then fix these parameters and employ HMC sampling to sample from the posterior distributions of the latent fields. To ensure convergence and mixing of MCMC chains we report effective sample sizes and potential scale reduction factors [Brooks and Gelman, 1998] (also, discussed in Section 2.6.2). The convergence diagnostics plots are shown in Figure 5.6 for the synthetic data and in Figure 5.7 for the wildeebest data. The potential scale reduction factors for all the parameters are less than 1.1, thus there is no indication of non-convergence and the effective sample size for all the parameters is sufficiently large.

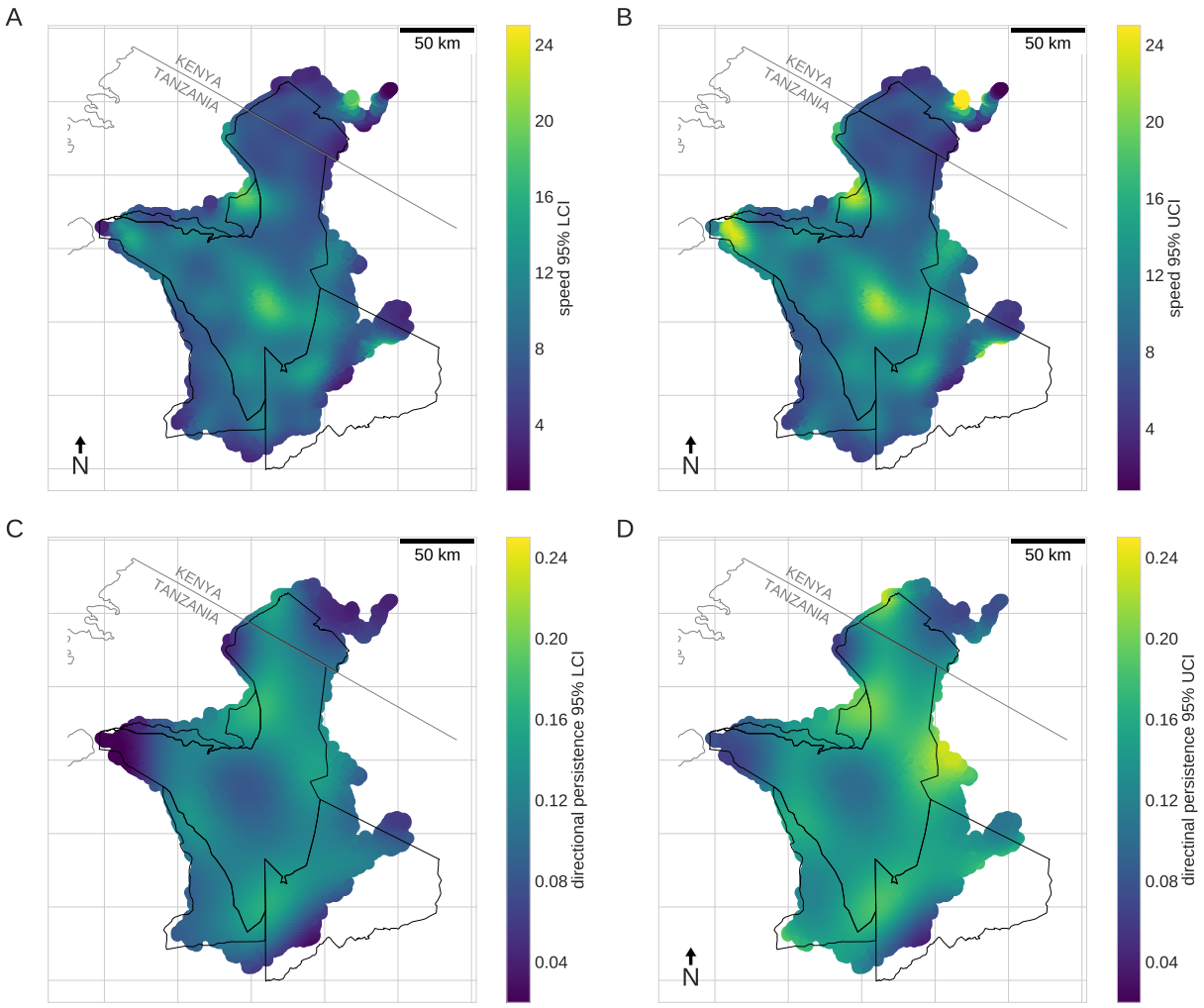


Figure 5.5: Inference of real environment. (A) and (B) show the lower and upper 95% credible intervals for the average speed. (C) and (D) show the lower and upper 95% credible intervals for the directional persistence.

5.3.1 Simulation model

Following the optimisation of hyperparameters, we ran 10 independent HMC chains, each consisting of 500 samples, after a burn-in period of 200 samples. This procedure was followed for 4 warm-up runs during which the momentum distribution of the HMC sampler was tuned. This was an essential step to ensure effective mixing of the final chain.

The mean of the posterior distributions of the latent spatial fields are shown in Figure 5.3. As we are using simulated data, this can be compared to the values used to create the movement trajectories. We observe a very close agreement between the inferred values and the simulated environment. It should be noted that there is not an exact match between the hierarchical GP model and the simulation model, however we are able to accurately locate the regions of different movement characteristics and recover the parameter values within the regions. The model is unable to perfectly capture the shape of the lengthscale function as it transitions between regions and this is due to the transient dynamics in the velocity process when an animal enters a region

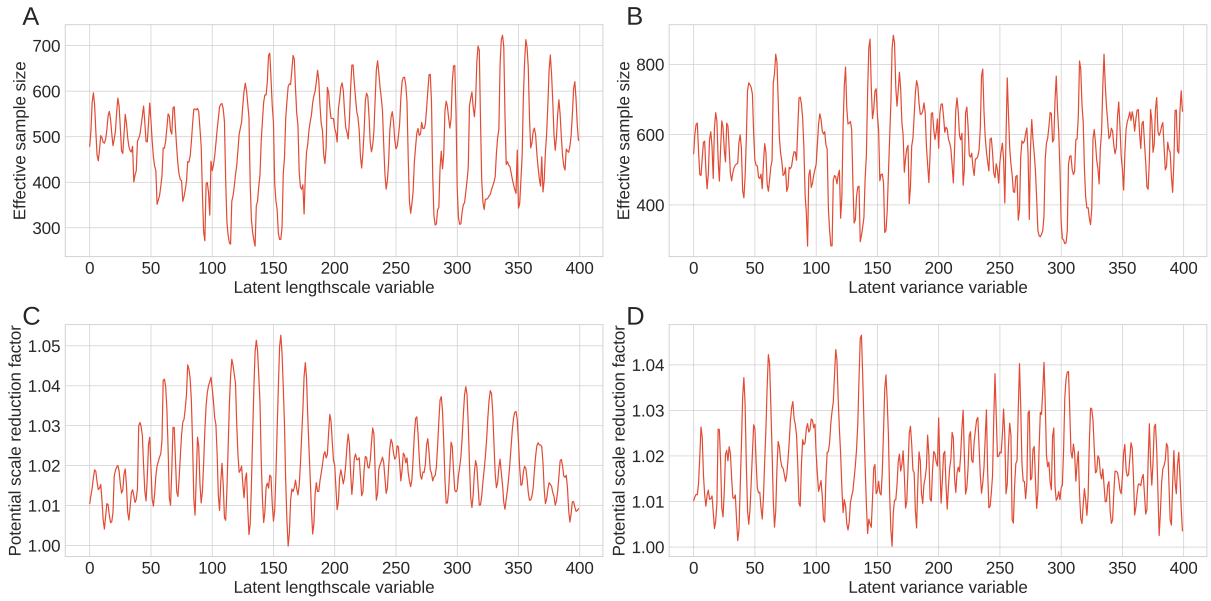


Figure 5.6: Convergence diagnostics of the synthetic environment inference: (A) and (B) show the effective sample size for the latent variables. (C) and (D) show the potential scale reduction factor for the latent variables.

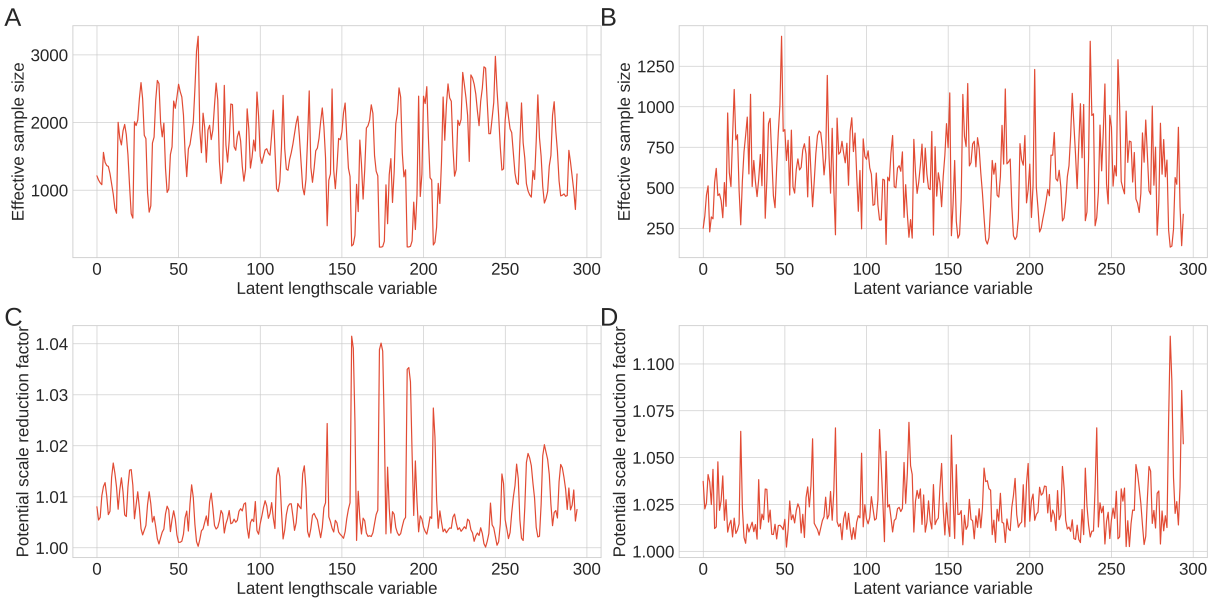


Figure 5.7: Convergence diagnostics of the empirical environment inference. (A) and (B) show the effective sample size for the latent variables. (C) and (D) show the potential scale reduction factor for the latent variables.

where its directional persistence alters. When the degree of persistence alters there is a delay before this is detectable in the data, and hence a blurring of the borders between regions.

As we employ a Bayesian framework we are able to quantify the uncertainty in the latent spatial fields. To visualize the uncertainty quantification we show the one-dimension profiles of the true environment, inferred values, and credible intervals in Figures 5.3C, 5.3F.

5.3.2 Wildebeest movement

Inferring the spatial characteristics of the wildebeest migration followed a similar approach to the synthetic data. Full movement trajectories of the wildebeest were split into trajectory segments consisting of 500 points which equated to roughly 3-months depending on the sampling schedule of the collar. We ran 10 independent HMC chains after first optimizing kernel hyperparameters as before. Each chain consisted of 2000 steps following a burn-in of 200 steps. We again ran multiple warm-up chains in order to improve mixing by specifying the proposal distribution of the sampler to match the target posterior.

Inferred posterior means for the latent fields are shown in Figure 5.4. For uncertainty quantification we also show the posterior standard deviation of the field, along with the 95% credible intervals for the posterior samples in Figure 5.5. Our results reveal the migratory pathways of the wildebeest; regions of high directional persistence can be found in a circuit around the southern extent of the Serengeti, corresponding to the main pathway that moves south along the east of the park and then north through the western corridor. A region of high speeds, but low directional persistence can be found at the centre of the migration where the long grass plains of the Serengeti are found. We expect that this pattern can be attributed to rapid forays by animals either moving towards or retreating from the ephemeral but nutrient-rich short grass plains in the south-east, as observed by Hopcraft et al. [2014].

We further detect significantly different movement behaviour in the north west of the park close to the boundary and south of the Tanzania-Kenya border. Here, we observe high speeds and high directional persistence, meaning we can identify a region through which wildebeest move directly and rapidly. This is a region of high human density and, while we can not attribute causality, our results are strongly suggestive of an effect of human presence on the movement behaviour of wildebeest [Rija and Kideghesho, 2020]. Finally, we note that uncertainty in the spatial fields is in general low. High uncertainty is only found at the edges of the wildebeest's migratory range in regions of very little data. This highlights a key advantage of our Bayesian approach. We observe high speeds at three main locations, the centre of the park, in the north-west close to villages and human activity, and at the northern border of the Masai Mara region in Kenya. Two of these locations have low uncertainty and we can be confident that we are detecting regions of significantly different movement behaviour, however there is high uncertainty associated with the high speed region in the north where data is very sparse so we are unable to draw any firm conclusions.

5.4 Discussion

In this chapter, we present a Bayesian hiearchical framework for learning the latent spatial fields that underlie observed animal movement patterns. Our framework links two fundamental concepts in statistical ecology; spatial random fields and correlated random walk models of animal

movement. As both these methods can be formulated as GPs, we adopt a multi-layer GP approach implemented within the high-performance machine learning package, TensorFlow.

Our framework has several advantages over existing approaches to animal movement modelling. Notably, multi-layer GPs offer a flexible, non-parametric method of inferring latent spatial fields. We are not required to make any restrictive assumptions about the functional form of the underlying field, however we can encode prior knowledge into the kernels of the low level GPs by employing appropriate covariance kernels.

Other popular approaches in the movement ecology literature make significant assumptions about the scale over which animal movement decisions are made, or have to select specific environmental covariates on which to regress movement parameters. As animal movement is inherently a multiscale process [Torney et al., 2018a] in which animals respond to multiple, often contradictory cues [Hopcraft et al., 2014], a latent spatial field approach can offer key insights into the different behaviours that animals exhibit across a landscape. This can be achieved without having to make decisions about which environmental features to include in a model, or how to discretise movement data into the individual choices of an animal. While our model accepts spatial location as inputs, it could in principle be adapted to accept specific covariates if required. For example, if distance to a protected area boundary was a priori the key factor of interest, it would be straightforward to substitute this metric in place of the two-dimensional spatial coordinate.

As our framework accepts irregularly sampled data with measurement error, is scalable to relatively large datasets, and formally quantifies uncertainty in inferred values, it may be applicable to many movement ecology studies. Of particular interest would be to investigate the effects of natural versus man-made barriers to movement, or the spatial charateristics of the movement behaviours of predatory species. While we have shown that we are able to analyse datasets consisting of 100,000 observations using Markov chain Monte Carlo sampling, when considering very high-frequency, long-term telemetry studies generating millions of observations it is unlikely that MCMC approaches will be practical. However, variational inference [Blei et al., 2017] offers a potential solution to this issue and has been applied to GP inference for very large datasets [Hensman et al., 2013]. Applying variational inference to multi-layer GP models of animal movement is a promising avenue of future research. Moreover, switching from a hierarchical model presented in this chapter to a deep Gaussian process [Damianou and Lawrence, 2013, Dunlop et al., 2017], which combines a deep neural network with a Gaussian process, and where the independent GP priors are set on each stochastic function can be a viable and exciting prospect for inferring complicated patterns in large movement datasets.

Chapter 6

Variational inference for a non-stationary GP

A natural extension to standard Gaussian processes (GP) is the non-stationary Gaussian process, an approach where the parameters of the covariance kernel are allowed to vary in time or space. The non-stationary GP is a realistic and flexible model that relaxes the strong prior assumption of standard GP regression, that parameters are constant across the input space. Non-stationary GPs typically model varying kernel parameters as further lower-level GPs, thereby enabling sampling-based inference. However, due to the high computational costs of sampling associated with the non-stationary GPs, these methods do not scale to large datasets. Here we develop a variational inference approach to fitting non-stationary GPs that combines sparse GP regression with a trajectory segmentation technique. Our method is scalable to large datasets containing potential millions of data points. We demonstrate the effectiveness of our approach on both synthetic and real world datasets.

Note: The chapter is a collaboration formed by Ionut Paun (first author), Colin J. Torney, Dirk Husmeier. Colin Torney, Dirk Husmeier and Ionut Paun designed the study, Ionut Paun performed the analysis, Ionut Paun wrote the manuscript with inputs from Colin Torney and Dirk Husmeier. I confirm that my contribution to each section of the chapter is more than 50%.

6.1 Introduction

Gaussian process models represent a non-parametric supervised learning approach frequently used in the machine learning community for both regression and classification purposes. Learning from data using Gaussian processes involves specifying a covariance structure for the process via a covariance kernel, inferring the parameters of the kernel, then calculating or sampling from the posterior distribution of the process conditional on observed data. Typically, a stationary GP is used so that the covariance kernel parameters depend only on the difference between data

points i.e. they are invariant to translations in the input space [Rasmussen and Williams, 2006].

A disadvantage of stationary GPs is that they lack the flexibility to fit non-stationary data, where the characteristics of the function differs across the input domain [Paciorek and Schervish, 2004, Gibbs, 1997, MacKay, 1997]. For these types of data, a non-stationary GP, where all or a subset of the kernel parameters are allowed to vary may be more appropriate. For example, in the context of tracking animal movement the error in position estimates may depend on the animal's location within a receiver array [Guzzo et al., 2018], meaning the observation noise parameter (or nugget term) of the covariance kernel will be spatially varying. Observed phenomena may also have varying smoothness or amplitude. This could be due to temporal drivers of the process which lead to periods of rapid change and high volatility [Blum and Riedmiller, 2013], the nonlinear dynamics underlying the observed variables [Heinonen et al., 2015], or varying spatial characteristics such as differences in elevation and in the nature of the environment [Lang et al., 2007].

To model data that presents characteristics with varying degrees of smoothness, Heinonen et al. [2016] proposed a non-stationary GP, where all or a subset of the kernel covariance parameters are input-dependent and modelled by other GPs. The hierarchical structure presented by this model is intrinsically linked with deep Gaussian Processes (DGP) [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017]. DGP models are a multi-layer generalisation of a GP, where the prior is defined recursively on multiple stochastic functions [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017, Wang et al., 2016] and the outputs from one layer become the inputs of the next layer. This is in contrast to the hierarchical model introduced by Heinonen et al. [2016] as in this case the outputs of the lower layers specify the kernel parameters of the final layer GP.

GPs have a high computational complexity, scaling cubically with the number of training points, which makes them impractical to implement when the datasets are large. To overcome this limitation, sparse GPs that make use of a set of m inducing points have been developed in the literature [Lázaro-Gredilla and Titsias, 2011, Titsias, 2009, Monterrubio-Gòmez et al., 2019, Hensman et al., 2013, Snelson and Ghahramani, 2006]. Thus, the computational complexity will be reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 m)$.

Using a small set of inducing points, popular variational inference methods [Hensman et al., 2013, Titsias, 2009, Lázaro-Gredilla and Titsias, 2011] construct an approximate posterior distribution to the true posterior. The distance between the true posterior and the approximate posterior is then minimised by maximising a lower bound on the marginal log likelihood. This is equivalent to minimising the Kullback-Leibler ($\mathcal{KL}$) divergence between the true posterior and the variational distribution.

Titsias [2009] defines the inducing variables as variational parameters that get inferred together with the hyperparameters either by applying continuous optimisation or by using a variational EM algorithm. However, Hensman et al. [2013] retain an explicit representation of

the inducing variables that get treated as global variables, thus defining a model suitable for stochastic variational inference (SVI). Another difference between these approaches lies in the approximation to the lower bound on the log likelihood, with the approach by Titsias [2009] resulting in a tighter lower bound to the true posterior distribution than Hensman et al. [2013]'s SVI method. However, the latter lower bound factorises, which allows the implementation of methods such as mixtures of experts [Rasmussen and Ghahramani, 2002] making the method scalable to large datasets. The variational parameters and the kernel hyperparameters are then inferred via stochastic optimisation with standard gradient descent methods [Hensman et al., 2013].

In this chapter, we adapt the inference method of Hensman et al. [2013] to the hierarchical GP model of Heinonen et al. [2016]. To make our method scalable to very large datasets, we use the mixture of GP experts technique [Rasmussen and Ghahramani, 2002] to approximate the full likelihood by using trajectory segmentation into smaller and more computationally manageable sections. We demonstrate that this novel combination leads to a substantial boost in computational efficiency at sustained high accuracy and enables large-scale applications that neither of these methods could have tackled on its own.

6.2 Methods

6.2.1 Model formulation

In this chapter, we use a double-layer non-stationary GP, where the lengthscale and the signal variance parameters are also modelled by GPs. We assume that the observation noise variance parameter is kept at a constant value in the examples we consider but it is straightforward to extend the method to include heteroscedasticity in the final layer GP. More specifically, we assume a regression model, $y_i = f(t_i) + \varepsilon_i$, where y_i is the observation at a random time point t_i and $\varepsilon_i \sim \mathcal{N}\left(0, \omega^2\right)$. We then place a zero mean GP prior on the latent function $f(t)$,

$$f(t) \sim \mathcal{GP}\left(0, \mathbf{K}_f\left(t_i, t_j\right)\right), \tag{6.1}$$

where $\mathbf{K}_f\left(t_i, t_j\right) = k_f(t_i, t_j)$ is a covariance matrix, and $k_f(t_i, t_j)$ is the Matérn 1/2 non-stationary kernel [Paciorek and Schervish, 2004] evaluated at random times t_i and t_j, given by the following relationship

$$k_f(t_i, t_j) = \sigma_i \sigma_j \sqrt{\frac{2 l_i l_j}{l_i^2 + l_j^2}} \exp\left(-\sqrt{\frac{2 d_{ij}}{l_i^2 + l_j^2}}\right), \tag{6.2}$$

where $d_{ij} = (t_i - t_j)^2$, σ_i, l_i are the signal variance, respectively lengthscale parameters at the time point t_i. This formula is derived in the Appendix, Section D.4. This is referred as the first layer of the hierarchical GP. Through the kernel k_f, which is dependent on the latent parameters

lengthscale and signal variance, the chain between the two layers of the GP is constructed, as we set separate GP priors on the aforementioned latent functions. This is referred as the second layer of the hierarchical GP such that we have

$$\tilde{P}(t) \sim \mathscr{GP}\left(\mu_P, \mathbf{K}_P\left(t_i, t_j\right)\right), \tag{6.3}$$

where $P \in \{l, \sigma^2\}$ and $\mathbf{K}_P(t_i, t_j) = k_P(t_i, t_j)$ is a covariance matrix. In order to ensure positivity of these functions, we use a softplus transformation[1] such that we have $P(t) \equiv \log\left[1 + \exp\left(\mu_P + \tilde{P}(t)\right)\right]$. For the synthetic data inference, shown in Section 6.4.1, the chosen kernel for the latent parameters is RBF for each parameter,

$$k_P(t_i, t_j) = \alpha_P^2 \exp\left(-\frac{(t_i - t_j)^2}{2\beta_P^2}\right), \tag{6.4}$$

where $P \in \{l, \sigma^2\}$, α^2 is the signal variance and β is the lengthscale. For the empirical data inference, shown in Section 6.4.2, the kernel for each parameter is a periodical kernel, namely Exponential Sine Squared

$$k_P(t_i, t_j) = \alpha_P^2 \exp\left(-\frac{2}{\beta_P^2} \sin^2\left(\pi \frac{|t_i - t_j|}{p}\right)\right), \tag{6.5}$$

where $P \in \{l, \sigma^2\}$, α_P^2 is the signal variance, β_P is the lengthscale and p is the period. The total log probability density of the hierarchical GP model is calculated by summing the log likelihood of the data on the first layer and the log probability density of the GP priors on the second layer,

$$\mathscr{L} = \log\left(\mathscr{N}\left(\mathbf{y}|0, \mathbf{K}_f + \omega^2\mathbf{I}\right)\right) + \log\left(\mathscr{N}\left(\tilde{\mathbf{l}}|\mu_l, \mathbf{K}_l\right)\right) + \log\left(\mathscr{N}\left(\tilde{\boldsymbol{\sigma}}|\mu_\sigma, \mathbf{K}_{\sigma^2}\right)\right). \tag{6.6}$$

The model developed by Heinonen et al. [2016] and used in this chapter has the likelihood function depend on one latent function $\mathbf{f}$, as shown in Equation 6.1, which in turn depends on two latent functions $\mathbf{l}$ and $\boldsymbol{\sigma}$ modelled by GPs, as shown in Equation 6.3. However, the model described by Saul et al. [2016] has the likelihood function depend directly on two independent latent functions $\mathbf{f}$ and $\mathbf{g}$ modelled by GPs. Moreover, in a hierarchical non-stationary GP model [Heinonen et al., 2016], the likelihood does not factorise over the data as it does in a standard GP model or in the model used by Saul et al. [2016] (illustrated in Chapter 2, Equations 2.162 and 2.166), as the observations y_i, given the latent parameters are dependent. Our goal in this chapter is to combine the hierarchical non-stationary GP model developed by Heinonen et al. [2016] with the variational inference framework for multiple latent functions by Saul et al. [2016].

[1] A log transformation can potentially be used as well.

6.2.2 Variational inference for non-stationary hierarchical Gaussian processes

The variational inference method for a standard GP and for the chained GP model employed by Saul et al. [2016] was illustrated in Chapter 2, Section 2.6.3. Now, we extend the variational inference framework from a simple GP to a double-layer hierarchical GP model. Therefore, at the inducing points $\mathbf{z}$, the corresponding latent functions are $\mathbf{l}_z$ and $\boldsymbol{\sigma}_z$. Likewise, at the training points $\mathbf{x}$, the corresponding latent function values are $\mathbf{l}$ and $\boldsymbol{\sigma}$. We do not define a set of inducing points for the function $\mathbf{f}$, and the latent function $\mathbf{f}$ is not inferred, as the primary focus is to infer the latent functions $\mathbf{l}$ and $\boldsymbol{\sigma}$. However, it is straightforward to extend the method to infer the function $\mathbf{f}$. More details are presented in Section 6.2.3.

We first state our model assumptions. We assume that the latent functions $\mathbf{l}$ and $\boldsymbol{\sigma}$ are a priori independent and that the prior distributions $p(\mathbf{l}_z)$ and $p(\boldsymbol{\sigma}_z)$ are multivariate Normal distributions i.e.

$$p(\mathbf{l}_z) \sim \mathcal{N}(\mathbf{m}_l, \mathbf{K}_{l_{mm}}). \tag{6.7}$$

$$p(\boldsymbol{\sigma}_z) \sim \mathcal{N}(\mathbf{m}_\sigma, \mathbf{K}_{\sigma_{mm}}), \tag{6.8}$$

where m is the number of inducing points $\mathbf{z}$. To derive the variational lower bound to the true posterior distribution we further assume that the the following relationship holds

$$p(\mathbf{l}, \boldsymbol{\sigma}|\mathbf{l}_z, \boldsymbol{\sigma}) = p(\mathbf{l}|\mathbf{l}_z)p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z). \tag{6.9}$$

Using the previous assumptions, the following relationship regarding the posterior distribution of the latent functions at the training points holds

$$p(\mathbf{l}, \boldsymbol{\sigma}, \mathbf{l}_z, \boldsymbol{\sigma}_z|\mathbf{y}) = p(\mathbf{l}|\mathbf{l}_z)p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z)p(\mathbf{l}_z, \boldsymbol{\sigma}_z|\mathbf{y}), \tag{6.10}$$

where $p(\mathbf{l}_z, \boldsymbol{\sigma}_z|\mathbf{y})$ is the joint posterior distributions at the inducing points and $p(\mathbf{l}|\mathbf{l}_z)$, $p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z)$ can be found in closed form by using conditional probabilities of Gaussian distributions.

To perform variational inference, we introduce a variational approximation distribution ϕ to the posterior distribution,

$$p(\mathbf{l}, \boldsymbol{\sigma}, \mathbf{l}_z, \boldsymbol{\sigma}_z|\mathbf{y}) \approx p(\mathbf{l}|\mathbf{l}_z)p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z)\phi(\mathbf{l}_z, \boldsymbol{\sigma}_z), \tag{6.11}$$

where we take the variational distributions to be of the following form

$$\phi(\mathbf{l}_z, \boldsymbol{\sigma}_\mathbf{z}) \sim \mathcal{N}\left(\boldsymbol{\mu}_q, \mathbf{K}_q\right). \tag{6.12}$$

Our aim is to infer the parameters of the joint variational distribution, $\boldsymbol{\mu}_q$ and $\mathbf{K}_q$.

The marginal log-likelihood has the following formula

$$\log p(\mathbf{y}) = \log \iiiint p(\mathbf{y}|\mathbf{l},\boldsymbol{\sigma})p(\mathbf{l},\boldsymbol{\sigma}|\mathbf{l}_z,\boldsymbol{\sigma}_z)p(\mathbf{l}_z)p(\boldsymbol{\sigma}_z)d\mathbf{l}d\boldsymbol{\sigma}d\mathbf{l}_z d\boldsymbol{\sigma}_z, \tag{6.13}$$

and using Equation 6.9 we get that

$$\log p(\mathbf{y}) = \log \iiiint p(\mathbf{y}|\mathbf{l},\boldsymbol{\sigma})p(\mathbf{l}|\mathbf{l}_z)p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z)p(\mathbf{l}_z)p(\boldsymbol{\sigma}_z)d\mathbf{l}d\boldsymbol{\sigma}d\mathbf{l}_z d\boldsymbol{\sigma}_z. \tag{6.14}$$

Furthermore, to perform variational inference we obtain a lower bound on the marginal log likelihood using Jensen's inequality

$$\log p(\mathbf{y}) \geq \iint \log p(\mathbf{y}|\mathbf{l},\boldsymbol{\sigma})q(\mathbf{l},\boldsymbol{\sigma})d\mathbf{l}d\boldsymbol{\sigma} - \mathscr{K}\mathscr{L}(\phi(\mathbf{l}_z,\boldsymbol{\sigma}_z)||p(\mathbf{l}_z)p(\boldsymbol{\sigma}_z)), \tag{6.15}$$

where $q(\mathbf{l},\boldsymbol{\sigma}) = \iint p(\boldsymbol{\sigma}|\boldsymbol{\sigma}_z)p(\mathbf{l}|\mathbf{l}_z)\phi(\mathbf{l}_z,\boldsymbol{\sigma}_z)d\mathbf{l}_z d\boldsymbol{\sigma}_z$.

To make our method scalable to large datasets, we use a trajectory segmentation technique such that the data is split into L independent segments of equal length. However, inside a segment, the observations $\mathbf{y}_i$, given $\mathbf{l}_i$, $\boldsymbol{\sigma}_i$ are dependent, since we work within a non-stationary GP framework. The gradients and the log likelihood of each segment can be added up to get a good approximation to the full gradients and the log likelihood provided that the domain of the local GP is sufficiently large compared to the lengthscale of the GP being fitted. Thus, given that the likelihood factorises over segments we can use stochastic variational inference [Hensman et al., 2013].

The likelihood factorises across L segments such that we get

$$p(\mathbf{y}|\mathbf{l},\boldsymbol{\sigma}) = \prod_{i=1}^{L} p(\mathbf{y}_i|\mathbf{l}_i,\boldsymbol{\sigma}_i), \tag{6.16}$$

where $\mathbf{y}_i$ denotes the observation segment i, and $\mathbf{l}_i$, $\boldsymbol{\sigma}_i$ the segment i of parameters. Then, we have that the integral in Equation 6.15 can also be factorised such that we have

$$\begin{aligned}\iint \log\ p(\mathbf{y}|\mathbf{l},\boldsymbol{\sigma})q(\mathbf{l})q(\boldsymbol{\sigma})d\mathbf{l}d\boldsymbol{\sigma} &= \iint \log \prod_{i=1}^{L} p(\mathbf{y}_i|\mathbf{l}_i,\boldsymbol{\sigma}_i)q(\mathbf{l},\boldsymbol{\sigma})d\mathbf{l}d\boldsymbol{\sigma} \\ &= \sum_{i=1}^{L} \iint \log p(\mathbf{y}_i|\mathbf{l}_i,\boldsymbol{\sigma}_i)q(\mathbf{l}_i,\boldsymbol{\sigma}_i)d\mathbf{l}_i d\boldsymbol{\sigma}_i.\end{aligned} \tag{6.17}$$

Hence, Equation 6.15 becomes

$$\begin{aligned}\log p(\mathbf{y}) &\geq \sum_{i=1}^{L} \iint \log p(\mathbf{y}_i|\mathbf{l}_i,\boldsymbol{\sigma}_i)q(\mathbf{l}_i,\boldsymbol{\sigma}_i)d\mathbf{l}_i d\boldsymbol{\sigma}_i - \mathscr{K}\mathscr{L}(\phi(\mathbf{l}_z,\boldsymbol{\sigma}_z)||p(\mathbf{l}_z)p(\boldsymbol{\sigma}_z)) \\ &= \mathbb{E}_{q(\mathbf{l}_i,\boldsymbol{\sigma}_i)} \sum_{i=1}^{L} p(\mathbf{y}_i|\mathbf{l}_i,\boldsymbol{\sigma}_i) - \mathscr{K}\mathscr{L}(\phi(\mathbf{l}_z,\boldsymbol{\sigma}_z)||p(\mathbf{l}_z)p(\boldsymbol{\sigma}_z)).\end{aligned} \tag{6.18}$$

In a general case, the integral (the expected log-likelihood) in Equation 6.18 is intractable and methods such as Gauss-Hermite quadrature [Hensman et al., 2015] or Monte Carlo sampling [Salimbeni and Deisenroth, 2017, Bonilla et al., 2018] can generally be used to calculate the expectation. The Gaussian-Hermite quadrature method is poor in a non-stationary GP model [Monterrubio-Gòmez et al., 2019, Monterrubio-Gòmez and Wade, 2021]. Also, in our case, the multi-dimensional integral in Equation 6.18 can not be computed using the Gauss-Hermite approach, since the observations $\mathbf{y}_i$ inside a segment i, are not independent given the parameters $\mathbf{l}_i$ and $\boldsymbol{\sigma}_i$. Thus, the multi-dimensional integral can not be split into multiple one-dimensional integrals given the non-stationary GP framework.

The Monte Carlo sampling method is used in Salimbeni and Deisenroth [2017], Saul et al. [2016] to calculate the expected log likelihood and this is the approach that we are following here. More specifically, given Equation 6.12, we draw samples l_{ij} and σ_{ij} from the multivariate Normal distributions, given standard properties of Gaussian distributions, $q(l_{ij}, \sigma_{ij})$. Then, we calculate the log likelihood of a trajectory segment $\log p(\mathbf{y}_i | l_{ij}, \sigma_{ij})$ per each sample we draw, where $\mathbf{y}_i$ is the segment i of observations and l_{ij}, σ_{ij} are the j-th sample of the respective parameter for segment i. We then proceed to take the average over all the samples drawn to calculate the expected log likelihood term in Equation 6.18. Moreover, in Equation 6.18, we have closed form expressions for the $\mathscr{K}\mathscr{L}$ divergence terms, as $\phi(\mathbf{l}_z, \boldsymbol{\sigma}_z)$, together with $p(\mathbf{l}_z)$ and $p(\boldsymbol{\sigma}_z)$ are multivariate Normal distributions, as shown in Equations 6.7-6.8 and Equation 6.12. Thus, once the lower bound is calculated, it can be maximised using stochastic optimisation to determine the optimal parameters of the joint variational distributions $\phi(\mathbf{l}_z, \boldsymbol{\sigma}_z)$ and other parameters of interest such as the lengthscale, signal variance parameters $\mathbf{l}$ and $\boldsymbol{\sigma}$ on the first layer of the GP. Also, all of the hyperparameters of the second layer of the GP such as α_P^2 and β_P are inferred. The inducing points are chosen as evenly spaced across the domain (about 50 points, one inducing point every half an hour for the empirical data) and can potentially be inferred too, but in this chapter they are kept fixed as the number of inducing points is sufficiently dense for the domain.

6.2.3 Model inference

We implement our variational framework inference in TensorFlow, an open-source deep learning library [Abadi et al., 2016], using the package TensorFlow Probability. Using the TensorFlow library has multiple advantages including access to automatic differentiation for an efficient and easy method to calculate the gradients, without the specification of analytical formulae. It also facilitates access to GPU-accelerated calculations. To process large amounts of data, we use a trajectory segmentation technique, where we break the individual trajectories into multiple and computationally more accessible segments.

We use one set of inducing points $\mathbf{z}$ for each segment, shared by each latent parameter $\mathbf{l}$ and $\boldsymbol{\sigma}$. These are kept fixed and are not inferred using the rest of the parameters. Using the

inducing points $\mathbf{z}$, we define the latent function values for the lower level Gaussian GPs. To obtain the values of the lengthscale and the variance on the first layer of the GP, we predict the latent function values at the observed data points, given the inducing points locations.

In this paper we infer only the latent functions $\mathbf{l}$ and $\boldsymbol{\sigma}$, but not the function $\mathbf{f}$. For the empirical dataset inference, this is because we are interested in the behaviour of the power consumption over a period time, and this is controlled by the parameters $\mathbf{l}$ and $\boldsymbol{\sigma}$. The period of time usually considered in an ordinary energy bill is a quarter or a year, but in this paper we analyse the evolution of the power consumption over a day. It is straightforward to adapt the method to study the behaviour of the power consumption over a larger period of time.

6.3 Data

6.3.1 Synthetic data generation

We generate synthetic data from a non-stationary GP model, with mean zero and the covariance kernel given by the non-stationary Mátern 1/2 kernel defined in Equation 6.2 with constant observation error. We generate $\mathbf{l}$ and $\boldsymbol{\sigma}$ by taking a sample from a GP prior with an RBF kernel, such that there is no mismatch between our inference model and the synthetic data.

We simulate from our model trajectories of 1, 8, respectively 128 individuals and collect about 8,000 observations per individual. In Figure 6.1, we show the observed synthetic dataset for 1 individual, and the values of the lengthscale and amplitude parameters generated from the RBF kernels. The values of these parameters are then used to generate data from a GP with the non-stationary Mátern 1/2 kernel, shown in Equation 6.2.

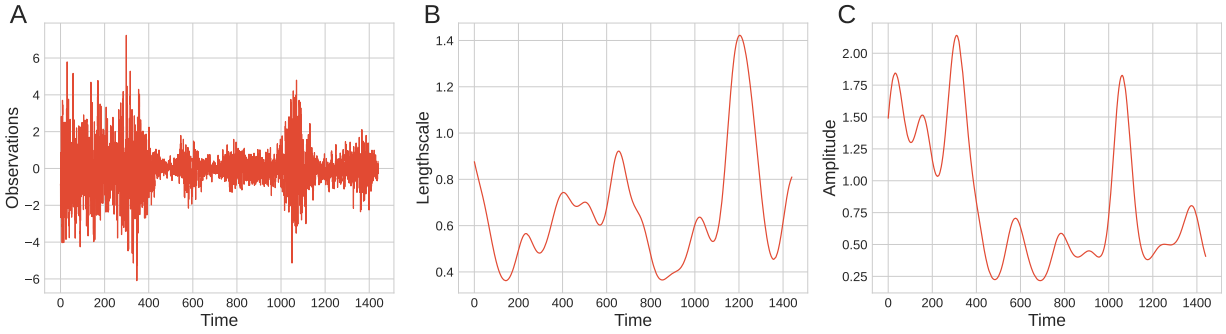


Figure 6.1: Synthetic data: (A) shows the observed synthetic data for 1 individual. (B) and (C) show the true lengthscale parameter, respectively the true amplitude parameter that generated the dataset shown in (A).

6.3.2 Empirical data

We apply our methods to empirical data, where the observations are the average power consumption usage per minute (in watts) [Dua and Graff, 2017] recorded in one individual household in

Paris, France. Our dataset consists of 44,640 readings recorded every single minute for a month, October 2017.

6.4 Results

We fit our non-stationary GP and we apply a variational inference method. We optimise the lower bound derived in Equation 6.18 using the Adam optimiser [Kingma and Ba, 2017]. All the parameters of the hierarchical GP are optimised, together with the parameters of the variational distributions $\phi(\mathbf{l}_z, \boldsymbol{\sigma}_z)$ in Equation 6.12.

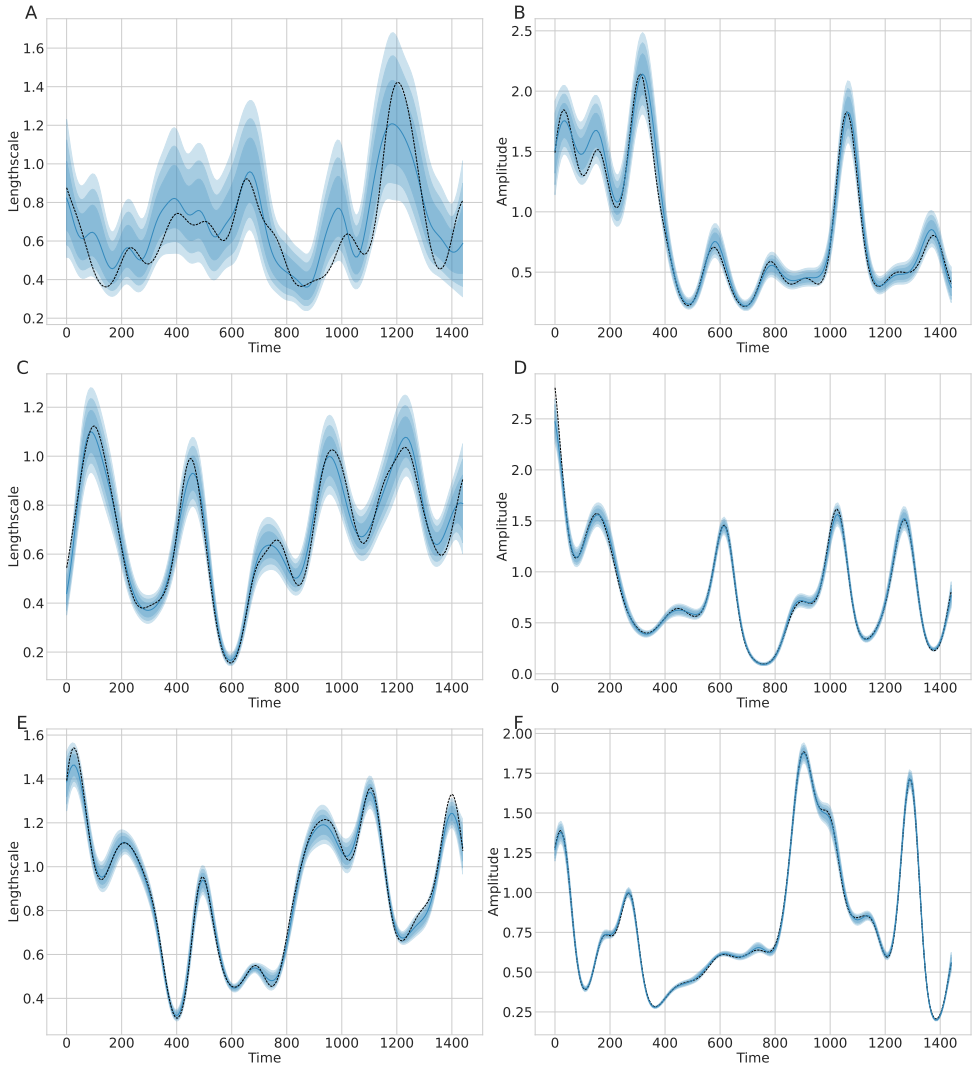


Figure 6.2: Synthetic inference: (A), (C), (E) show the inferred mean lengthscale parameter (blue line) for 1 individual, 8 individuals, respectively 128 individuals datasets. (B), (D), (F) show the inferred mean amplitude parameter (blue line) for 1 individual, 8 individuals, respectively 128 individuals datasets. The black dashed line represents the true parameter value. The blue regions (from dark to light) represent the 80%, 95%, respectively 99% credible intervals.

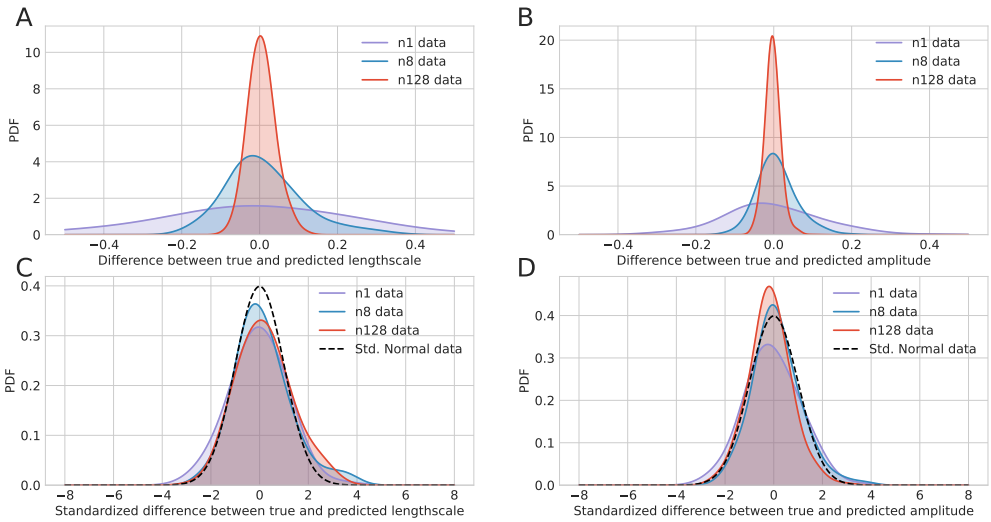


Figure 6.3: Synthetic inference kernel density estimation: Figures A and C show the pdfs of the difference, respectively of the standardized difference between the true and the predicted lengthscale. Figures B and D show the pdfs of the difference, respectively of the standardized difference between the true and predicted amplitude. The purple line is the pdf for the 1 individual dataset, the blue line is the pdf for the 8 individuals dataset and the red line is the pdf for the 128 individuals dataset. The blacked dashed line in Figures C and D represents the pdf of a standard Normal distribution.

6.4.1 Synthetic model inference results

The inference results for the synthetic model are shown in Figure 6.2 for 1, 8, respectively 128 individuals simulated trajectories. The means of the posterior distributions are shown together with the uncertainty quantification (80%, 95%, respectively 99% credible intervals) and with the ground-truth values. As more data is added (increasingly from top to bottom), the uncertainty decreases for both parameters, as expected. There is a very close agreement, between the inferred means of the parameters and the true values and almost all of the deviations from the true parameter values are within the credible intervals. In Figure 6.3 we show the pdfs for the difference and standardized differences between the true and the predicted parameters. To produce the plots we simulate 5 replicate datasets consisting of 1 individual, 8 individuals, respectively 128 individuals observations, perform inference for the parameters, compute the differences and the standardized differences between the true and the predicted parameters (computed by dividing the differences by the standard deviation of the posterior samples) for each dataset. Then, we calculate the pdfs using kernel density estimation after concatenating the differences/standard differences (the bandwith was selected using cross-validation). In Figures 6.3 A and B, as expected, the pdfs get more peaked around 0 as we add more data, signifying less differences between the true values of the parameters and the predicted parameters as more data is being analysed. In Figure 6.3 C, given the increasingly small standard deviation as more data is being

added, the pdf for the 8 individuals dataset (blue line) is more peaked around 0 than the pdfs for the 128 individuals dataset (red line) and for the 1 individual (purple line) dataset. In both Figures 6.3 C and D, there is a small discrepancy between our expectations and our results. In a case, where the gold standard is met, we would expect that all the pdfs would be overlapping and be perfectly bell-shaped and be centered around 0 with a standard deviation of 1, as is the pdf of a standardised Normal distribution (black dashed line).

6.4.2 Empirical data inference results

The inference results for the empirical model are shown in Figure 6.4. Figure 6.4 A measures how likely the average power consumption is to remain constant over a period of time. The power persistence is high during the night (12AM-5AM), given that the power usage remains constantly low during this time period. This is consistent with the low variance values and relatively low uncertainty in Figure 6.4 B during the same time frame. In Figure 6.4 A, from 7AM until midnight, the average power consumption persistence remains constantly low, signifying a fairly steady consumption of energy during these hours and this is in agreement with the steady average power consumption variance values in Figure 6.4 B, although there is higher uncertainty than during the night hours.

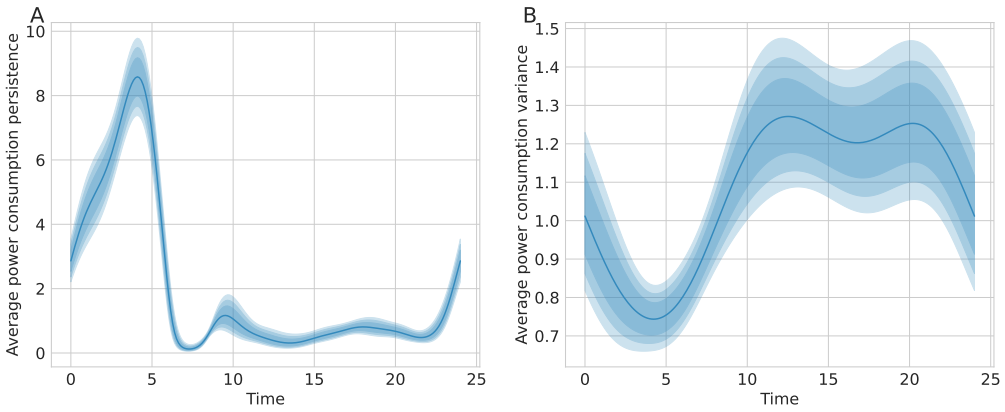


Figure 6.4: Empirical dataset inference: (A) shows the mean power consumption persistence (blue line). (B) shows the mean variance power usage (blue line). The blue regions (from dark to light) represent the 80%, 95%, respectively 99% credible intervals.

6.5 Conclusions

In this chapter we presented a variational Bayesian inference method within a hierarchical non-stationary GP framework. We combined the flexibility and robustness of non-stationary GP models with the computationally efficient variational inference scheme and with the mixture-

of-experts technique to make the model scalable to large datasets. We successfully applied our method to synthetic and empirical datasets producing reliable and intuitive results.

Our method presents significant advantages compared to other methods developed in the literature. The variational inference scheme is faster compared to MCMC methods that are difficult to sample efficiently from when using a non-stationary GP model, and unlike the MAP method that produces only point-estimates, it quantifies uncertainty. Moreover, using the mixture of experts technique, our method is scalable to large datasets compared to the small datasets' size used by Heinonen et al. [2016]. While the inference method is the same as Hensman et al. [2013], the model used is not a standard GP, but a hierarchical GP, which is more flexible than the standard GP and has strong real life applications potential in various domains where there is a lot of function variability in the input space, such as modelling animal movement or terrain surfaces. Future work might include changing the inference framework from a sparse variational Bayesian framework to a state-space formulation that performs inference in linear time [Grigorievskiy et al., 2016]. Moreover, future work might also consist in applying MCMC inference methods to the current model and data and drawing a comparison between the MCMC method and the current variational inference approach.

Chapter 7

Discussion and future work

Statistical models of movement data are divided into two types, based on its time-formulations - discrete-time or continuous-time. The discrete-time movement models such as the random walk and its extensions are the bedrock of movement models, and are preferred by ecologists since they are easy to understand and to implement. The continuous-time movement models arise as solutions to SDEs, and can more easily deal with irregular or missing observations than the discrete-time movement models. The more realistic continuous movement models, where various drivers of movement such as internal state or environmental characteristics are accounted for have seen a limited use by the ecologists due to their complexity in formulation, the reduced scalability of the inference methods to large datasets and high computational demands. Hence, in this thesis, I develop inference methods that are scalable to large movement datasets, where the autocorrelated nature of movement data and its multiscale complexity driven by environmental characteristics are addressed.

Original contributions start in Chapter 3 when through a study I illustrate one major disadvantage when working with discrete-time movement models - the specification in advance of the unknown discretisation-step. Hence, for the rest of the thesis, I work with continuous-time movement models, that do not present such limitations. I focus exclusively on non-parametric probabilistic methods, namely GPs, since their flexibility in choosing a covariance functions makes them equivalent to many continuous-time movement models. I show this fact in Chapter 4, where I demonstrate how popular and widely implemented movement models such as OUF [Fleming et al., 2014a] or OUV [Johnson et al., 2008] can be reintroduced as GPs. Among the benefits are working within a non-parametric Bayesian framework with access to powerful machine learning libraries such as TensorFlow [Abadi et al., 2016].

A parametric alternative to the non-parametric approach represented by GPs that is particularly popular and has been applied to animal movement [Whetten, 2021] is the splines model. An advantage that splines have over GPs is the reduction in computation complexity, which is linear rather than cubic in the data samples size. However, among the disadvantages of the splines model is the lack of flexibility in specifying the covariance function, which is not explic-

itly specified by the modeller. Another disadvantage is the lack of uncertainty quantification as the splines model is equivalent to the MAP point estimate of a GP [Rasmussen and Williams, 2006], Section 6.3.

Regarding modelling animal movement, GPs have been used recently [Cobb et al., 2017, Torney et al., 2021]. Fleming et al. [2014a] illustrate a flexible and rigorous method that makes use of the first two cumulants of a stochastic process (the mean and autocorrelation function, thus, basically defining a GP). Another important example is Hooten and Johnson [2017], where the authors make use of basis functions and convolutions to obtain smooth continuous-time movement models (thus, leading to a GP), called functional movement models.

In Chapters 5-6, I extend the stationary GP to a hierarchical non-stationary GP, by having a subset of the parameters be modelled by another GPs [Heinonen et al., 2016]. The non-stationary GPs are capable of modelling non-stationary data, where there are different levels of function smoothness in the input space and can be considered a continuous-time alternative to the discrete-time model HMM. In the literature, continuous-time movement models that incorporate different drivers of movement are called state-switching models [Harris and Blackwell, 2013, Blackwell et al., 2016]. Other flexible models make use of potential functions, which incorporate attraction points and landscape characteristics. These models have been been developed using stochastic differential equations [Preisler et al., 2001, 2004, 2013, Brillinger et al., 2001, 2002, 2004, Brillinger, 2010]. Other methods that incorporate the landscape characteristics are wavelet analysis [Wittemyer et al., 2008b], step-selection functions [Michelot et al., 2020, Thurfjell et al., 2014] and integrated step-selection analysis [Avgar et al., 2016, Prokopenko et al., 2017]. An interesting example is presented in Cobb et al. [2017], where the authors use a different approach, namely a GP to infer a spatio-temporal vector field from observed trajectories of an animal. Arguably, my approach is more flexible, due to the hierarchical nature of the GP models, as it allows to model a variety of behavioural states that the animals can exhibit in a smooth and continuous manner. This is done through the selection of preferred covariance functions on the second layer of the hierarchical GP that control the smoothness and structure of the movement parameters on the first layer.

More specifically, in Chapter 5, I consider a spatial hierarchical non-stationary GP model, where the parameters on the first-layer of the GP depend on the actual telemetry locations. I fit the model to relate these latent fields to simulated trajectories in a sinusoidal environment and in empirical movement tracks of wildebeest by using a non-stationary version of the correlated velocity model. The model is used to analyse the wildebeest migration and detect regions of high and low directional persistence and speed. The scalability of the model is improved to relatively large datasets and to accomplish this, I implement it in an open-source deep learning platform, TensorFlow [Abadi et al., 2016]. TensorFlow has multiple advantages over other packages that implement GPs such as GPy as it uses automatic differentiation to calculate the gradients of the log likelihood with respect to the parameters in a straightforward manner. Also, it provides

a significant computational boost as it allows fast computations on GPU, unlike GPy or other libraries that implement GPs, that perform calculations on CPU. To make the method able to process large amounts of data I use the method of trajectory segmentation, where I divide the individual trajectories into multiple and computationally more accessible sections. I follow a sparse GP approach, where I define the GP using a set of inducing or support points in order to reduce the computational complexity of $\mathcal{O}(N^3)$ of a GP. Moreover, I perform distributed training of my data on multiple GPUs, further increasing the inference speed on large datasets. However, this method is not scalable to millions of datapoints as it has the drawback that sampling based methods such as MCMC are computationally expensive, might mix poorly and have trouble reaching convergence.

In Chapter 6, I develop a model scalable to potential millions of points. I accomplish this by modifying the inference scheme used in Chapter 5. Instead of using sampling-based methods such as MCMC, I employ variational inference methods. While I infer just an approximate posterior distribution that is constructed and optimised to get as close as possible to the true posterior distribution, the method is now scalable to very large datasets. I test the methods on multiple synthetic and real world datasets.

Future work for Chapter 5 might include model checking that the GP model is a good fit to the real data. I could use a similar approach to the one used in Chapter 3 by using summary statistics and posterior predictive p-values. I could compare the observed lengthscale with the simulated lengthscales, or compare the observed environmental characteristics with the simulated environmental characteristics. If the posterior predictive values are close to the extremes, this could indicate a systematic model mismatch. In addition, I could use the variational inference method instead of the slow-mixing MCMC to infer the spatial-latent fields. Comparisons between the accuracy and the speed of the MCMC versus the variational inference method could be performed. Further work could be done by introducing various covariates into the hierarchical GP model such as distance to boundaries, richness of soil, quantity of rainfall, numbers of tourists visiting the site, location of human settlements, etc.. While the model currently accepts spatial location as inputs, it can be modified such that the inputs can be the environmental covariates aforementioned. The goal would be to improve the general contribution of my framework to gain new insights in the movement ecology and behaviour of individual animals. Moreover, another significant research topic that could be further developed is to make the framework accessible to ecologists by creating a package that can implement non-stationary hierarchical GPs to model tracking data. Furthermore, allowing for the inclusion of environmental covariates might make my approach suitable to a wider ecological audience. A similar R package that implements HMMs in an easy and accessible manner to ecologists was developed by Michelot et al. [2016]. Future work for Chapter 6 might include changing the variational inference method to MCMC and subsequently a comparison could be made between these two methods. The aforementioned ideas are a promising and fascinating area for future research.

Appendix A

Appendix section for Chapter 2

A.1 Induction proof

I prove by induction that the identity in Equation 2.115 holds. Let

$$P(N): a_0 + a_1(iw)^2 + \cdots + a_N(iw)^{2N} = \left(b_0 + b_1(iw) + \cdots + b_N(iw)^N\right)\left(b_0 + b_1(-iw) + \cdots + b_N(-iw)^N\right). \tag{A.1}$$

I check first if the statement holds for $N = 1$. I have that

$$a_0 + a_1(iw)^2 = (b_0 + b_1(iw))(b_0 - b_1(iw)). \tag{A.2}$$

Equating the coefficients results we have

$$a_0 = b_0^2. \tag{A.3}$$

$$a_1 = -b_1^2. \tag{A.4}$$

Therefore, the coefficients b_0 an b_1 can be found such that $P(N)$ for $N = 1$ holds.

I prove that $P(N) \to P(N+1)$. I have

$$\begin{aligned} P(N+1): a_0 + a_1(iw)^2 + \cdots + a_N(iw)^{2N} + a_{N+1}(iw)^{2N+2} \\ = \left(b_0 + b_1(iw) + \cdots + b_N(iw)^N + b_{N+1}(iw)^{N+1}\right) \\ \times \left(b_0 + b_1(-iw) + \cdots + b_N(-iw)^N + b_{N+1}(-iw)^{N+1}\right). \end{aligned} \tag{A.5}$$

For simplicity let $k = b_0 + b_1(iw) + \cdots + b_N(iw)^N$ and $l = b_0 + b_1(-iw) + \cdots + b_N(-iw)^N$. Therefore, the RHS of $P(N+1)$ is

$$\begin{aligned} \text{RHS} &= \left(k + b_{N+1}(iw)^{N+1}\right)\left(l + b_{N+1}(-iw)^{N+1}\right) \\ &= kl + kb_{N+1}(-iw)^{N+1} + lb_{N+1}(iw)^{N+1} + b_{N+1}^2(-1)^{N+1}(iw)^{2N+2}. \end{aligned} \tag{A.6}$$

Now the induction step can be used. The coefficients c_i exist such that the following relationship holds

$$kl = c_0 + c_1(iw)^2 + \cdots + c_N(iw)^{2N}. \tag{A.7}$$

I calculate the second and third terms in Equation A.6 by replacing k and l and I get

$$\begin{aligned}
&\left(b_0 + b_1(iw) + \cdots + b_N(iw)^N\right) b_{N+1}(-iw)^{N+1} + \left(b_0 + b_1(-iw) + \cdots + b_N(-iw)^N\right) b_{N+1}(iw)^{N+1} \\
&= b_0 b_{N+1}(-iw)^{N+1} + b_1 b_{N+1}(-iw)^{N+1}(iw) + \cdots + b_N b_{N+1}(iw)^N(-iw)^{N+1} + b_0 b_{N+1}(iw)^{N+1} \\
&+ b_1 b_{N+1}(-iw)(iw)^{N+1} + \cdots + b_N b_{N+1}(-1)^{N+1}(iw)^{2N} \\
&= b_0 b_{N+1}(iw)^{N+1}\left((-1)^{N+1} + 1\right) + b_1 b_{N+1}(iw)^{N+2}\left((-1)^{N+1} - 1\right) + \cdots + \\
&+ b_N b_{N+1}(iw)^{2N+1}\left((-1)^{N+1} + (-1)^N\right).
\end{aligned} \tag{A.8}$$

From this equation, it can be seen that whether N is even or odd, the odd powers of (iw) disappear. Combining Equations A.5, A.6, A.7 and A.8, the coefficients b_i's can always be found such that Equation A.5 holds. Therefore, the identity in Equation 2.115 or Equation A.1 is true.

Appendix B

Appendix for Chapter 3

B.1 Validating the first model

Introduction

In this section we validate the first model from Chapter 3, Section 3.2.2 by using a different prior than a uniform distribution for the scale parameter of the Weibull distribution. The new prior chosen is the Inverse-Gamma distribution, which is a conjugate prior for the scale parameter of the Weibull distribution when the shape parameter is kept fixed. Using this prior allows the calculation of the analytical posterior distribution of the scale parameter. The MCMC algorithm of choice is the MH instead of the natural MCMC method, Gibbs sampling, because the aim is to test the MH implementation by comparing the posterior samples distribution against the analytical posterior pdf (more details are offered in Chapter 3, Section 3.2.4). A Kolmogorov–Smirnov [Hodges, 1958] test is calculated that checks whether the underlying distribution of the posterior samples is identical to the analytical posterior distribution. If the KS test returns a small KS statistic or a high p-value, then the null hypothesis that the underlying distribution of the posterior samples is identical to the analytical posterior distribution cannot be rejected in favour of the alternative, that is the underlying distribution of the posterior samples is not identical to the analytical posterior distribution.

Data

Step-lengths data is simulated from a Weibull distribution with shape parameter 4 and scale parameter 1 and the turning angles data is simulated from a Uniform $(0, 2\pi)$ distribution. The dataset size is 5000 for each component.

Models

We implement the first model from Section 3.2.2. Let r_t represent the observed step-lengths and let θ_t represent the associated observed turning angle. The model considered is

$$r_t \sim \text{Weibull}(a,b),$$
$$\theta_t \sim \text{Uniform}(0,2\pi).$$

The pdf of the Weibull distribution is

$$p(r|a,b) = \frac{ar^{a-1}}{b}\exp\left(-\frac{r^a}{b}\right), \tag{B.1}$$

where $r \geq 0$, a is the shape parameter and b is the scale parameter, both positive parameters.

Prior distribution

We set an Inverse-Gamma (1,1) prior on the scale parameter b. The pdf of the Inverse-Gamma(α, β) distribution is

$$p(b|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}b^{-\alpha-1}\exp\left(\frac{-\beta}{b}\right), \tag{B.2}$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter.

Likelihood calculation

Suppose the step-lengths $r_1,\ldots,r_n$ are independent and identically distributed draws from a Weibull distribution, where the scale parameter b is unknown and the shape parameter a is known. The likelihood function is proportional to

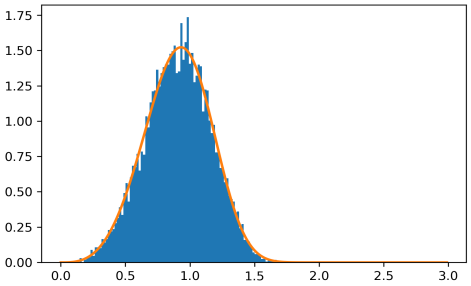
$$\mathscr{L}(b;r) \propto b^{-n}\exp\left(\frac{-\sum_{i=1}^n r_i^a}{b}\right). \tag{B.3}$$

Using the prior distribution defined in Section B.1 and keeping in mind that the posterior probability is proportional to the likelihood multiplied by the prior probability we get that the posterior distribution $\pi(b|a',b')$ is Inverse-Gamma with shape a' and scale b'

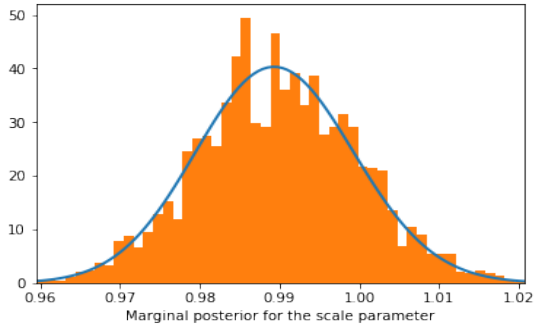
$$a' = \alpha + n. \tag{B.4}$$
$$b' = \beta + \sum_{i=1}^n r_i^a. \tag{B.5}$$

Inference

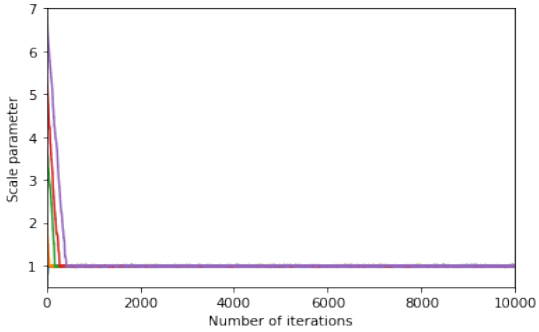
The algorithm chosen to infer the scale parameter is the MH and the proposal distribution is a symmetric Normal distribution. The number of iterations is 10,000.



(a) Histogram of the step-lengths from the synthetic data and the analytical pdf of Weibull (4,1) (orange line).



(b) Plot of the analytical posterior pdf of the scale parameter (blue line) and the histogram of the marginal posterior samples of the scale parameter.



(c) Traceplots of the scale parameter of the Weibull distribution starting from different initialisations.

Figure B.1: Inference and convergence plots when the prior is the Inverse-Gamma distribution.

Results of the inference

In Figure B.1a we plot the distribution of step-lengths from the data and the analytical pdf of Weibull (4,1) distribution. In Figure B.1b we plot the marginal posterior samples for the scale parameter and we fit the analytical posterior pdf using Equation B.2 with the corresponding shape and scale parameters from Section B.1, Equations B.4 and B.5. There is an agreement with the two plots and this is confirmed by the KS test which returned a KS statistic value of 0.048 and a p-value of 0.199. Since the p-value is not extreme, we cannot reject the null hypothesis that the underlying distribution of the posterior samples is identical to the analytical posterior distribution.

Assessing convergence

The Gelman-Rubin statistic for the scale parameter is $\hat{R} = 1.009$, which is smaller than 1.1, indicating that the chain does not show a lack of convergence. Also, by analysing the traceplots

in Figure B.1c, we do not detect a lack of convergence.

B.2 Conclusions

In this section we fitted the first model from Section 3.2.2 with a conjugate prior for the scale parameter while keeping the shape parameter fixed at 4. After performing inference using the MH algorithm and checking for convergence, the posterior samples were compared to the analytical posterior distribution. A KS test was calculated and the null hypothesis that the underlying distribution of the posterior samples is identical to the analytical posterior distribution could not be rejected. Thus, the MH algorithm implementation was successful.

Appendix C

Appendix section for Chapter 4

C.1 Brownian bridge covariance function derivation

In this subsection we show the Brownian bridge covariance function derivation. This derivation is based on Ibe [2016], Chapter 9, Section 9.9, pages 270-271.

The Brownian bridge model is Brownian motion restricted on the interval $[0,1]$ and can be defined as follows

$$\{W(t),\ t \in [0,1] |\ W(1) = 0\}, \tag{C.1}$$

where $W(t)$ is the Brownian motion process realisation at time t. We can redefine the Brownian bridge process as

$$x(t) = W(t) - tW(1),\ 0 \leq t \leq 1, \tag{C.2}$$

where $x(t)$ is the realisation of the Brownian bridge process at time t. From Equation 4.1 we can deduce that $\mathbb{E}[x(t)] = 0$, given that $\mathbb{E}[W(t)] = 0$ at any time t. We have that for $0 \leq s < t \leq 1$, the covariance of $x(t)$ and $x(s)$ is given by

$$\begin{aligned}
\text{Cov}[x(s),x(t)] &= \mathbb{E}[\{x(s) - \mathbb{E}[x(s)]\}]\{x(t) - \mathbb{E}[x(t)]\} = \mathbb{E}[x(t)x(s)] \\
&= \mathbb{E}[\{W(s) - sW(1)\}\{W(t) - tW(1)\}] \\
&= \mathbb{E}[W(s)W(t) - tW(s)W(1) - sW(t)W(1) + stW^2(1)] \\
&= \sigma^2 \min(s,t) - \sigma^2 t \min(s,1) - \sigma^2 s \min(t,1) + \sigma^2 st \\
&= \sigma^2\{s - st - st + st\} = \sigma^2(s - st) \\
&= \sigma^2 s(1-t),
\end{aligned} \tag{C.3}$$

where we used the fact that $\mathbb{E}[W(s)W(t)] = \sigma^2 \min(s,t)$.

C.2 Multivariate OU process covariance function derivation

We might be interested using a multivariate OU process instead of the univariate case. The multivariate stochastic differential equation of the OU process is

$$d\mathbf{x}_t = \mathbf{a}(\mathbf{b} - \mathbf{x}_t)dt + \boldsymbol{\sigma} d\mathbf{W}_t, \tag{C.4}$$

where $\mathbf{a}$ is a $n \times n$ invertible matrix, $\mathbf{b}$ is a n-dimensional real vector, $\boldsymbol{\sigma}$ is a $n \times m$ positive real matrix and $\mathbf{W}_t$ is a m-dimensional Wiener process at a time point t.

Following a similar procedure to the univariate case or by following the method used by Vatiwutipong and Phewchean [2019] the n-dimensional OU process $\mathbf{x}_t$ has a n-dimensional Normal distribution with mean vector

$$\mathbf{m}_t = e^{-\mathbf{a}t}\mathbf{x}_0 + \left(\mathbf{I} - e^{-\mathbf{a}t}\right)\mathbf{b}, \tag{C.5}$$

and covariance matrix

$$\boldsymbol{\Sigma}_t = \int_0^t e^{\mathbf{a}(s-t)}\boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{\mathbf{a}^T(s-t)}ds, \tag{C.6}$$

for a time point t. We can obtain the covariance between two points $\mathbf{x}_s$ and $\mathbf{x}_t$ either by following the procedure as in the univariate case or by using Theorem 2 of Vatiwutipong and Phewchean [2019]

$$\mathrm{Cov}(\mathbf{x}_s, \mathbf{x}_t) = \int_0^{\min(s,t)} e^{-\mathbf{a}(s-u)}\boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{-\mathbf{a}^T(t-u)}du. \tag{C.7}$$

We can do further work on the previous results such that the following relationship holds

$$\begin{aligned}
\mathrm{Cov}(\mathbf{x}_s, \mathbf{x}_t) &= \boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{-(\mathbf{a}s + \mathbf{a}^T t)} \int_0^{\min(s,t)} e^{u(\mathbf{a}+\mathbf{a}^T)}du \\
&= \boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{-(\mathbf{a}s + \mathbf{a}^T t)} \left(\mathbf{a} + \mathbf{a}^T\right)^{-1} e^{u(\mathbf{a}+\mathbf{a}^T)}\Big|_{u=0}^{u=\min(s,t)} \\
&= \boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{-(\mathbf{a}s + \mathbf{a}^T t)} \left(\mathbf{a} + \mathbf{a}^T\right)^{-1} \left(e^{\min(s,t)(\mathbf{a}+\mathbf{a}^T)} - 1\right) \\
&= \boldsymbol{\sigma}\boldsymbol{\sigma}^T \left(\mathbf{a} + \mathbf{a}^T\right)^{-1} \left(e^{\frac{-\left(\mathbf{a}+\mathbf{a}^T\right)}{2}((s+t-|s-t|)-\left(\mathbf{a}s+\mathbf{a}^T t\right)} - e^{-\left(\mathbf{a}s+\mathbf{a}^T t\right)}\right).
\end{aligned} \tag{C.8}$$

As s and t grow large, we have that

$$\mathrm{Cov}(\mathbf{x}_s, \mathbf{x}_t) = \boldsymbol{\sigma}\boldsymbol{\sigma}^T \left(\mathbf{a} + \mathbf{a}^T\right)^{-1} e^{-\frac{\left(\mathbf{a}+\mathbf{a}^T\right)|s-t|}{2}}. \tag{C.9}$$

C.3 OUF kernel function implementation in GPy

In this Appendix section, we implement the OUF kernel in GPy, a library that can implement GPs. In order to do this, we simulate OUF data using the Equations 4.53 and 4.54 and then fit a GP with the OUF kernel function derived in Equation 4.65. The OUF kernel is not a standard kernel implemented in GPy, therefore we create a new kernel inside the package. Since we want to do inference either by using an optimiser or HMC, we need the derivatives of the OUF kernel in Equation 4.65 with respect to all the parameters of interest that we want to infer. We show all the derivations in the following subsections.

OUF kernel derivatives with respect to the parameters

The OUF kernel formula, reproduced here again for clarity is

$$k(t,t') = \frac{\sigma_a \tau_H^2 \tau_F^2}{2(\tau_F + \tau_H)} \left(\frac{\tau_H e^{\frac{-|t-t'|}{\tau_H}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}}}{\tau_H - \tau_F} \right). \tag{C.10}$$

We denote

$$r_H = \frac{|t-t'|}{\tau_H}. \tag{C.11}$$

$$r_F = \frac{|t-t'|}{\tau_F}. \tag{C.12}$$

Let

$$Ke(\tau_H) = Ke(\tau_F) = \frac{\tau_H e^{\frac{-|t-t'|}{\tau_H}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}}}{\tau_H - \tau_F} = \frac{\tau_H e^{-r_H} - \tau_F e^{-r_F}}{\tau_H - \tau_F}, \tag{C.13}$$

and let

$$f(\tau_H) = f(\tau_F) = \tau_H e^{\frac{-|t-t'|}{\tau_H}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}} = \tau_H e^{-r_H} - \tau_F e^{-r_F}. \tag{C.14}$$

Also, we denote

$$g(\tau_H) = g(\tau_F) = \frac{\sigma_a \tau_H^2 \tau_F^2}{2(\tau_F + \tau_H)}. \tag{C.15}$$

We calculate first the derivative of the OUF kernel with respect to the σ_a parameter such that we have

$$\frac{dk}{d\sigma_a} = \frac{\tau_H^2 \tau_F^2}{2(\tau_F + \tau_H)} \times Ke. \tag{C.16}$$

The derivative with respect to the τ_H parameter is

$$\frac{dk}{d\tau_H} = (g(\tau_H) \times Ke(\tau_H))' = g'(\tau_H) \times Ke(\tau_H) + Ke'(\tau_H) \times g(\tau_H). \tag{C.17}$$

We calculate the derivatives separately and we get

$$\begin{aligned} g'(\tau_H) &= \frac{\sigma_a \tau_F^2}{2} \frac{2\tau_H(\tau_H+\tau_F)-\tau_H^2}{(\tau_H+\tau_F)^2} \\ &= \frac{\sigma_a \tau_F^2}{2} \frac{\tau_H^2+2\tau_H\tau_F}{(\tau_H+\tau_F)^2}. \end{aligned} \tag{C.18}$$

We calculate the derivative of Ke with respect to τ_H and we obtain

$$Ke'(\tau_H) = \frac{f'(\tau_H)(\tau_H-\tau_F)-f(\tau_H)}{(\tau_H-\tau_F)^2}, \tag{C.19}$$

where we have that $f = \tau_H e^{-r_H} - \tau_F e^{-r_F}$. We compute the derivative of f with respect to τ_H and we get

$$\begin{aligned} f'(\tau_H) &= (\tau_H e^{-r_H})' = e^{-r_H} + \tau_H e^{-r_H}(-r_H)' \\ &= e^{-r_H} + e^{-r_H} r_H \\ &= e^{-r_H}(1+r_H), \end{aligned} \tag{C.20}$$

where we used the fact $(r_H)' = \frac{-1}{\tau_H^2}|t-t'|$. Inserting the derivative of f into Equation C.19 we get that

$$Ke'(\tau_H) = \frac{e^{-r_H}(1+r_H)(\tau_H-\tau_F)-(\tau_H e^{-r_H}-\tau_F e^{-r_F})}{(\tau_H-\tau_F)^2}. \tag{C.21}$$

We have that $k(\tau_H) = g(\tau_H) \times Ke(\tau_H)$. Therefore, $\frac{dk}{\tau_H} = g'(\tau_H) \times Ke(\tau_H) + g(\tau_H) \times Ke'(\tau_H)$. Then, using the derivatives of g and Ke we get that

$$\begin{aligned} \frac{dk}{\tau_H} = &\frac{\sigma_a \tau_F^2}{2} \frac{\tau_H^2+2\tau_H\tau_F}{(\tau_H+\tau_F)^2} \frac{\tau_H e^{-r_H}-\tau_F e^{-r_F}}{\tau_H-\tau_F} \\ &+ \frac{\sigma_a \tau_H^2 \tau_F^2}{2(\tau_F+\tau_H)} \frac{e^{-r_H}(1+r_H)(\tau_H-\tau_F)-(\tau_H e^{-r_H}-\tau_F e^{-r_F})}{(\tau_H-\tau_F)^2}. \end{aligned} \tag{C.22}$$

Similarly, we calculate the derivative of k with respect to τ_F, and we obtain that

$$\frac{dk}{d\tau_F} = (g(\tau_F) \times Ke(\tau_F))' = g'(\tau_F) \times Ke(\tau_F) + Ke'(\tau_F) \times g(\tau_F), \tag{C.23}$$

We calculate the derivatives separately, and we get that

$$\begin{aligned} g'(\tau_F) &= \frac{\sigma_a \tau_H^2}{2} \frac{2\tau_F(\tau_H+\tau_F)-\tau_F^2}{(\tau_H+\tau_F)^2} \\ &= \frac{\sigma_a \tau_H^2}{2} \frac{\tau_F^2+2\tau_H\tau_F}{(\tau_H+\tau_F)^2}. \end{aligned} \tag{C.24}$$

We compute the derivative of Ke with respect to τ_F, and we obtain that

$$Ke'(\tau_F) = \frac{f'(\tau_F) \times (\tau_H - \tau_F) + f(\tau_F)}{(\tau_H - \tau_F)^2}. \tag{C.25}$$

We have that $f(\tau_F) = \tau_H e^{-r_H} - \tau_F e^{-r_F}$. We calculate the derivative of f with respect to τ_F such that we have

$$\begin{aligned} f'(\tau_F) &= -(\tau_F e^{-r_F})' = -\left(e^{-r_F} + \tau_F e^{-r_F}(-r_F)'\right) \\ &= -e^{-r_F} - e^{-r_F} r_F \\ &= -e^{-r_F}(1 + r_F), \end{aligned} \tag{C.26}$$

where we used the fact $(r_F)' = \frac{-1}{\tau_F^2}|t - t'|$. Inserting the derivative of f into Equation C.25 we get that

$$Ke'(\tau_F) = \frac{-e^{-r_F}(1 + r_F)(\tau_H - \tau_F) + (\tau_H e^{-r_H} - \tau_F e^{-r_F})}{(\tau_H - \tau_F)^2}. \tag{C.27}$$

We have that $k(\tau_F) = g(\tau_F) \times Ke(\tau_F)$. Therefore, $\frac{dk}{\tau_F} = g'(\tau_F) \times Ke(\tau_F) + g(\tau_F) \times Ke'(\tau_F)$, and using the derivatives of g and Ke we get that

$$\begin{aligned} \frac{dk}{\tau_F} = &\frac{\sigma_a \tau_H^2}{2} \frac{\tau_F^2 + 2\tau_H \tau_F}{(\tau_H + \tau_F)^2} \frac{\tau_H e^{-r_H} - \tau_F e^{-r_F}}{\tau_H - \tau_F} \\ &+ \frac{\sigma_a \tau_H^2 \tau_F^2}{2(\tau_F + \tau_H)} \frac{-e^{-r_F}(1 + r_F)(\tau_H - \tau_F) + (\tau_H e^{-r_H} - \tau_F e^{-r_F})}{(\tau_H - \tau_F)^2}. \end{aligned} \tag{C.28}$$

OUF kernel rewritten

We rewrite the OUF kernel from Equation 4.65 to allow for the difference $\tau_H - \tau_F$ to be always different from zero. We choose the difference $\tau_H - \tau_F$ to be always positive. We do this by defining another parameter $\delta = \tau_H - \tau_F$ and use a log exp transformation to make the parameter δ to be always positive. We also rewrite the OUF kernel formula from Equation 4.65 in terms of the parameters σ_a, τ_F and δ.

$$K = \frac{\sigma_a (\tau_F + \delta)^2 \tau_F^2}{2(2\tau_F + \delta)} \frac{(\tau_F + \delta) e^{-r_{\delta F}} - \tau_F e^{-r_F}}{\delta}, \tag{C.29}$$

where we have that

$$r_{\delta F} = \frac{|t - t'|}{\delta + \tau_F}. \tag{C.30}$$

$$r_F = \frac{|t - t'|}{\tau_F}. \tag{C.31}$$

We denote

$$Ke(\delta) = Ke(\tau_F) = \frac{(\tau_F+\delta)e^{\frac{-|t-t'|}{\tau_F+\delta}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}}}{\delta} = \frac{(\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}}{\delta}, \tag{C.32}$$

and let

$$f(\delta) = f(\tau_F) = (\tau_F+\delta)e^{\frac{-|t-t'|}{\tau_F+\delta}} - \tau_F e^{\frac{-|t-t'|}{\tau_F}} = (\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}. \tag{C.33}$$

Also, we denote

$$g(\delta) = g(\tau_F) = \frac{\sigma_a(\tau_F+\delta)^2\tau_F^2}{2(2\tau_F+\delta)}. \tag{C.34}$$

Firstly, we calculate the derivative of the OUF kernel with respect to the σ_a parameter such that

$$\frac{dk}{d\sigma_a} = \frac{(\tau_F+\delta)^2\tau_F^2}{2(2\tau_F+\delta)} \times Ke. \tag{C.35}$$

We compute the derivative with respect to the δ parameter and we get

$$\frac{dk}{d\delta} = (g(\delta) \times Ke(\delta))' = g'(\delta) \times Ke(\delta) + Ke'(\delta) \times g(\delta), \tag{C.36}$$

and we calculate the derivatives separately to obtain

$$\begin{aligned} g'(\delta) &= \frac{\sigma_a\tau_F^2}{2}\frac{2(\tau_F+\delta)(2\tau_F+\delta) - (\tau_F+\delta)^2}{(2\tau_F+\delta)^2} \\ &= \frac{\sigma_a\tau_F^2}{2}\frac{3\tau_F^2+4\tau_F\delta+\delta^2}{(2\tau_F+\delta)^2}. \end{aligned} \tag{C.37}$$

We compute the derivative of Ke with respect to δ and we get

$$Ke'(\delta) = \frac{f'(\delta)\delta - f(\delta)}{\delta^2}, \tag{C.38}$$

where we have that $f = (\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}$. Now, we calculate the derivative of f with respect to δ such that we obtain

$$\begin{aligned} f'(\delta) &= ((\tau_F+\delta)e^{-r_{\delta F}})' = e^{-r_{\delta F}} + (\tau_F+\delta)e^{-r_{\delta F}}(-r_{\delta F})' \\ &= e^{-r_{\delta F}} + e^{-r_{\delta F}}r_{\delta F} = e^{-r_{\delta F}}(1+r_{\delta F}), \end{aligned} \tag{C.39}$$

where we used the fact $(r_{\delta F})' = \frac{-1}{(\delta+\tau_F)^2}|t-t'|$. Inserting the derivative of f into Equation C.38 results in

$$Ke'(\delta) = \frac{e^{-r_{\delta F}}(1+r_{\delta F})\delta - ((\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F})}{\delta^2}. \tag{C.40}$$

We have that $k(\delta) = g(\delta) \times Ke(\delta)$. Therefore, $\frac{dk}{d\delta} = g'(\delta) \times Ke(\delta) + g(\delta) \times Ke'(\delta)$. Using the

derivatives of g and Ke we obtain

$$\begin{aligned}\frac{dk}{d\delta} = \frac{\sigma_a \tau_F^2}{2} \frac{3\tau_F^2 + 4\tau_F\delta + \delta^2}{(2\tau_F+\delta)^2} \frac{(\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}}{\delta} \\ + \frac{\sigma_a(\tau_F+\delta)^2\tau_F^2}{2(2\tau_F+\delta)} \frac{e^{-r_{\delta F}}(1+r_{\delta F})\delta - ((\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F})}{\delta^2}.\end{aligned} \tag{C.41}$$

Similarly, we calculate the derivative of k with respect to τ_F to get

$$\frac{dk}{d\tau_F} = (g(\tau_F) \times Ke(\tau_F))' = g'(\tau_F) \times Ke(\tau_F) + Ke'(\tau_F) \times g(\tau_F). \tag{C.42}$$

We calculate the derivatives separately, and we obtain

$$g'(\tau_F) = \frac{\sigma_a}{2} \frac{u'(2\tau_F+\delta) - 2u}{(2\tau_F+\delta)^2}, \tag{C.43}$$

where $u(\tau_F) = (\tau_F+\delta)^2\tau_F^2 = (\tau_F^2 + \delta\tau_F)^2$. We compute the derivative of u with respect to τ_F such that

$$\begin{aligned}\frac{du}{d\tau_F} &= 2(\tau_F^2+\delta\tau_F)(2\tau_F+\delta) \\ &= 2(2\tau_F^3 + \delta\tau_F^2 + 2\delta\tau_F^2 + \delta^2\tau_F) \\ &= 4\tau_F^3 + 6\delta\tau_F^2 + 2\delta^2\tau_F.\end{aligned} \tag{C.44}$$

Therefore, we get that

$$g'(\tau_F) = \frac{\sigma_a}{2} \frac{(4\tau_F^3 + 6\tau_F^2\delta + 2\delta^2\tau_F)(2\tau_F+\delta) - 2(\tau_F^2+\delta\tau_F)^2}{(2\tau_F+\delta)^2}. \tag{C.45}$$

We calculate the derivative of Ke with respect to τ_F, and we obtain

$$Ke'(\tau_F) = \frac{f'(\tau_F)}{\delta}. \tag{C.46}$$

We have that $f(\tau_F) = (\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}$. Then, we calculate the derivative of f with respect to τ_F such that

$$\begin{aligned}f'(\tau_F) &= \left((\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}\right)' \\ &= e^{\frac{-r}{\tau_F+\delta}} + (\tau_F+\delta)\left(e^{\frac{-r}{\tau_F+\delta}}\right)' - \left(e^{\frac{-r}{\tau_F}} + \tau_F\left(e^{\frac{-r}{\tau_F}}\right)'\right) \\ &= e^{\frac{-r}{\tau_F+\delta}} + r_{\delta F}e^{-r_{\delta F}} - (e^{\frac{-r}{\tau_F}} + e^{\frac{-r}{\tau_F}}r_F) \\ &= e^{-r_{\delta F}}(1+r_{\delta F}) - e^{-r_F}(1+r_F),\end{aligned} \tag{C.47}$$

where we used the fact $(r_F)' = \frac{-1}{\tau_F^2}|t-t'|$, $(r_{\delta F})' = \frac{-1}{(\delta+\tau_F)^2}|t-t'|$, and we denoted $|t-t'| = r$.

Inserting the derivative of f into Equation C.46 we get that

$$Ke'(\tau_F) = \frac{e^{-r_{\delta F}}(1+r_{\delta F}) - e^{-r_F}(1+r_F)}{\delta}. \tag{C.48}$$

We have that $k(\tau_F) = g(\tau_F) \times Ke(\tau_F)$. Therefore, $\frac{dk}{\tau_F} = g'(\tau_F) \times Ke(\tau_F) + g(\tau_F) \times Ke'(\tau_F)$. Finally, using the derivatives of g and Ke we get that

$$\begin{aligned}\frac{dk}{\tau_F} = \frac{\sigma_a}{2}\frac{(4\tau_F^3 + 6\tau_F^2\delta + 2\delta^2\tau_F)(2\tau_F+\delta) - 2(\tau_F^2+\delta\tau_F^2)}{(2\tau_F+\delta)^2}\frac{(\tau_F+\delta)e^{-r_{\delta F}} - \tau_F e^{-r_F}}{\delta} \\ + \frac{e^{-r_{\delta F}}(1+r_{\delta F}) - e^{-r_F}(1+r_F)}{\delta}\frac{\sigma_a(\tau_F+\delta)^2\tau_F^2}{2(2\tau_F+\delta)}.\end{aligned} \tag{C.49}$$

OUF model implemented in GPy

In this subsection, we simulate noiseless OUF data, and then fit a GP model with the OUF kernel. In Figures C.1 we plot the optimised model for a different number of time points (decreasing number of points from left to right, top to bottom). As expected, the uncertainty around the points is very small in Figure C.1 top left plot and increases in the Figure C.1 top right plot around the data points. Moreover, in Figures C.1 the uncertainty increases significantly in regions where there is no data. This is exactly the behaviour that we expected if our kernel was well defined.

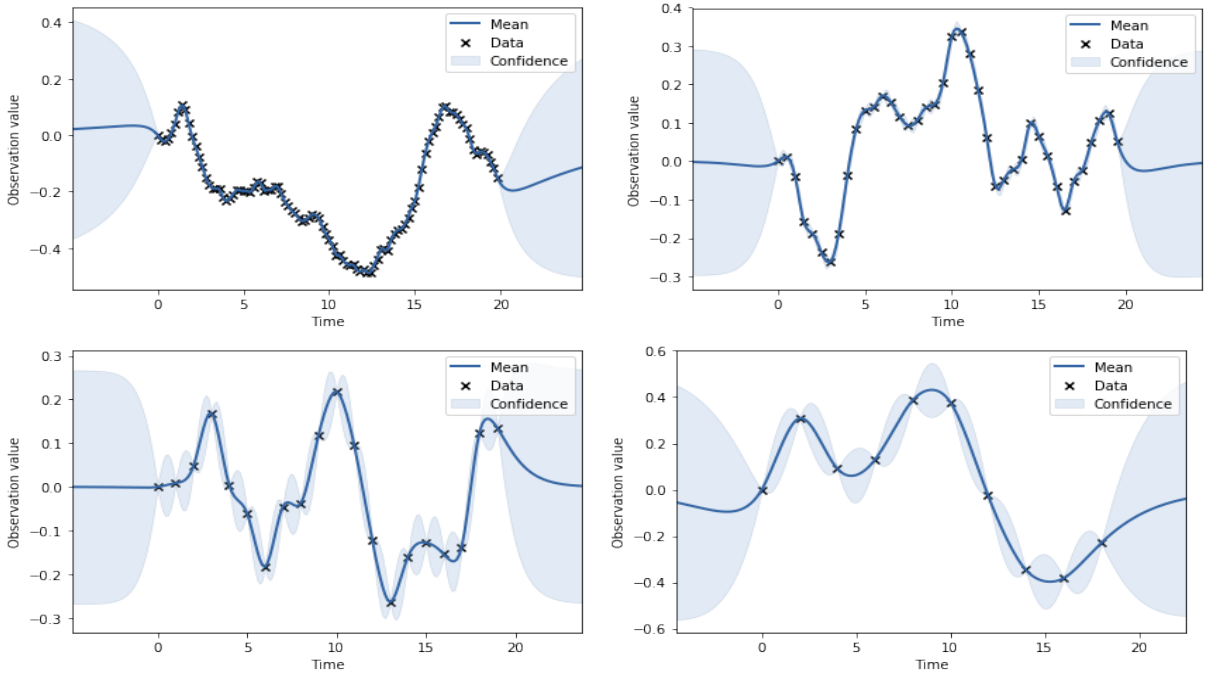


Figure C.1: Optimised GP model with the OUF kernel with varying number of datapoints.

Appendix D

Appendix section for Chapter 5

In this Appendix section we derive the derivatives formulas when the model is a hierarchical non-stationary GP with the kernels on the first-layer being non-stationary RBF kernel, respectively non-stationary Matérn 1/2. We also illustrate the mistakes in the implementation of the RBF non-stationary model in Heinonen et al. [2016] and derive the Matérn 1/2 non-stationary kernel formula.

D.1 Non-stationary Gaussian process model with a RBF kernel

Suppose $\mathbf{y}$ is a $n \times 2$ vector of observations over $\mathbf{x}$ inputs of size $n \times 1$. We assume an additive regression model,

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \omega^2 \mathbf{I}). \tag{D.1}$$

We then place a zero mean GP prior on the latent function $f(\mathbf{x})$,

$$f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k_f(\mathbf{x}, \mathbf{x}')), \tag{D.2}$$

where $k_f(\mathbf{x}, \mathbf{x}')$ is the kernel of the function f evaluated at $\mathbf{x}$ and $\mathbf{x}'$.

The RBF non-stationary formula [Heinonen et al., 2016] for the covariance function evaluated at points $\mathbf{x}$ and $\mathbf{x}'$ is

$$k_f(\mathbf{x}, \mathbf{x}') = \sigma(\mathbf{x})\sigma(\mathbf{x}') \sqrt{\frac{2l(\mathbf{x})l(\mathbf{x}')}{l(\mathbf{x})^2 + l(\mathbf{x}')^2}} \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^2}{l(\mathbf{x})^2 + l(\mathbf{x}')^2} \right), \tag{D.3}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ and $\sigma^2(\mathbf{x})$ and $l(\mathbf{x})$ are input-dependent amplitude (or signal variance) and lengthscale functions, respectively. We also put separate GP priors on the lengthscale and signal variance functions. In order to ensure positivity of these functions, we set the priors on the

logarithms

$$\log(\mathbf{l}(x)) \equiv \tilde{\mathbf{l}}(x) \sim \mathcal{GP}(0, k_l(\mathbf{x}, \mathbf{x}')). \tag{D.4}$$

$$\log\left(\boldsymbol{\sigma}^2(x)\right) \equiv \tilde{\boldsymbol{\sigma}}^2(x) \sim \mathcal{GP}(\mathbf{0}, k_{\sigma^2}(\mathbf{x}, \mathbf{x}')). \tag{D.5}$$

The chosen kernel for these two functions is a RBF kernel for each

$$k_l(\mathbf{x}, \mathbf{x}') = \alpha_l^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\beta_l^2}\right). \tag{D.6}$$

$$k_{\sigma^2}(\mathbf{x}, \mathbf{x}') = \alpha_{\sigma^2}^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\beta_{\sigma^2}^2}\right). \tag{D.7}$$

D.1.1 Derivatives of the parameters for the RBF non-stationary Gaussian process model

In order to optimise this model or perform HMC sampling we need the derivatives of the log likelihood with respect to all the parameters. Let $\mathbf{K}_f$, $\mathbf{K}_l$ and $\mathbf{K}_{\sigma^2}$ be the covariance matrices given by the kernels k_f, k_l, respectively k_{σ^2} evaluated at the datapoints $\mathbf{x}$.

The likelihood of the model using Bayes rule is

$$\mathcal{L} = p(\mathbf{y}|\tilde{\mathbf{l}}, \tilde{\boldsymbol{\sigma}}^2, \omega) p(\tilde{\mathbf{l}}, \tilde{\boldsymbol{\sigma}}^2). \tag{D.8}$$

We can calculate these quantities by using the fact

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_y),\ \mathbf{K}_y = \mathbf{K}_f + \boldsymbol{\Omega},\ \boldsymbol{\Omega} = \text{diag}(\omega^2). \tag{D.9}$$

$$\tilde{\mathbf{l}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_l). \tag{D.10}$$

$$\tilde{\boldsymbol{\sigma}}^2 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\sigma^2}). \tag{D.11}$$

Derivative of the lengthscale parameter $\tilde{l}_i$

Using the standard formula for the derivative of log likelihood with respect to a parameter [Rasmussen and Williams, 2006], Equation 5.9, page 114, we calculate the derivative of the lengthscale parameter $\tilde{l}_i$ i.e. the derivative of the lengthscale vector $\tilde{\mathbf{l}}$ at input i such that we have

$$\frac{\partial \log \mathcal{L}}{\partial \tilde{l}_i} = \frac{1}{2}\text{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}\right)\frac{\partial \mathbf{K}_y}{\partial \tilde{l}_i}\right) - [\mathbf{K}_l^{-1}\tilde{\mathbf{l}}]_i, \tag{D.12}$$

where $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y}$.

The first term of Equation D.12 is derived from differentiating $\log p(\mathbf{y}|\tilde{\mathbf{l}}, \tilde{\boldsymbol{\sigma}}^2, \omega)$ with respect to $\tilde{\mathbf{l}}$ conforming to Rasmussen and Williams [2006], equation 5.9, page 114. The second term of Equation D.12 comes from differentiating $\log p(\tilde{\mathbf{l}}|\mathbf{0}, \mathbf{K}_l)$ with respect to $\tilde{\mathbf{l}}$. We prove this below.

We have $\tilde{\mathbf{l}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_l)$. Therefore,

$$\log p(\tilde{\mathbf{l}}|\mathbf{0}, \mathbf{K}_l) = -\frac{1}{2}\tilde{\mathbf{l}}^t \mathbf{K}_l^{-1}\tilde{\mathbf{l}} - \frac{1}{2}\log|\mathbf{K}_l| - \frac{n}{2}\log(2\pi). \tag{D.13}$$

Differentiating the previous term with respect to $\tilde{\mathbf{l}}$ we get that $\frac{d\log p(\tilde{\mathbf{l}}|0,\mathbf{K}_l)}{d\tilde{\mathbf{l}}} = -\mathbf{K}_l^{-1}\tilde{\mathbf{l}}$, since the other terms are independent of $\tilde{\mathbf{l}}$. We now calculate the inner matrix $\frac{\partial \mathbf{K}_y}{\partial \tilde{l}_i}$ from Equation D.12. We have $\mathbf{K}_y = \mathbf{K}_f + \mathbf{\Omega}$, where $\mathbf{\Omega}$ is a diagonal matrix formed of the noise variance parameter ω^2, i.e. $\mathbf{\Omega} = \text{diag}(\omega^2)$. Therefore, we get that

$$\frac{\partial [\mathbf{K}_y]_{ij}}{\partial \tilde{l}_i} = \frac{\partial [\mathbf{K}_f]_{ij}}{\partial \tilde{l}_i} = \sigma_i \sigma_j \frac{\partial [\mathbf{T}_l]_{ij}}{\partial \log l_i}, \text{ where } [\mathbf{T}_l]_{ij} = \sqrt{\frac{2l(x_i)l(x_j)}{l(x_i)^2 + l(x_j)^2}} \exp\left(-\frac{(x_i - x_j)^2}{l(x_i)^2 + l(x_j)^2}\right). \tag{D.14}$$

For simplification, we denote $l(x_i) = l_i$ and $(x_i - x_j)^2 = d_{ij}$. We start calculating the partial derivative by first taking the diagonal case $i = j$ and we obtain

$$\frac{\partial [\mathbf{T}_l]_{ii}}{\partial \tilde{l}_i} = \frac{\partial [\mathbf{T}_l]_{ii}}{\partial \log l_i} = \frac{\partial \left(\sqrt{\frac{2l_i^2}{2l_i^2}} \exp\left(-\frac{d_{ij}}{2l_i^2}\right)\right)}{\partial \log l_i} = 0, \text{ since } d_{ij} = 0. \tag{D.15}$$

In the second case, we assume $i \neq j$. We have that

$$\frac{\partial [\mathbf{T}_l]_{ij}}{\partial \tilde{l}_i} = \frac{\partial [\mathbf{T}_l]_{ij}}{\partial \log l_i} = \frac{\partial \left(\sqrt{\frac{2l_i l_j}{l_i^2 + l_j^2}} \exp\left(-\frac{d_{ij}}{l_i^2 + l_j^2}\right)\right)}{\partial \log l_i}. \tag{D.16}$$

Using the chain rule we get that

$$\frac{\partial [\mathbf{T}_l]_{ij}}{\partial \log l_i} = \frac{\partial [\mathbf{T}_l]_{ij}}{\partial l_i} \frac{\partial l_i}{\partial \log l_i} = \frac{\partial [\mathbf{T}_l]_{ij}}{\partial l_i} l_i. \tag{D.17}$$

We calculate

$$
\begin{aligned}
\frac{\partial [\mathbf{T}_l]_{ij}}{\partial l_i} &= \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{\partial\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}}{\partial l_i} + \sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\frac{\partial \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)}{\partial l_i} \\
&= \frac{1}{2}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\left(\frac{2l_il_j}{{l_i}^2+{l_j}^2}\right)^{-\frac{1}{2}}\frac{2l_j({l_i}^2+{l_j}^2)-2l_i2l_il_j}{({l_i}^2+{l_j}^2)^2} \\
&+ \sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)(-d_{ij})\frac{(-2l_i)}{({l_i}^2+{l_j}^2)^2} \\
&= \frac{1}{2}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}}\frac{2l_j({l_i}^2+{l_j}^2-2l_i^2)}{({l_i}^2+{l_j}^2)^2} \\
&+ \sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{(2d_{ij}l_i)}{({l_i}^2+{l_j}^2)^2} \\
&= \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}}\frac{l_j({l_j}^2-{l_i}^2)}{({l_i}^2+{l_j}^2)^2} \\
&+ \sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{(2d_{ij}l_i)}{({l_i}^2+{l_j}^2)^2} \\
&= \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{({l_i}^2+{l_j}^2)^2}\left(\frac{1}{\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}}l_j({l_j}^2-{l_i}^2)+\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}2d_{ij}l_i)\right) \\
&= \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{({l_i}^2+{l_j}^2)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(\frac{{l_i}^2+{l_j}^2}{2l_il_j}l_j({l_j}^2-{l_i}^2)+2d_{ij}l_i\right) \\
&= \frac{1}{2}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{({l_i}^2+{l_j}^2)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(\frac{({l_i}^2+{l_j}^2)({l_j}^2-{l_i}^2)}{l_i}+4d_{ij}l_i\right) \\
&= \frac{1}{2}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{({l_i}^2+{l_j}^2)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(\frac{4d_{ij}{l_i}^2-{l_i}^4+{l_j}^4}{l_i}\right).
\end{aligned}
\tag{D.18}
$$

As shown in Equation D.17 we multiply the last equation by l_i to get to the final form

$$
\frac{\partial [\mathbf{T}_l]_{ij}}{\partial \log l_i} = \frac{1}{2}\exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{({l_i}^2+{l_j}^2)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(4d_{ij}{l_i}^2-{l_i}^4+{l_j}^4\right). \tag{D.19}
$$

Finally, multiplying the previous equation by $\sigma_i\sigma_j$ we get that

$$\frac{\partial[\mathbf{K}_y]_{ij}}{\partial\tilde{l}_i} = \frac{1}{2}\sigma_i\sigma_j \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{\left({l_i}^2+{l_j}^2\right)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(4d_{ij}{l_i}^2-{l_i}^4+{l_j}^4\right). \tag{D.20}$$

In Heinonen et al. [2016] the lengthscale derivative for the RBF non-stationary kernel shown in the supplemental material is the following

$$\frac{\partial[\mathbf{K}_y]_{ij}}{\partial\tilde{l}_i} = \frac{S_{ij}E_{ij}}{R_{ij}L_{ij}^3}l_il_j\left(4d_{ij}l_i^2-l_i^4+l_j^4\right), \tag{D.21}$$

where $d_{ij}=(x_i-x_j)^2$, and x is the input data. Also, $S_{ij}=\sigma_i\sigma_j$, $R_{ij}=\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}}$, $E_{ij}=\exp\left(\frac{-d_{ij}}{l_i^2+l_j^2}\right)$ and $L_{ij}=l_i^2+l_j^2$. We analyse the differences in Equations D.20 and D.21, and prove that these two equations are equivalent. Taking only the terms that are not common between these two equations we have (starting from Equation D.21)

$$\begin{aligned}\frac{1}{\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}}}\frac{1}{\left(l_i^2+l_j^2\right)^3}l_il_j &= \frac{1}{\left(l_i^2+l_j^2\right)^2}\frac{1}{\left(l_i^2+l_j^2\right)}\frac{\sqrt{l_i^2+l_j^2}}{\sqrt{2l_il_j}}l_il_j \\ &= \frac{1}{\left(l_i^2+l_j^2\right)^2}\frac{1}{\sqrt{l_i^2+l_j^2}}\sqrt{2l_il_j}\frac{1}{2} \\ &= \frac{1}{2}\frac{1}{\left(l_i^2+l_j^2\right)^2}\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}}.\end{aligned} \tag{D.22}$$

Therefore, Equations D.20 and D.21 are equivalent.

We are interested in calculating the inner matrix $\frac{\partial\mathbf{K}_y}{\partial\tilde{l}_i}$ from Equation D.12. To calculate this we make use of Equation D.20 reproduced again here for clarity

$$\frac{\partial[\mathbf{K}_y]_{ij}}{\partial\tilde{l}_i} = \frac{1}{2}\sigma_i\sigma_j \exp\left(-\frac{d_{ij}}{{l_i}^2+{l_j}^2}\right)\frac{1}{\left({l_i}^2+{l_j}^2\right)^2}\sqrt{\frac{2l_il_j}{{l_i}^2+{l_j}^2}}\left(4d_{ij}{l_i}^2-{l_i}^4+{l_j}^4\right). \tag{D.23}$$

$$\frac{\partial\log\mathscr{L}}{\partial\tilde{l}_i} = \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T-\mathbf{K}_y^{-1}\right)\frac{\partial\mathbf{K}_y}{\partial\tilde{l}_i}\right)-[\mathbf{K}_l^{-1}\tilde{\mathbf{l}}]_i. \tag{D.24}$$

To calculate the inner matrix $\frac{\partial\mathbf{K}_y}{\partial\tilde{l}_i}$ we form a sparse matrix from the matrix calculated from $\frac{\partial[\mathbf{K}_y]_{ij}}{\partial\tilde{l}_i}$ for all i's and j's. We call the latter matrix **dK**. This matrix will contain all the relevant information to calculate the partial derivative of $\mathbf{K}_y$ with respect to all $\tilde{l}_i$'s. We compute the sparse matrix by using the i-th row of dK. The sparse matrix will be symmetric given that the i-th row and i-th column are equal to the i-th row of **dK**. The inner matrix will be a 'plus' matrix where only the i-th row and i-th column are different from 0. For further clarification

of why that is we take a simple example. Suppose we have x_1, x_2 two input points. Therefore, we have two corresponding lengthscale parameters l_1, l_2 and two corresponding signal variance parameters σ_1^2 and σ_2^2. Given that $\mathbf{K}_f$ is a proper covariance matrix, then $\mathbf{K}_f$ is a symmetric matrix, thus $\mathbf{K}_y$ is also a symmetric matrix. We calculate the derivative of $\mathbf{K}_y$ with respect to $\tilde{l}_1$, where $\tilde{l}_1 = \log(l_1)$ and we obtain

$$\begin{aligned}\frac{\partial \mathbf{K}_y}{\partial \tilde{l}_1} &= \begin{pmatrix} \frac{\partial \mathbf{K}_y(x_1,x_1)}{\partial \tilde{l}_1} & \frac{\partial \mathbf{K}_y(x_1,x_2)}{\partial \tilde{l}_1} \\ \frac{\partial K_y(x_2,x_1)}{\partial \tilde{l}_1} & \frac{\partial \mathbf{K}_y(x_2,x_2)}{\partial \tilde{l}_1} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{K}_y(x_1,x_1)}{\partial \tilde{l}_1} & \frac{\partial \mathbf{K}_y(x_1,x_2)}{\partial \tilde{l}_1} \\ \frac{\partial \mathbf{K}_y(x_2,x_1)}{\partial \tilde{l}_1} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial \mathbf{K}_y(x_1,x_1)}{\partial \tilde{l}_1} & \frac{\partial \mathbf{K}_y(x_1,x_2)}{\partial \tilde{l}_1} \\ \frac{\partial \mathbf{K}_y(x_1,x_2)}{\partial \tilde{l}_1} & 0 \end{pmatrix}.\end{aligned} \tag{D.25}$$

given that $\mathbf{K}_y$ is a symmetric matrix, therefore $\frac{\partial \mathbf{K}_y(x_1,x_2)}{\partial \tilde{l}_1} = \frac{\partial \mathbf{K}_y(x_2,x_1)}{\partial \tilde{l}_1}$. The last entry in the matrix is zero, because we do not have a l_1 in the formula of $\mathbf{K}_y(x_2,x_2)$, therefore the partial derivative of it with respect to l_1 is zero.

It remains to explain how to get the quantities present inside the matrix. We will make use of Equation D.20. For this example, we calculate the matrix given by $\frac{\partial \mathbf{K}_y}{\partial \tilde{l}_i}$ for all i's and j's i.e. the matrix **dK**

$$\mathbf{dK} = \begin{pmatrix} \frac{\partial [\mathbf{K}_y]_{11}}{\partial \tilde{l}_1} & \frac{\partial [\mathbf{K}_y]_{12}}{\partial \tilde{l}_1} \\ \frac{\partial [\mathbf{K}_y]_{21}}{\partial \tilde{l}_2} & \frac{\partial [\mathbf{K}_y]_{22}}{\partial \tilde{l}_2} \end{pmatrix}. \tag{D.26}$$

Using Equations D.15 and D.20

$$\frac{\partial [\mathbf{K}_y]_{11}}{\partial \tilde{l}_1} = 0. \tag{D.27}$$

$$\frac{\partial [\mathbf{K}_y]_{22}}{\partial \tilde{l}_2} = 0. \tag{D.28}$$

Also,

$$\frac{\partial [\mathbf{K}_y]_{12}}{\partial \tilde{l}_1} = \frac{1}{2}\sigma_1\sigma_2 \exp\left(-\frac{(x_1-x_2)^2}{{l_1}^2+{l_2}^2}\right) \frac{1}{({l_1}^2+{l_2}^2)^2} \sqrt{\frac{2l_1l_2}{{l_1}^2+{l_2}^2}} \left(4(x_1-x_2)^2{l_1}^2 - {l_1}^4 + {l_2}^4\right). \tag{D.29}$$

$$\frac{\partial [\mathbf{K}_y]_{21}}{\partial \tilde{l}_2} = \frac{1}{2}\sigma_1\sigma_2 \exp\left(-\frac{(x_1-x_2)^2}{{l_1}^2+{l_2}^2}\right) \frac{1}{({l_1}^2+{l_2}^2)^2} \sqrt{\frac{2l_1l_2}{{l_1}^2+{l_2}^2}} \left(4(x_1-x_2)^2{l_2}^2 - {l_2}^4 + {l_1}^4\right). \tag{D.30}$$

Is important to notice that the order of the indices is given by l_i's not by x_i's, given that $\mathbf{K}_f$ is a symmetric covariance matrix. However, the matrix generated by Equation D.20 is not symmet-

ric, given the last term: $\left(4d_{ij}l_i^2 - l_i^4 + l_j^4\right)$ is not symmetric for all i's and j's in this equation. Therefore, once we compute the matrix of derivatives **dK** using Equation D.20 for all i's and j's, we only need to select the i-th row of this matrix, given that we need the derivative of $\mathbf{K}_y$ with respect to l_i. We also use the fact that the inner matrix is symmetric.

Signal variance derivative

We proceed similarly as in the lengthscale derivative, the log likelihood derivative with respect to the signal variance parameter is the following

$$\frac{\partial \log \mathscr{L}}{\partial \tilde{\sigma}_i^2} = \frac{1}{2}\text{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}\right)\frac{\partial \mathbf{K}_y}{\partial \tilde{\sigma}_i^2}\right) - [\mathbf{K}_{\boldsymbol{\sigma}^2}^{-1}\tilde{\boldsymbol{\sigma}}^2]_i, \tag{D.31}$$

where $\boldsymbol{\alpha} = \mathbf{K}_y{}^{-1}\mathbf{y}$. We are interested in calculating the inner matrix $\frac{\partial \mathbf{K}_y}{\partial \tilde{\sigma}_i^2}$. We have $\mathbf{K}_y = \mathbf{K}_f + \boldsymbol{\Omega}$, where $\boldsymbol{\Omega} = \text{diag}(w^2)$, and hence $\frac{\partial \mathbf{K}_y}{\partial \tilde{\sigma}_i^2} = \frac{\partial \mathbf{K}_f}{\partial \tilde{\sigma}_i^2}$. Firstly, we consider the case when $i \neq j$. We denote $[\mathbf{L}_f]_{ij} = \sqrt{\frac{2l_i l_j}{l_i^2+l_j^2}}\exp\left(-\frac{(x_i-x_j)^2}{l_i^2+l_j^2}\right)$ for simplification and using Equation D.3 we have that

$$\begin{aligned}
\frac{\partial [\mathbf{K}_f]_{ij}}{\partial \log \sigma_i^2} &= [\mathbf{L}_f]_{ij}\frac{\partial \sigma_i \sigma_j}{\partial \log \sigma_i^2} = \frac{1}{2}[\mathbf{L}_f]_{ij}\sigma_j \frac{\partial \sigma_i}{\partial \log \sigma_i} \\
&= \frac{1}{2}[\mathbf{L}_f]_{ij}\sigma_j \times 1/\frac{\partial \log \sigma_i}{\partial \sigma_i} \\
&= \frac{1}{2}[\mathbf{L}_f]_{ij}\sigma_j \times 1/\frac{1}{\sigma_i} \\
&= \frac{1}{2}[\mathbf{K}_f]_{ij}.
\end{aligned} \tag{D.32}$$

Secondly, we consider the case when we are on the diagonal i.e. $i = j$ and we obtain

$$\begin{aligned}
\frac{\partial [\mathbf{K}_f]_{ii}}{\partial \log \sigma_i^2} &= \frac{\partial \sigma_i^2 [\mathbf{L}_f]_{ii}}{\partial \log \sigma_i^2} = [\mathbf{L}_f]_{ii} \times 1/\frac{\partial \log \sigma_i^2}{\partial \sigma_i^2} \\
&= [\mathbf{L}_f]_{ii} \times 1/\frac{1}{\sigma_i^2} = \sigma_i^2 [\mathbf{L}_f]_{ii} = [\mathbf{K}_f]_{ii}.
\end{aligned} \tag{D.33}$$

In conclusion we have,

$$\frac{\partial [\mathbf{K}_f]_{ij}}{\partial \log \sigma_i^2} = \begin{cases} [\mathbf{K}_f]_{ij}, & \text{for } i = j. \\ \frac{1}{2}[\mathbf{K}_f]_{ij}, & \text{for } x_i, x_j,\ i \neq j. \\ 0, & \text{for } x_k,\ k \neq i. \end{cases} \tag{D.34}$$

We now try to find a more convenient formula to use when trying to compute the signal variance derivative. We start by denoting $\mathbf{M} = \boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}$ and we want to prove that $\mathbf{M}$ is symmetric. We use the fact that $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$ ($\mathbf{A}$, $\mathbf{B}$ random square matrices), and that $\mathbf{K}_y$ is

symmetric, therefore $\mathbf{K}_y = \mathbf{K}_y^T$. We obtain

$$\begin{aligned}\mathbf{M}^T &= (\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1})^T \\ &= (\boldsymbol{\alpha}\boldsymbol{\alpha}^T)^T - (\mathbf{K}_y^{-1})^T \\ &= (\boldsymbol{\alpha}^T)^T\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1} \\ &= \boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1} \\ &= \mathbf{M}.\end{aligned} \tag{D.35}$$

For simplicity we assume that $i = 1$. Making use of Equation D.34 we have that

$$\begin{aligned}\frac{\partial \mathbf{K}_f}{\partial \log \sigma_1^2} &= \begin{pmatrix} \frac{\partial \mathbf{K}_f(x_1,x_1)}{\partial \log \sigma_1^2} & \cdots & \cdots & \frac{\partial K_f(x_1,x_n)}{\partial \log \sigma_1^2} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \mathbf{K}_f(x_n,x_1)}{\partial \log \sigma_1^2} & \cdots & \cdots & 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{K}_f(x_1,\, x_1) & \cdots & \cdots & \frac{1}{2}\mathbf{K}_f(x_1,x_n) \\ \vdots & \ddots & \ddots & \vdots \\ \frac{1}{2}\mathbf{K}_f(x_n,x_1) & \cdots & \cdots & 0 \end{pmatrix}.\end{aligned} \tag{D.36}$$

Let

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \cdots & \cdots & m_{1n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & \cdots & m_{nn} \end{pmatrix}. \tag{D.37}$$

We calculate

$$\begin{aligned}\operatorname{tr}\left(\mathbf{M}\frac{\partial \mathbf{K}_f}{\partial \log \sigma_1^2}\right) &= m_{11}\mathbf{K}_f(x_1,x_1) + \frac{1}{2}\mathbf{K}_f(x_2,x_1) + \cdots + \frac{1}{2}m_{1n}\mathbf{K}_f(x_n,x_1) \\ &\quad + \frac{1}{2}m_{21}\mathbf{K}_f(x_1,x_2) + \cdots + \frac{1}{2}m_{n1}\mathbf{K}_f(x_1,x_n) \\ &= m_{11}\mathbf{K}_f(x_1,x_1) + m_{12}\mathbf{K}_f(x_1,x_2) + \cdots + m_{1n}\mathbf{K}_f(x_1,x_n) \\ &= \operatorname{diag}(\mathbf{M}\mathbf{K}_f).\end{aligned} \tag{D.38}$$

In the previous equation we made use of the symmetry of the matrices K_f and M. Similarly, for other i's we get

$$\operatorname{tr}\left(\mathbf{M}\frac{\partial \mathbf{K}_f}{\partial \log \sigma_i^2}\right) = \operatorname{diag}(\mathbf{M}\mathbf{K}_f). \tag{D.39}$$

Thus, now we can make use of the previous relation in Equation D.31.

Derivative of the log likelihood with respect to the noise variance ω^2

Since we do not have a GP prior on ω^2, the derivative of the log likelihood with respect to ω^2 has a simplified version. As in the previous cases we get

$$\frac{\partial \log \mathscr{L}}{\partial \omega^2} = \frac{1}{2}\text{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1})\frac{\partial \mathbf{K}_y}{\partial \omega^2}\right). \tag{D.40}$$

We have that $\mathbf{K}_y = \mathbf{K}_f + \boldsymbol{\Omega}$, where $\boldsymbol{\Omega} = \text{diag}(\omega^2)$. Therefore, we get $\frac{\partial \mathbf{K}_y}{\partial \omega^2} = \frac{\partial \boldsymbol{\Omega}}{\partial \omega^2} = \mathbf{I}$. We can further simplify Equation D.40 to obtain the following

$$\frac{\partial \log \mathscr{L}}{\partial \omega^2} = \frac{1}{2}\text{tr}(\mathbf{M}), \text{where } \mathbf{M} = (\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}). \tag{D.41}$$

Derivatives of the hyperparameters

The kernels of choice on the lower level layer of the non-stationary GP are RBF kernels

$$k_l(\mathbf{x}, \mathbf{x}') = \alpha_l^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\beta_l^2}\right). \tag{D.42}$$

$$k_{\sigma^2}(\mathbf{x}, \mathbf{x}') = \alpha_{\sigma^2}^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\beta_{\sigma^2}^2}\right). \tag{D.43}$$

In general, for k a RBF kernel, we have that

$$\frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \alpha^2} = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\beta^2}\right) = \frac{k(\mathbf{x}, \mathbf{x}')}{\alpha^2}. \tag{D.44}$$

$$\frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \beta} = k(\mathbf{x}, \mathbf{x}')\frac{-||\mathbf{x} - \mathbf{x}'||^2}{2}(-2)\beta^{-3} = k(\mathbf{x}, \mathbf{x}')||\mathbf{x} - \mathbf{x}'||^2\beta^{-3}. \tag{D.45}$$

Let $\mathbf{K}_l$ and $\mathbf{K}_{\sigma^2}$ be the covariance matrices given by the kernels k_l, respectively k_{σ^2} evaluated at the datapoints $\mathbf{x}$. As in previous cases, to calculate the derivative of the log likelihood with respect to the hyperparameters we make use of the formulas found in Rasmussen and Williams [2006] and we obtain

$$\frac{\partial \log \mathscr{L}}{\partial \alpha_l^2} = \frac{1}{2}\text{tr}\left((\mathbf{a}_l\mathbf{a}_l^T - \mathbf{K}_l^{-1})\frac{\partial \mathbf{K}_l}{\partial \alpha_l^2}\right). \tag{D.46}$$

$$\frac{\partial \log \mathscr{L}}{\partial \alpha_{\sigma^2}^2} = \frac{1}{2}\text{tr}\left((\mathbf{a}_{\text{var}}\mathbf{a}_{\text{var}}^T - \mathbf{K}_{\sigma^2}^{-1})\frac{\partial \mathbf{K}_{\sigma^2}}{\partial \alpha_{\sigma^2}^2}\right), \tag{D.47}$$

where $\mathbf{a}_l = \mathbf{K}_l^{-1}\tilde{\mathbf{I}}$ and $\mathbf{a}_{\text{var}} = \mathbf{K}_{\text{var}}^{-1}\tilde{\boldsymbol{\sigma}}^2$. Similarly, we obtain

$$\frac{\partial \log \mathscr{L}}{\partial \beta_l} = \frac{1}{2}\text{tr}\left((\mathbf{a}_l\mathbf{a}_l^T - \mathbf{K}_l^{-1})\frac{\partial \mathbf{K}_l}{\partial \beta_l}\right). \tag{D.48}$$

$$\frac{\partial \log \mathscr{L}}{\partial \beta_{\sigma^2}} = \frac{1}{2}\text{tr}\left((\mathbf{a}_{\text{var}}\mathbf{a}_{\text{var}}^T - \mathbf{K}_{\sigma^2}^{-1})\frac{\partial \mathbf{K}_{\sigma^2}}{\partial \beta_{\sigma^2}}\right). \tag{D.49}$$

D.2 Non-stationary Gaussian process model with a Matérn 1/2 kernel

The Matérn 1/2 non-stationary covariance function [Paciorek and Schervish, 2004] evaluated at the points x_i and x_j is[1]

$$k_f(x_i, x_j) = \sigma_i\sigma_j\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}}\exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right), \tag{D.50}$$

where $d_{ij} = (x_i - x_j)^2$, σ_i, l_i are the signal variance, respectively lengthscale parameters at the input point x_i, for every i. As before, to optimise the model, or perform MCMC sampling, we need the derivative of the log likelihood of the model with respect to all the parameters. The only derivative that is different will be the lengthscale derivative. The other derivatives have the same formula as for the RBF non-stationary kernel.

Derivative of the log likelihood with respect to the lengthscale parameter

We have that

$$\begin{aligned}\frac{\partial [\mathbf{L}_f]_{ij}}{\partial l_i} = &\frac{1}{2}\exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right)2l_j\left(\frac{l_i}{l_i^2+l_j^2}\right)'\left(\frac{2l_il_j}{l_i^2+l_j^2}\right)^{-\frac{1}{2}} \\ &+\exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right)\frac{-1}{2}\left(\frac{2d_{ij}}{l_i^2+l_j^2}\right)^{-\frac{1}{2}}\left(\frac{2d_{ij}}{l_i^2+l_j^2}\right)'\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}},\end{aligned} \tag{D.51}$$

[1] We offer a derivation of this formula in the Appendix, Section D.4.

where $[\mathbf{L}_f]_{ij} = \sqrt{\frac{2l_i l_j}{l_i^2+l_j^2}}\exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right)$. For simplicity, we denote $E = \exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right)$, $R = \sqrt{\frac{2l_i l_j}{l_i^2+l_j^2}}$. Therefore, we have

$$\begin{aligned}\frac{\partial[\mathbf{L}_f]_{ij}}{\partial l_i} &= El_j\frac{l_i^2+l_j^2-2l_i^2}{(l_i^2+l_j^2)^2}\frac{1}{R} + RE\frac{-1}{2}\sqrt{\frac{l_i^2+l_j^2}{2d_{ij}}}2d_{ij}\frac{-2l_i}{(l_i^2+l_j^2)^2} \\ &= El_j\frac{l_j^2-l_i^2}{(l_i^2+l_j^2)^2}\frac{1}{R} + ER\sqrt{l_j^2+l_i^2}\sqrt{2d_{ij}}\frac{l_i}{(l_i^2+l_j^2)^2} \\ &= E\frac{1}{(l_i^2+l_j^2)^2}\left(\frac{l_j(l_j^2-l_i^2)}{R} + 2l_i\sqrt{d_{ij}l_i l_j}\right).\end{aligned} \tag{D.52}$$

Thus, the derivative of the Matérn 1/2 non-stationary kernel with respect to $\tilde{l}_i$, where $\log(l_i) = \tilde{l}_i$ is

$$\frac{\partial[\mathbf{K}_y]_{ij}}{\partial\tilde{l}_i} = \sigma_i\sigma_j l_i\exp\left(-\sqrt{\frac{2d_{ij}}{l_i^2+l_j^2}}\right)\frac{1}{(l_j^2+l_i^2)^2}\left(\frac{l_j(l_j^2-l_i^2)}{\sqrt{\frac{2l_i l_j}{l_i^2+l_j^2}}} + 2l_i\sqrt{d_{ij}l_i l_j}\right). \tag{D.53}$$

As before we used the fact that

$$\frac{\partial[\mathbf{L}_f]_{ij}}{\partial\log l_i} = \frac{\partial[\mathbf{L}_f]_{ij}}{\partial l_i}\frac{\partial l_i}{\partial\log l_i} = \frac{\partial[\mathbf{L}_f]_{ij}}{\partial l_i}l_i. \tag{D.54}$$

Then, we proceed as in the previous cases to calculate the derivative of log likelihood of the model with respect to the other parameters and hyperparameters.

Posterior whitening

We represent the latent function values by centered (whitened) variables. If we denote the whitened variables $\mathbf{v}$, we have

$$\mathbf{v} \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{D.55}$$

$$\mathbf{f} = \mathbf{L}\mathbf{v} + \mathbf{m}, \text{ where } \mathbf{L}\mathbf{L}^T = \mathbf{K}, \text{ and } \mathbf{m} \text{ is a mean function.} \tag{D.56}$$

We perform whitening on the top layer of the non-stationary GP in order to reduce the correlation between variables so that the HMC sampling can be done more efficiently. In our case we have that

$$\mathring{\mathbf{l}} = \mathbf{L}_l^{-1}\tilde{\mathbf{l}},\ \mathbf{K}_l = \mathbf{L}_l\mathbf{L}_l^T. \tag{D.57}$$

$$\mathring{\boldsymbol{\sigma}}^2 = \mathbf{L}_{\sigma^2}^{-1}\tilde{\sigma}^2,\ \mathbf{K}_{\sigma^2} = \mathbf{L}_{\sigma^2}\mathbf{L}_{\sigma^2}^T. \tag{D.58}$$

We can calculate the derivatives of the log likelihood with respect to the whitened parameters by multiplying the derivatives by the transpose of the Choleski decomposition. For example, for the whitened lengthscale parameter we have

$$\frac{\partial \log \mathscr{L}}{\partial \mathring{\mathbf{l}}} = \frac{\partial \log \mathscr{L}}{\partial \tilde{\mathbf{l}}} \frac{\partial \tilde{\mathbf{l}}}{\partial \mathring{\mathbf{l}}} = \frac{\partial \log \mathscr{L}}{\partial \tilde{\mathbf{l}}} \frac{\partial \mathbf{L}_l \mathring{\mathbf{l}}}{\partial \mathring{\mathbf{l}}} = \mathbf{L}_l^T \frac{\partial \log \mathscr{L}}{\partial \tilde{\mathbf{l}}}. \tag{D.59}$$

In the previous equation, we made use of the chain rule and the fact that $\frac{\partial \mathbf{L}_l \mathring{\mathbf{l}}}{\partial \mathring{\mathbf{l}}} = \mathbf{L}_l$ ($\mathbf{L}_l$ is a constant with respect to $\mathring{\mathbf{l}}$). Since we want the derivatives as column vectors we transpose $\mathbf{L}_l$ and put it at the beginning of the equation.

Lengthscale derivative

Regarding the lengthscale derivative of the log likelihood with respect to the lengthscale parameter $\tilde{l}_i$ we reached the same formula for the lengthscale derivative as Heinonen et al. [2016]. The derivative of $\mathbf{K}_f$ with respect to $\tilde{l}_i$ is found in the Supplemental Material of Heinonen et al. [2016], however the formula has an important typo and is unclear about what d is. We corrected the formula using Heinonen et al. [2016]'s Matlab code and our own formula, which correspond i.e.

$$\frac{\partial [\mathbf{K}_y]_{ij}}{\partial \tilde{l}_i} = \frac{S_{ij} E_{ij}}{R_{ij} L_{ij}^3} l_i l_j \left(4 d_{ij} l_i^2 - l_i^4 + l_j^4\right), \tag{D.60}$$

where $d_{ij} = (x_i - x_j)^2$, and $\mathbf{x}$ is the input data. Also, $S_{ij} = \sigma_i \sigma_j$, $R_{ij} = \sqrt{\frac{2 l_i l_j}{l_i^2 + l_j^2}}$, $E_{ij} = \exp\left(\frac{-d_{ij}}{l_i^2 + l_j^2}\right)$ and $L_{ij} = l_i^2 + l_j^2$.

The derivative of the log likelihood with respect to $\tilde{\mathbf{l}}_\mathbf{i}$ is

$$\frac{\partial \log \mathscr{L}}{\partial \tilde{l}_i} = \frac{1}{2} \mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}\right) \frac{\partial \mathbf{K}_y}{\partial \tilde{l}_i}\right) - [\mathbf{K}_l^{-1}(\tilde{\mathbf{l}} - \boldsymbol{\mu}_l)]_i. \tag{D.61}$$

The derivative $\frac{\partial \mathbf{K}_y}{\partial \tilde{l}_i}$ is a 'plus' matrix where only the i-th column and row are non-zero. Given that $\mathbf{K}_y$ is a symmetric covariance matrix, its derivative with respect to $\tilde{l}_i$, for a certain i, is also a symmetric matrix.

D.3 Heinonen et al. [2016]'s implementation errors: Matlab code

In this section we highlight the mistakes in Heinonen et al. [2016]'s implementation in Matlab. In Figure D.1, the Matlab code will create the matrix **dK**, given by the formula in equation D.21. In Figure D.2, the Matlab code is trying to compute the derivative of the log likelihood with respect to the parameter $\tilde{l}_i$, shown in Equation D.61. In line 66, in Figure D.1, the first sum

corresponds to the trace term in Equation D.61. In the same equation the multiplication of the matrix $\mathbf{A} = \boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}$ with the sparse matrix, created by the 'sparse' command is given by the command 'sum' of the dot product between the matrix **A** and the sparse matrix. In general, the command 'sum(**C**,2)' returns a column vector containing the sum of each row, where **C** is a matrix. In any case, lines 66 and 70 should reproduce exactly Equation D.61, whether a dot product or matrix multiplication is used. The problem with Heinonen et al. [2016]'s code is line 66, highlighted part. Given that **dK** is not a symmetric matrix, selecting the i-th columns and the i-th rows will not result in a symmetric matrix, as it should. The correct code would be [**dK**(i,:) **dK**(i,:)']), resulting in a inner symmetric matrix.

In Figure D.3 we show the implementation of the derivative of the log likelihood with respect to σ_i parameter. This is the Matlab implementation of Equation 4, (second equation) on page 734 in Heinonen et al. [2016]. It has a factor of two in front of the first term. However, this is an error, as the formula in Equation 4 is correct, and this can be proven either by following the same procedure used here or by using Equations D.31 and D.39.

```
dK = zeros(n,n);

for i=1:n
        li = ell(i);
        for j=1:n
                lj = ell(j);
                lij = li^2 + lj^2;
                eij = exp(-pars.D(i,j)/lij);
                rij = sqrt( (2*li*lj)/(li^2 + lj^2) );
                sij = sigma(i)*sigma(j);
                dK(i,j) = li*lj * sij*eij*rij^(-1)*lij^(-3) * (4*pars.D(i,j)*li^2 - li^4 + lj^4);
        end
end
```

Figure D.1: Heinonen et al. [2016]'s Matlab code, creation of the matrix $\frac{\partial [\mathbf{K}_y]_{ij}}{\partial \tilde{l}_i}$ for all i's and j's i.e. the matrix **dK**.

```
53              dl_l = zeros(n,1);
54              for i=1:n
55  %                   ei = zeros(n,1);
56  %                   ei(i) = 1;
57  %                   Mi = bsxfun(@or, ei, ei');
58
59                      % original code:
60  %                   dl(i) = 0.5*sum(diag(A * (Mi .* dK) ));
61                      % optimised:
62                      % i) replace diag() with sum(,2) since we only need diagonal results
63                      % ii) make Mi.*dK sparse
64  %                   dl(i) = 0.5*sum( sum(A .* sparse(Mi .* dK)',2) );
65                      % iii) construct sparse matrix directly from the col/row
66                      dl_l(i) = 0.5*sum( sum(A .* sparse([1:n i*ones(1,n)], [i*ones(1,n) 1:n], [dK(:,i) dK(i,:)'])',2));
67              end
68
69
70              dl_l = dl_l - pars.Kl\(pars.l_ell - pars.l_muell);
```

Figure D.2: The derivative of the log likelihood with respect to $\tilde{l}_i$ [Heinonen et al., 2016].

```
function dwl_s = deriv_sigma(gp, scalar)
% derivative of the sigma latent function wrt MLL

        if ~exist('scalar','var')
                scalar = 0;
        end

        n = length(gp.xtr);

        if sum(ismember('ab', gp.nsfuncs))
                gp.Ks = gausskernel(gp.xtr,gp.xtr,gp.betasigma, gp.alphasigma, gp.tol);
        end

        Ky = nsgausskernel(gp.xtr, gp.xtr, gp.l_ell, gp.l_ell, gp.l_sigma, gp.l_sigma, gp.l_omega);
        Kf = nsgausskernel(gp.xtr, gp.xtr, gp.l_ell, gp.l_ell, gp.l_sigma, gp.l_sigma, log(0));

        a = Ky\gp.ytr;
        A = a*a' - inv(Ky);

        dl_s = 2*diag( A * Kf ) - gp.Ks\(gp.l_sigma - gp.l_musigma);

        if scalar
                dl_s = ones(n,1) * sum(dl_s);
        end

        if ismember('s', gp.nsfuncs)
                dwl_s = gp.Ls'*dl_s;
        else
                dwl_s = gp.Ls\dl_s;
        end
end
```

Figure D.3: The derivative of the log likelihood with respect to $\tilde{\sigma}_i$ [Heinonen et al., 2016].

D.4 Deriving the non-stationary Matérn 1/2 kernel formula

To derive the non-stationary Matérn 1/2 kernel formula we make use of the Equations 2 and 3 in Paciorek and Schervish [2004]. We have that

$$\begin{aligned}\mathrm{Cov}(x_i,x_j) &= \sqrt{l_i l_j}\left(\frac{l_i^2+l_j^2}{2}\right)^{-\frac{1}{2}}\sigma_i\sigma_j\exp\left(-\sqrt{Q_{ij}}\right)\\ &= \sigma_i\sigma_j\sqrt{\frac{2l_il_j}{l_i^2+l_j^2}}\exp\left(-\sqrt{\frac{2(x_i-x_j)^2}{l_i^2+l_j^2}}\right),\end{aligned} \tag{D.62}$$

where we used the fact that Σ_i in Paciorek and Schervish [2004] in one-dimension is equal to the square of the lengthscale at that time point i.e. $\Sigma_i = l_i^2$. Also, we have used the fact that the stationary Matérn 1/2 kernel formula is the following

$$k(t_i, t_j) = \exp\left(-\frac{|t_i - t_j|}{l}\right). \tag{D.63}$$

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. 2016. doi: 10.48550/ARXIV.1605.08695.

R. Anderson-Sprecher and J. Ledolter. State-space analysis of wildlife telemetry data. *Journal of The American Statistical Association*, 86:596–602, 1991.

T. Avgar, R. Deardon, and J.M. Fryxell. An empirically parameterized individual based model of animal movement, perception, and memory. *Ecological Modelling*, 251:158–172, 2013.

T. Avgar, J. R. Potts, M. A. Lewis, and M. S. Boyce. Integrated step selection analysis: bridging the gap between resource selection and animal movement. *Methods in Ecology and Evolution*, 7(5):619–630, 2016.

L. Bachelir. Théorie de la spéculation. *Annales scientifiques de l'É.N.S*, 3e(17):21–86, 1900.

S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical modeling and analysis for spatial data*, volume 101. Chapman & Hall/CRC Monographs on Statistical and Applied Probability;, 01 2004. doi: 10.1201/9780203487808.

K. Banner, K. Irvine, and T. Rodhouse. The use of Bayesian priors in ecology: The good, the bad, and the not great. *Methods in Ecology and Evolution*, 11, 05 2020. doi: 10.1111/2041-210X.13407.

D. Barber and Y. Wang. Gaussian processes for Bayesian estimation in ordinary differential equations. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1485–1493, Bejing, China, 22–24 Jun 2014. PMLR.

O. E. Barndorff-Nielsen and N. Shephard. Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):167–241, 2001a.

E. Batschelet. *Circular Statistics in Biology*. Mathematics in biology. Academic Press, London and New York, 1981. ISBN 9780120810505.

S. Benhamou. Detecting an orientation component in animal paths when the preferred direction is individual dependent. *Ecology*, 87:518–528, 2006.

J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, Germany, 2006. ISBN 0387310738.

K. Bjørneraas, B. Moorter, C. Rolandsen, and I. Herfindal. Screening global positioning system location data for errors using animal movement characteristics. *Journal of Wildlife Management*, 74:1361–1366, 08 2010. doi: 10.2193/2009-405.

P. G. Blackwell, M. Niu, M. S. Lambert, and S. D. LaPoint. Exact Bayesian inference for animal movement in continuous time. *Statistical Ecology*, 7:184–195, 2016.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

M. Blum and M. Riedmiller. Electricity demand forecasting using Gaussian processes. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, volume 10, pages 10–13, 01 2013.

E. V. Bonilla, K. Krauth, and A. Dezfouli. Generic inference in latent Gaussian process models. 2018. doi: 10.48550/ARXIV.1609.00577.

P. Bovet and S. Benhamou. Spatial analysis of animals movements using a correlated random walk model. *Journal of Theoretical Biology*, 131:419–433, 1988.

D. R. Brillinger. *Modeling spatial trajectories (From Handbook of Spatial Statistics by A. Gelfand, P. Diggle, M. Fuentes and P. Guttorp)*. CRC Press, Boca Raton, Florida, USA, 2010. ISBN 9781420072877.

D. R. Brillinger, H.K. Preisler, A. Ager, and J. G. Kie. *The use of potential functions in modeling*, page 369–386. Nova Science Publishers, Huntington, New York, USA., 2001. ISBN 978-1-4614-1344-8. doi: 10.1007/978-1-4614-1344-8_22.

D. R. Brillinger, H. K. Preisler, A. A. Ager, J. G. Kie, and B. S. Stewart. Employing stochastic differential equations to model wildlife motion. *Bulletin Brazilian Mathematical Society*, 33: 385–408, 2002.

D. R. Brillinger, H. K. Preisler, A. A. Ager, and J. G. Kie. An exploratory data analysis (EDA) of the paths of moving animals. *Journal of Statistical Planning and Inference*, 122(1-2):43–63, 2004.

S. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *J. Comput. Graphi. Stat.*, 7:434–455, 1998. doi: 10.1080/10618600.1998.10474787.

D. D. Brown, R. Kays, M. Wikelski, R. Wilson, and A. P. Klimley. Observing the unwatchable through acceleration logging of animal behavior. *Animal Biotelemetry*, 1(1):1–16, 2013.

R. Brown. A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and the general existence of active molecules in organic and inorganic bodies. *The Philosophical Magazine*, 4:161–173, 1828.

J. Brynjarsdóttir and A. O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30, 11 2014. doi: 10.1088/0266-5611/30/11/114007.

F. Cagnacci, L. Boitani, R. Powell, and M. Boyce. Animal ecology meets GPS-based radio telemetry: A perfect storm of opportunities and challenges. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 365:2157–62, 07 2010. doi: 10.1098/rstb.2010.0107.

N. Cain, A. K. Barreiro, M. Shadlen, and E. Shea-Brown. Neural integrators for decision making: A favorable trade-off between robustness and sensitivity. *Journal of Neurophysiology*, 109(10):2542–2559, 2013.

C. Calder. Dynamic factor process convolution models for multivariate space-time data with application to air quality assessment. *Environmental and Ecological Statistics*, 14:229–247, 2007.

M. Campbell, J. Ringrose, J. Boulanger, A. Roberto-Charron, K. Methuen, C. Mutch, T. Davison, and C. Wray. An aerial abundance estimate of the dolphin and union caribou (*Rangifer tarandus groenlandicus x pearyi*) herd, kitikmeot region, nunavut – fall 2020. *Government of Nunavut Department of Environment GN Technical Report Series*, (01-2021), 2021. URL `https://gov.nu.ca/sites/default/files/wildlife_-_20210301_du_fall_2020_caribou_survey_file_report_feb_1_2021_final.pdf`.

N. Campioni, D. Husmeier, J. Morales, J. Gaskell, and C. J. Torney. Inferring microscale properties of interacting systems from macroscale observations. *Physical Review Research*, 3: 043074, Oct 2021. doi: 10.1103/PhysRevResearch.3.043074.

A. D. Cobb, A. Markham, and S. J. Roberts. Learning from lions: inferring the utility of agents from their trajectories. 2017. doi: 10.48550/ARXIV.1709.02357.

E. A. Codling and N. A. Hill. Sampling rate effects on measurements of correlated and biased random walks. *Journal of Theoretical Biology*, 233:573–588, 2005.

E. A. Codling, N. A. Hill, J. W. Pitchford, and S. D. Simpson. Random walk models for the movement and recruitment of reef fish larvae. *Marine Ecology Progress Series*, 279:215–224, 2004.

E. A. Codling, M. J. Plank, and S. Benhamou. Random walk models in biology. *Journal of the Royal Society, Interface*, 5(25):813–814, 2008. doi: 10.1098/rsif.2008.0014.

N. Cressie and C. K. Wikle. *Space-time Kalman filter (From Encyclopedia of Environmetrics by A. H. El-Shaarawi and W. W. Piegorsch*. John Wiley and Sons, Ltd, Chichester, New York, 2002. ISBN 0471899976.

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215. Proceedings of Machine Learning Research (PMLR), 2013.

T. S. Doherty, G. C. Hays, and D. A. Driscoll. Human disturbance causes widespread disruption of animal movement. *Nature Ecology & Evolution*, 2021.

D. Dua and C. Graff. UCI machine learning repository. 2017. URL `http://archive.ics.uci.edu/ml`.

M. M. Dunlop, M. A. Girolami, A. M. Stuart, and A. L. Teckentrup. How deep are deep Gaussian processes? 2017. doi: 10.48550/ARXIV.1711.11280.

A. Einstein. Uber die von der molekularkinetischen theorie der warme geforderte bewegung von in ruhenden flussigkeiten suspendierten teilchen. *Ann. Phys*, 17:549–560, 1905.

A. Einstein. Zur theorie der brownschen bewegung. *Ann. Phys*, 19:371–381, 1906.

A. Ellison. Bayesian inference in ecology. *Ecology Letters*, 7:509 – 520, 05 2004. doi: 10.1111/j.1461-0248.2004.00603.x.

W. F. Fagan and J. M. Calabrese. The correlated random walk and the rise of movement ecology. *Bulletin of the Ecological Society of America*, 95(3):204–206, 2014.

E. F. Fama. Random walks in stock market prices. *Financial Analysts Journal*, 21(5):55–59, 1965.

C. H. Fleming, J. M. Calabrese, T. Mueller, K. A. Olson, P. Leimgruber, and W. F. Fagan. From fine-scale foraging to home ranges: A semi-variance approach to identifying movement modes across spatio-temporal scales. *The American Naturalist*, 183(5):E154–67, 2014a.

C. H. Fleming, J. M. Calabrese, T. Mueller, K. A. Olson, P. Leimgruber, and W. F. Fagan. Non-Markovian maximum likelihood estimation of autocorrelated movement processes. *Methods in Ecology and Evolution*, 5:462–472, 2014b.

C. H. Fleming, W. F. Fagan, T. Mueller, K. A. Olson, P. Leimgruber, and J. M. Calabrese. Rigorous home range estimation with movement data: a new autocorrelated kernel density estimator. *Ecology*, 96(5):1182–1188, 2015.

J. Frair, J. Fieberg, M. Hebblewhite, F. Cagnacci, N. DeCesare, and L. Pedrotti. Resolving issues of imprecise and habitat-biased locations in ecological analyses using GPS telemetry data. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 365:2187–200, 07 2010. doi: 10.1098/rstb.2010.0084.

J. M. Fryxell, M. Hazell, L. Börger, B. D. Dalziel, D. T. Haydon, J. M. Morales, T. McIntosh, and R. C. Rosatte. Multiple movement modes by large herbivores at multiple spatio-temporal scales. *Proceedings of the National Academy of Sciences*, 105(49):19114 – 19119, 2008.

A. E. Gelfand and E. M. Schliep. Spatial statistics and Gaussian processes: A beautiful marriage. *Spatial Statistics*, 18:86–104, 2016.

A. E. Gelfand, P. J. Diggle, M. Funtes, and P. Guttorp. *Handbook of Spatial Statistics*. CRC Press, Boca Raton, Florida, 2010. ISBN 978-1420072877.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992.

A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubi. *Bayesian Data Analysis*. CRC Press, Taylor and Francis Group, 6000 Broken Sound Parkway, NW, Suite 300, Boca Raton, FL 33487-2742, 2013. ISBN 1439840954.

J. Geweke. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Bayesian Statistics 4, Oxford University Press, Oxford, United Kingdom, 1992.

M. N. Gibbs. *Bayesian Gaussian Processes for Classification and Regression*. PhD thesis, Univ. of Cambridge, Cambridge, U.K., 1997.

T. Glad and L. Ljung. *Control Theory: Multivariable and non-linear Methods*. Taylor and Francis, Milton Park, Abingdon-on-Thames, Oxfordshire, United Kingdom, 2000. ISBN 9780748408788.

M. S. Grewal and A. P. Andrews. *Kalman Filtering, Theory and Practice using MATLAB*. John Wiley & Sons, Inc, Chichester, New York, 2011. ISBN 9781118851210.

A. Grigorievskiy, N. Lawrence, and S. Särkkä. Parallelizable sparse inverse formulation Gaussian processes (spinGP). *10.48550/ARXIV.1610.08035*, 2016.

E. Gurarie, F. Cagnacci, W. Peters, C. H. Fleming, J. M. Calabrese, T. Mueller, and W. F. Fagan. A framework for modelling range shifts and migrations: asking when, whither, whether and will it return. *Journal of Animal Ecology*, 86(4):943–959, 2017.

M. M. Guzzo, T. E. Van Leeuwen, J. Hollins, B. Koeck, M. Newton, D. M. Webber, F. I. Smith, D. M. Bailey, and S. S. Killen. Field testing a novel high residence positioning system for monitoring the fine-scale movements of aquatic organisms. *Methods in Ecology and Evolution*, 9(6):1478–1488, 2018.

R. L. Hall. Amoeboid movement as a correlated walk. *Journal of Mathematical Biology*, 4: 327–335, 1977.

H. Hang. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. 2004.

G. Harris, S. Thirgood, J. G. C. Hopcraft, J. PGM Cromsigt, and J. Berger. Global decline in aggregated migrations of large terrestrial mammals. *Endangered Species Research*, 7(1): 55–76, 2009.

K. J. Harris and P. G. Blackwell. Flexible continuous-time modelling for heterogeneous animal movement. *Ecological Modelling*, 255:29–37, 2013.

J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 379–384, 2010. doi: 10.1109/MLSP.2010.5589113.

J. Hartikainen and S. Särkkä. On convergence and accuracy of state-space approximations of squared exponential covariance functions . *In Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014. doi: 10.1109/MLSP.2014.6958890.

D. T. Haydon, J. M. Morales, A. Yott, D. A. Jenkins, R. Rosatte, and J. M. Fryxell. Socially informed random walks: incorporating group dynamics into models of population spread and growth. *Proceedings of the Royal Society B: Biological Sciences*, 275(1638):1101–1109, 2008.

M. Heinonen, O. Guipaud, F. Milliat, V. Buard, B. Micheau, G. Tarlet, M. Benderitter, F. Zehraoui, and F. d'Alche Buc. Detecting time periods of differential gene expression using gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction. *Bioinformatics*, 31(5):728–735, 2015.

M. Heinonen, H. Mannerström, J. Rousu, S. Kaski, and H. Lähdesmäki. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pages 732–740. Proceedings of Machine Learning Research (PMLR), 2016.

J. Hensman, N. Fusi, and N. Lawrence. Gaussian processes for big data. *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*, 09 2013. doi: 10.48550/ARXIV.1309.6835.

J. Hensman, A. G. Matthews, M. Filippone, and Z. Ghahramani. MCMC for variationally sparse Gaussian processes. 2015. doi: 10.48550/ARXIV.1506.04000.

P. Hiltunen, S. Särkkä, I. Nissila, A. Lajunen, and J. Lampinen. State space regularization in the non stationary inverse problem for diffuse optical tomography. *Inverse Problems*, 27(2), 2011.

J. L. Hodges. The significance probability of the Smirnov two-sample test. *Arkiv för Matematik*, 3(5):469 – 486, 1958. doi: 10.1007/BF02589501.

M. B. Hooten and D. S. Johnson. Basis function models for animal movement. *Journal of the American Statistical Association*, 112(518):578–589, 2017.

M. B. Hooten, D. S. Johnson, B. T. McClintock, and J. M. Morales. *Animal Movement Statistical Models for Telemetry Data*. CRC Press, Boca Raton, Florida, United States, 03 2017. ISBN 9781315117744. doi: 10.1201/9781315117744.

J. G. C. Hopcraft, J. M. Morales, H. L. Beyer, M. Borner, E. Mwangomo, A. Sinclair, H. Olff, and D. T. Haydon. Competition, predation, and migration: individual choice patterns of Serengeti migrants captured by hierarchical models. *Ecological Monographs*, 84(3):355–372, 2014.

O. Ibe. *Markov Processes for Stochastic Modeling (second edition)*. Elsevier, 9-10 St Andrew Square, Edinburgh, United Kingdom, 2016. ISBN 9780323282956.

D. D. Johnson and D. C. Ganskopp. GPS collar sampling frequency: Effects on measures of resource use. *Rangeland Ecology & Management*, 2008. ISSN 0022-409X. doi: 10.2111/07-044.1.

D. S. Johnson, J. London, M. Lea, J., and Durban. Continuous-time correlated random walk model for animal telemetry data. *Ecology*, 89:1208–1215, 2008. doi: 10.1890/07-1032.1.

P. J. Jones, A. Sim, H. B. Taylor, L. Bugeon, M. J. Dallman, B. Pereira B, M. P. Stumpf, and J. Liepe. Inference of random walk models to describe leukocyte migration. *Physical Biology*, 12(6), 2015.

P. M. Kareiva and N. Shigesada. Analyzing insect movement as a correlated random walk. *Oecologia*, 56:234–238, 1983.

R. Kays, M. C. Crofoot, W. Jetz, and M. Wikelski. Terrestrial animal tracking as an eye on life and planet. *Science*, 348(6240), 2015.

K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic Gaussian process regression. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 393–400, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273546.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2017. doi: 10.48550/ARXIV.1412.6980.

C. Kluppelberg, A. Lindner, and R. Maller. *Continuous time volatility modelling: COGARCH versus Ornstein-Uhlenbeck models (From Y. Kabanov and R. Liptser and J. Stoyanov (Eds.): From stochastic calculus to mathematical finance)*. Springer, Berlin, Germany, 04 2007. ISBN 978-3-540-30782-2. doi: 10.1007/978-3-540-30788-4_21.

A. N. Kolmogorov. Dissipation of energy in locally isotropic turbulence. *Doklady Akademii Nauk SSSR*, 32:16–18, 1941.

R. Lande. Natural-selection and random genetic drift in phenotypic evolution. *Evolution*, 30: 314–334, 1976.

T. Lang, C. Plagemann, and W. Burgard. Adaptive non-stationary kernel regression for terrain modeling. In *Robotics: Science and Systems III*, 06 2007.

R. Langrock, R. King, J. Matthiopoulos, L. Thomas, D. Fortin, and J. M. Morales. Flexible and practical modeling of animal telemetry data: Hidden Markov models and extensions. *Ecology*, 93(11):2336–2342, 2012.

R. Langrock, J. G. C. Hopcraft, P. G. Blackwell, V. Goodall, R. King, M. Niu, T. A. Patterson, M. W. Pedersen, A. Skarin, , and R. S. Schick. Modelling group dynamic animal movement. *Methods in Ecology and Evolution*, 5:190–199, 2014.

R. Langrock, T. Kneib, A. Sohn, and S. DeRuiter. Non-parametric inference in hidden Markov models using p-splines. *Biometrics*, 71, 01 2015. doi: 10.1111/biom.12282.

M. Lázaro-Gredilla and M. Titsias. Variational Heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 841–848, 01 2011.

N. Lemoine. Moving beyond noninformative priors: why and how to choose weakly informative priors in Bayesian analyses. *Oikos*, 128, 04 2019. doi: 10.1111/oik.05985.

J. Liepe, H. Taylor, C. P. Barnes, M. Huvet, L. Bugeon, T. Thorne, J. R. Lamb, M. J. Dallman, and M. P. H. Stumpf. Calibrating spatio-temporal models of leukocyte dynamics against in vivo live-imaging data using approximate Bayesian computation. *Integrative Bioliogy*, 4(3): 335–345, 2012.

F. Lindgren, H. Rue, and J. Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

M. Løvschal, P. K. Bøcher, J. Pilgaard, I. Amoke, A. Odingo, A. Thuo, and J. C. Svenning. Fencing bodes a rapid collapse of the unique Greater Mara ecosystem. *Scientific Reports*, 7 (1):1–7, 2017.

B. Macdonald, C. Higham, and D. Husmeier. Controversy in mechanistic modelling with Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1539–1547, Lille, France, 07–09 Jul 2015. PMLR.

D. J. C. MacKay. Introduction to Gaussian processes (technical report). Technical report, Univ. of Cambridge, Cambridge, UK, 1997.

R. Mancini. *Op Amps for Everyone: Design Reference 5*. Newnes, Halley Court, Jordan Hill, Oxford, OX2 8EJ, United Kingdom, 2003. ISBN 0750677015.

K. V. Mardia and P. E. Jupp. *Directional Statistics*. World Scientific, New Jersey and London, 1999. ISBN 0124711502.

G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5:439–468, 1973.

M. A. McCarthy and P. Masters. Profiting from prior information in Bayesian analyses of ecological data. *Journal of Applied Ecology*, 42:1012–1019, 2005.

B. T. McClintock, R. King, L. Thomas, J. Matthiopoulos, B. J. McConnell, and J. M. Morales. A general discrete-time modeling framework for animal movement using multi-state random walks. *Ecological Monographs*, 82(3):335–349, 2012.

B. T. McClintock, D. S. Johnson, M. B. Hooten, J. Hoef, and J. M. Morales. When to be discrete: the importance of time formulation in understanding animal movement. *Movement Ecology*, 2(21), 2014. doi: 10.11159/icsta19.27.

T. Michelot and P. G. Blackwell. State-switching continuous-time correlated random walks. *Methods in Ecology and Evolution*, 10:637–649, 2019.

T. Michelot, R. Langrock, and T. A. Patterson. moveHMM: An R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, 7, 04 2016. doi: 10.1111/2041-210X.12578.

T. Michelot, P. G. Blackwell, S. C. Jammes, and J. Matthiopoulos. Inference in MCMC step selection models. *Biometrics*, 76(2):438–447, 2020.

T. Michelot, R. Glennie, C. Harris, and L. Thomas. Varying-coefficient stochastic differential equations with applications in ecology. *Journal of Agricultural, Biological and Environmental Statistics*, 26:446–463, 2021. doi: 10.1007/s13253-021-00450-6.

K. Monterrubio-Gòmez and S. Wade. On MCMC for variationally sparse Gaussian processes: A pseudo-marginal approach. 2021. doi: 10.48550/ARXIV.2103.03321.

K. Monterrubio-Gòmez, L. Roininen, S. Wade, T. Damoulas, and M. Girolami. Posterior inference for sparse hierarchical non-stationary models. 2019. doi: 10.48550/ARXIV.1804.01431.

J. M. Morales, D. T. Haydon, J. Frair, K. E. Holsinger, and J. M. Fryxell. Extracting more out of relocation data: building movement models as mixtures of random walks. *Ecology*, 85(9): 2436–2445, 2004.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*. The MIT Press, Cambridge, Massachusetts and London, England, 2012. ISBN 0262018020.

R. Nathan, W. M. Getz, E. Revilla, M. Holyoak, R. Kadmon, D. Saltz, and P. E. Smouse. A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, 105(49):19052–19059, 2008. doi: 10.1073/pnas.0800375105.

R. Neal. An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111:194 – 203, 1992.

R. Neal. *Bayesian Learning for Neural Networks*. Springer, Berlin, Germany, 1996. ISBN 0387947248.

P. Nouvellet, J. P. Bacon, and D. Waxman. Fundamental insights into the random movement of animals from a single distance related statistic. *The American Naturalist*, 174(4):506–514, 2015.

A. O'Hagan and J. F. C. Kingman. Optimum smoothing of two-dimensional fields. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978.

B. Oksendal. *Stochastic Differential Equations, An Introduction with Applications (5th edition)*. Springer, Berlin, Germany, 1998. ISBN 3540047581.

C. Paciorek and M. Schervish. Non-stationary covariance functions for Gaussian procesess regression. In *Neural Information Processing Systems (NIPS)*, page 273–280, 2004.

C. Paciorek and M. Schervish. Spatial modelling using a new class of non-stationary covariance functions. *Environmetrics*, 17:483–506, 2006.

G. P. Panotopoulos, S. Aguayo, and Z. S. Haidar. The "extreme dumping limit" for cell-to-cell communications. *Journal of Healthcare Engineering*, 2018, 2018. doi: 10.1155/2018/9680713.

T. A. Patterson, M. Basson, M. V. Bravington, and J. S. Gunn. Classifying movement behaviour in relation to environmental conditions using Hidden Markov models. *Journal of Animal Ecology*, 78(6):1113–1123, 2009.

T. A. Patterson, B. Mcconnell, M. Fedak, M. Bravington, and M. Hindell. Using GPS data to evaluate the accuracy of state–space methods for correction of argos satellite telemetry error. *Ecology*, 91:273–85, 01 2010. doi: 10.1890/08-1480.1.

I. Paun, D. H. Husmeier, and C. J. Torney. A study on discrete-time movement models. In *Proceedings of the International Conference on Statistics: Theory and Applications (ICSTA'19)*, 08 2019.

K. Pearson. The problem of the random walk. *Nature*, 72(1):294, 1905.

M. Plumlee and V. R. Joseph. Orthogonal Gaussian process models. *Statistica Sinica*, 11 2016. doi: 10.5705/ss.202015.0404.

H. K. Preisler, D. R. Brillinger, A. A. Ager, J. G. Kie, and R. P. Akers. Stochastic differential equations: a tool for studying animal movement. In *Proceedings of the IUFRO Conference Session on Forest Biometry Modelling and Information Science*, pages 1–9, 2001.

H. K. Preisler, A. A. Ager, B. K. Johnson, and J. G. Kie. Modeling wildlife movements using stochastic differential equations. *Environmetrics*, 15:643–657, 2004.

H. K. Preisler, A. A. Ager, and M. J. Wisdom. Analyzing animal movement patterns using potential functions. *Ecosphere*, 4(32), 2013.

C. M. Prokopenko, M. S. Boyce, and T. Avgar. Characterizing wildlife behavioural responses to roads using integrated step selection analysis. *Journal of Applied Ecology*, 54:470–479, 2017.

P. Protter. *Stochastic Integration and Differential Equations (2nd ed.)*. Springer, Berlin, Germany, 2004. ISBN 366202621X.

C. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, page 881–888. MIT Press, 2002.

C. E. Rasmussen and K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts, 2006. ISBN 9780262182539.

L. Rayleigh. The problem of the random walk. *Nature*, 72:318, 1905.

A. A. Rija and J. R. Kideghesho. Poachers' strategies to surmount anti-poaching efforts in western Serengeti, Tanzania. In *Protected Areas in Northern Tanzania*, pages 91–112. Springer, 05 2020. ISBN 978-3-030-43301-7. doi: 10.1007/978-3-030-43302-4_7.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, 71(2):319–392, 2009.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

H. Salimbeni and M. Deisenroth. Doubly stochastic variational inference for Deep Gaussian processes. 2017. doi: 10.48550/ARXIV.1705.08933.

S. Särkkä. Stochastic (partial) differential equations and Gaussian processes (lecture slides). 2017. URL `http://gpss.cc/gpss17/slides/spde-lecture.pdf`.

S. Särkkä and J. Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. *AISTATS: Fifteenth International Conference on Artificial Intelligence and Statistics*, 22:993–1001, 2012.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019. doi: 10.1017/9781108186735.

S. Särkkä, A. Solin, and J. Hartikainen. Spatio-temporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, 30(5):51–61, 2013. doi: 10.1109/MSP.2013.2246292.

A. D. Saul, J. Hensman, A. Vehtari, and N. D. Lawrence. Chained Gaussian processes. 2016. doi: 10.48550/ARXIV.1604.05263.

D. W. Scott. On optimal and data-based histograms. *Ecological Monographs*, 66:605–610, 1979.

S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-time models*. Springer, Berlin, Germany, 2004. ISBN 978-1-4419-2311-0. doi: 10.1007/978-1-4757-4296-1.

D. P. Siniff and C. R. Jessen. A simulation model of animal movement patterns. *Advances in Ecological Research*, 6:185–219, 1969.

J. G. Skellam. Random dispersal in theoretical populations. *Biometrika*, 38(1-2):196–218, 1951.

J. G. Skellam. The formulation and interpretation of mathematical models of diffusionary processes in biology. *The Mathematical Theory of the Dynamics of Biological Populations*, pages 63–85, 01 1973.

M. Smoluchowski. Drei vortrage uber diffusion, brownsche bewegung und koagulation von kolloidteilchen. *Phys. Zeit*, 17:557–582, 1916.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 18. MIT Press, 2006.

E. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximations. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 524–531, San Juan, Puerto Rico, 21-24 Mar 2007. Proceedings of Machine Learning Research (PMLR).

C. E. Studds, B. E. Kendall, N. J. Murray, H.B. Wilson, D. I. Rogers, R. S. Clemens, K. Gosbell, C. J. Hassell, R. Jessop, D. S. Melville, et al. Rapid population decline in migratory shorebirds relying on Yellow sea tidal mudflats as stopover sites. *Nature Communications*, 8(1):1–7, 2017.

H. B. Taylor, J. Liepe, C. Barthen, L. Bugeon, M. Huvet, P. D. Kirk, S. B. Brown, J. R. Lamb, M. P. Stumpf, and M. J. Dallman. P38 and jnk have opposing effects on persistence of in vivo leukocyte migration in zebrafish. *Immunol Cell Biol.*, 91(60-69):21–86, 2013.

P. D. Thompson. Optimum smoothing of two-dimensional fields. *Tellus*, 8:384–393, 1956.

H. Thurfjell, S. Ciuti, and M. S. Boyce. Applications of step-selection functions in ecology and conservation. *Movement Ecology*, 2(4), 2014.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. V. Dyk and M. Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16-18 Apr 2009. Proceedings of Machine Learning Research (PMLR).

V. Tolvanen, P. Jylänki, and A. Vehtari. Expectation propagation for nonstationary heteroscedastic Gaussian process regression. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014. doi: 10.1109/MLSP.2014.6958906.

C. J. Torney, J. G. C. Hopcraft, T. A. Morrison, I. D. Couzin, and S. A. Levin. From single steps to mass migration: the problem of scale in the movement ecology of the Serengeti wildebeest. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1746):20170012, 2018a.

C. J. Torney, M. Lamont, L. DeBell, R. Angohiatok, L. M. Leclerc, and A. Berdahl. Inferring the rules of social interaction in migrating caribou. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373:20170385, 2018b. doi: 10.1098/rstb.2017.0385.

C. J. Torney, J.M. Morales, and D. Husmeier. A hierarchical machine learning framework for the analysis of large scale animal movement data. *Movement Ecology*, 9(6), 2021.

V. Tresp. Mixtures of Gaussian processes. In *Neural Information Processing Systems (NIPS 13)*, page 654–660, 2001.

M. A. Tucker, K. Böhning-Gaese, W. F. Fagan, J. M. Fryxell, B. Van Moorter, S. C. Alberts, A. H. Ali, A. M. Allen, N. Attias, T. Avgar, et al. Moving in the Anthropocene: Global reductions in terrestrial mammalian movements. *Science*, 359(6374):466–469, 2018.

P. Turchin. *Quantitative analysis of movement: measuring and modeling population redistribution in animals and plants*. Sinauer Associates, Sunderland, MA, 1998. ISBN 9780878938476.

L. Tweedy, D. A. Knecht, G. M. Mackay, and R. H. Insall. Self-generated chemoattractant gradients: Attractant depletion extends the range and robustness of chemotaxis. *PLoS biology*, 14(3), 1977.

G.E. Uhlenbeck and L.S. Ornstein. On the theory of Brownian motion. *Physical Review*, 36: 823–841, 1930.

O. Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5(2):177–188, 1977.

P. Vatiwutipong and N. Phewchean. Alternative way to derive the distribution of the multivariate Ornstein–Uhlenbeck process. *Advances in Difference Equations*, 276, 2019.

D. D. Vvedensky. *Transformations of Materials*. 2053-2571. Morgan & Claypool Publishers, 2019. ISBN 978-1-64327-620-5. doi: 10.1088/2053-2571/ab191e.

Y. Wang, M. A. Brubaker, B. Chaib-draa, and R. Urtasun. Sequential inference for deep Gaussian process. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 694–703, Cadiz, Spain, 09–11 May 2016.

J. S. Wesner and J. P. F. Pomeranz. Choosing priors in Bayesian ecological models by simulating from the prior predictive distribution. *bioRxiv*, 2020.

A. Whetten. Smoothing splines of apex predator movement: Functional modeling strategies for exploring animal behavior and social interactions. *Ecology and Evolution*, 11:17786–17800., 12 2021. doi: 10.13140/RG.2.2.16722.68806/2.

P. Whittle. Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, 40:974–994, 1963.

N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series With Engineering Applications*. The MIT Press, Cambridge, Massachusetts, 1949. ISBN 1614275173.

J. J. Wiens, D. D. Ackerly, A. P. Allen, B. L. Anacker, L. B. Buckley, H. V. Cornell, E. I. Damschen, T. J. Davies, J. A. Grytnes, S. P. Harrison, B. A. Hawkins, R. D. Holt, C. M. McCain, and P. R. Stephens. Niche conservatism as an emerging principle in ecology and conservation biology. *Ecology Letters*, 13:1310–1324, 2010.

D. S. Wilcove and M. Wikelski. Going and going, gone: Is animal migration disappearing. *PLoS Biology*, 6, e188, 2008.

C. E. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

C. C. Wilmers, B. Nickel, C. M. Bryce, J. A. Smith, R.E. Wheat, and V. Yovovich. The golden age of bio-logging: How animal-borne sensors are advancing the frontiers of ecology. *Ecology*, 96(7):1741–1753, 2015.

A. G. Wilson and R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pages 1067–1075, Atlanta, Georgia, USA, 17–19 Jun 2013. Proceedings of Machine Learning Research (PMLR).

G. Wittemyer, P. Elsen, W. T. Bean, A.C. Burton, and J. S. Brashares. Accelerated human population growth at protected area edges. *Science*, 321(5885):123–126, 2008a.

G. Wittemyer, L. Polansky, I. Douglas-Hamilton, and W. M. Getz. Disentangling the effects of forage, social rank, and risk on movement autocorrelation of elephants using Fourier and

wavelet analyses. *Proceedings of the National Academy of Sciences of the United States of America*, 105:19108–19113, 2008b.