Novel applications of machine learning for the study of animal movement

Nazareno Campioni

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

School of Mathematics & Statistics
College of Science and Engineering
University of Glasgow

October 2023

Eat raw meat. Sleep. Repeat.

Abstract

Animal movement data play a crucial role in our endeavour to decode the wildlife dynamics that unfold across the Earth's varied terrains and waterways. Collecting animal movement data involves employing a variety of modern technologies and techniques. One common method is the use of Global Positioning System (GPS) devices that are attached to animals, providing accurate location data at regular intervals. These devices allow us to track animals' movements over time and space, giving us access to intricate details about their ranging behaviours and daily routines.

These data provide insights into migration patterns, foraging strategies, habitat preferences, and responses to environmental changes and are therefore key to an improved understanding of the behaviours, habitats and ecological interactions of various species. There are many challenges associated with the analysis of such datasets and movement ecologists have been dilligently advancing state-of-the-art statistical methods for analysing animal movement data. This effort is crucial because it enables us to extract meaningful information from the complex, large-scale datasets generated by animal tracking studies. These advanced statistical techniques help uncover hidden patterns, such as the identification of significant stopover sites during migrations or the characterisation of nuanced movement behaviours. Moreover, they facilitate the integration of environmental variables, enhancing our ability to understand how animals respond to changing landscapes and climate conditions. By refining our analytical tools, movement ecologists can provide more accurate and comprehensive insights into wildlife behaviours, aiding conservation efforts and ecological research on a global scale.

Throughout this thesis, we will talk about models of animal movement and we will give our contribution to expand the array of statistical methods that can be employed for the analysis of telemetry datasets. We will begin by reviewing some of the most commonly employed methods found in the literature. Then, we will focus on a simple one-dimensional self-propelled particle model used to simulate the dynamics of a group of locusts placed in a ring-shaped arena for an experiment, and we will leverage Gaussian processes to infer microscale properties of the group from macroscale observed variables, without deriving a formal mathematical link between the two scales, which is in many cases intractable.

Our focus will then shift to the problem of identifying different behavioural patterns from relocation data. In the fourth chapter we will describe various methods that we have concep-

tualised all aimed at simulating semi-Markov chains. This will be needed in the subsequent chapter, where we will introduce a flexible model scalable to large datasets that finds its applications in solving the so-called switching problems. This will be achieved by modelling the locations via an integrated OU process and by reconstructing the latent behavioural patterns via a Monte Carlo Expectation-Maximisation algorithm, where the method introduced in the previous chapter will be employed.

In Chapter 6 we will employ this method on a flock of sheep, specifically on a group-level metric describing the changes in coordination of the group motion. This will enable us to reconstruct the behavioural pattern of the flock. We end the thesis with a conclusion chapter.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank from the bottom of my heart my supervisors Dirk, Juan and Colin for the enormous patience and care they have always shown me during these 4 years of PhD. I also thank them for all the help I was given - without them I could not possibly be in the position where I am now. And I would also like to thank Dirk and Colin again for offering me this PhD position 4 years ago, when I simply showed interest in working with them back in Summer 2019 - you have offered me the possibility to grow not only under an academical point of view, but also on a personal level.

I take this (maybe) once-in-a-life-time opportunity to express my gratitude towards the people of my inner circle that have enriched my private life so greatly. I'm grateful for my mum, my dad, my sister for their authenticity and I'm grateful to have such a smart and generous young nephew. I'm especially grateful for my beloved cat Harry who left us recently, for all his 18 years of purring and affection.

I'm grateful for my beautiful girlfriend Alice and for all the numerous adventures we have shared and for those that we will share in the future. I'm grateful for Matteo, Daniele, Fabio, Elena, Francesco, Stefano and il Trafe because I've come to the realisation that it is very hard to surround yourself with like-minded people. I'm grateful for my band, Riding Rhinos - we rock!! Also, I'm grateful for my University friends Federico, with whom I've shared 8 years in Glasgow and always made me feel somewhat at home, Ross, the most friendly, kindhearted, Scotsman out there, and Cyrus, because even though we became friends only recently I'm very happy to have shared my PhD journey with such a humble, open-minded person. With my inner circle, we've always been there for each other no matter what and I'm very grateful for this.

Finally, I also want to thank myself, for not giving up and for continuing learning at the best of my abilities although numerous were the times I have felt incapable of performing an adequate research.

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, Juan M. Morales and Colin J. 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
GP: Gaussian Process
HMM: Hidden Markov model
DTMC : discrete-time Markov chain
CTMC: continuous-time Markov chain
IGM: infinitesimal generator matrix
CRW: correlated random walk
CTCRW: continuous-time correlated random walk
OU: Ornstein-Uhlenbeck
IBM: individual-based modeling
SSM: state-space model
MLE: Maximum Likelihood Estimation
EM: Expectation-Maximisation
MCMC: Markov Chain Monte Carlo
VI: variational inference
MH: Metropolis-Hastings
RJMCMC: reversible jump Markov chain Monte Carlo
KL: Kullback-Leibler
MVN: multivariate normal
RBF: radial basis function
FPE: Fokker-Planck equation
SPP: self-propelled particle
SPD: stationary probability density
MCEM: Monte Carlo Expectation-Maximisation
SDE: stochastic differential equation

Mathematical notation:

x: scalar
$\mathbf{x}$: vector

X: matrix

Chapter 1

Introduction

The phenomenon of animal movement, ranging from the intricate aerial maneuvers of avian species to the extensive migratory patterns exhibited by wildebeest, constitutes a subject of profound scientific intrigue within the natural world. In epochs characterised by a celestial backdrop that underscored the interconnectedness of the universe, the observation and analysis of nature held intrinsic appeal and necessity. During these periods, an acute understanding of animal locomotion was essential for subsistence, primarily in the context of hunting activities.

However, as time elapsed and societal dynamics evolved, our motivations for studying animal movement underwent a discernible shift. Modern times have witnessed a departure from the imperative of survival-driven inquiry to one marked by a more diversified and nuanced set of interests. While vestiges of traditional practices, such as bird-watching and, to a diminishing extent, hunting, persist as niches of fascination, contemporary motivations have expanded beyond the realm of leisurely observation.

Presently, the study of animal movement encompasses a multifaceted domain known as movement ecology, characterised by a commitment to understanding and mitigating the human impact on the natural world. The driving force behind this shift lies in the recognition of our responsibility towards safeguarding the welfare of non-human species. Nevertheless, it is evident that, in certain instances, the study of animal movement serves as a pretext for the development of advanced statistical methodologies. These methodological innovations, although refined and rigorously tested within the scientific community, often lack immediate applicability in practical, real-world contexts.

This need, or rather will, to create more sophisticated mathematical and statistical models of animal movement is justified by the technological advances in the field of tracking devices, such as GPS tags, that have been employed to track various animals across their habitats. Indeed, more-lasting batteries and higher-precision devices have led to an increasing amount of high-resolution animal tracking data being collected (Nathan et al., 2022, Cagnacci et al., 2010), thus the increase of new statistical methodologies developed to tackle the challenges

associated with the analysis of such large data sets.

When it comes to analysing a set of data, it is important to understand the type of data that is collected. In this thesis we are mainly interested in telemetry datasets that contain the positions of animals in space over a sequence of discrete points in time. The animals' positions recorded via GPS are stored as latitude and longitude coordinates. Given the spherical nature of these measurements, they are often projected onto the Universal Transverse Mercador (UTM) grid so that Euclidean geometry applies. It is very important to define a metric of movement that can be used during the analysis. In the movement ecology literature, one of the most commonly employed movement metrics is the bivariate time series of step lengths and turning angles. Step lengths are defined as the distance between two consecutive locations, whereas the turning angles are the change in the direction of movement considering three consecutive locations. This is illustrated in Figure 1.1.

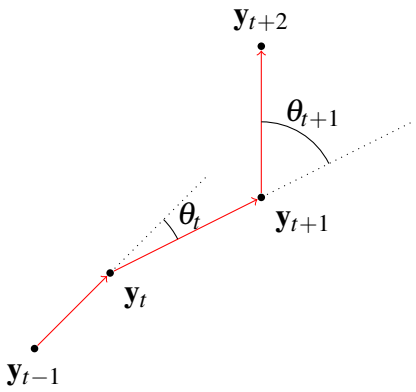


Figure 1.1: Illustration of step lengths and turning angles. The length of the red lines will be the step lengths. The turning angles are θ_{t+1} and θ_t.

However, other metrics of movement can be employed. As an example, a similar metric makes use of step length and bearing, which is defined as $tan^{-1}\ (y_t - y_{t-1})/(x_t - x_{t-1})$, whereas in other cases the positions themselves are used as metrics of movement.

One other important choice that will impact the analysis is the choice of temporal domain, namely whether the model is defined in discrete-time or continuous-time. Clearly, animal movement is a continuous-time continuous-space process, however this has been approximated by discrete-time discrete-space process in the literature. The topic of discrete- vs continuous-time model has been analysed thoroughly in McClintock et al. (2014).

Contrarily to what many researchers believed, McClintock et al. (2014) showed that modelling telemetry data in discrete-time and continuous-time are not merely two different approaches used for the same end. Discrete-time models are generally preferred in the movement ecology community because of their simpler mathematical structure and easier interpretability of parameters, whereas continuous-time models may discourage practitioners because of the perhaps more difficult biological interpretation of the parameters. Having said that, in the work of McClintock et al. (2014) it is shown that these two models also differ in

the formulation for step length and bearings distributions. In the continuous-time case, the step length and bearings are correlated, specifically as the step length increases, the distribution of the bearings becomes more concentrated around the velocity bearing. Furthermore, assuming that the movement parameters are fixed (namely the model is constant state), in the continuous-time model step lengths are correlated via the auto-correlated speed process, which implies that the model maintains directional persistence as well as persistence in speed. This does not apply to discrete-time models.

As stated before, discrete-time models have been the preferred choice by the movement ecologists in the past year, however such models have some disadvantages. One disadvantage is that discrete-time models are not time scale-invariant (McClintock et al., 2014). This means that the analysis must be preceded by the choice of the sampling interval, that is, the time interval between observations. The sampling interval varies significantly across studies, ranging from fractions of seconds to days. Thus, the choice of the sampling interval has a significant impact on the types of inferences that can be drawn and the appropriate modelling techniques. Hence, careful considerations about the animal's behavioural dynamics and the goals of the analysis need to be taken when selecting the sampling interval (Patterson et al., 2017a). Additionally, in the case of missing or irregularly sampled data, discrete-time models need the movement path to be discretised into temporally-regular locations, which may result in a greater computational cost than that associated with some continuous-time models (McClintock et al., 2014).

Despite of the differences between discrete-time and continuous-time models, Gurarie et al. (2017) proposed characteristic spatial and temporal scales to unify all models of animal movement. Generally, movement can be characterised by two clearly distinct limits - a so called ballistic limit, in which the correlation in movement is observed at high sampling rates, and a diffusive limit, meaning at large scales the movement can be approximated by diffusion processes. This motivated the conceptualisation of the characteristic time and spatial scales to rigorously quantify the transitions between the two movement limits, that is the transitions between correlated movements and uncorrelated movement. The characteristic time scale will be explored in more details in Chapter 5, where we employ a diffusion process to model animal movement.

In this work, we only focus on two-dimensional measurements consisting of latitude and longitude, however there are also studies that consider three-dimensional observations or one-dimensional observations, where in the latter case the observations may represent the diving and emerging behaviours of air-breathing marine mammals. As we shall see, in Chapters 5 and 6 we will use the Easting-Northing positions themselves as movement metrics employed in our method.

1.1 Aim of the thesis

The animal movement discipline is vast and there are many models that can be used to analyse location data, each depending on the scope of the analysis. In this thesis, we introduce new models of animal movement that can be employed in various different analysis tasks. Firstly, in Chapter 2 we give an overview of the background theory that is needed to understand the material discussed in the later chapters. Specifically, since sampling of locations is done at regular time intervals, we describe hidden Markov models (HMMs) and state-space models (SSMs), which are particularly suitable for this type of analysis. We will see how these methods may not be well-suited for observations irregularly spaced in time, and how continuous-time approaches, such as those based on diffusion models, are more accommodating of irregular time intervals, whether intentional or due to limitations within the telemetry devices.

In Chapter 3 the analysis takes a "top-down" approach, meaning that from group-level information our aim is to infer individual-level information. We study a one-dimensional self-propelled particle (SPP) model to simulate the dynamics of locusts in a ring-shaped arena. We employ Gaussian processes (GP) to link macroscale, group-level properties to the microscale, individual-level properties of the system without the need of a formal mathematical equation. Specifically, we first assume the existence of an empirical Fokker-Planck equation (FPE), namely a partial differential equation that describes the evolution in time of the probability density of a macroscale variable (which will be defined). Then, through the application of sparse Gaussian process (GP) regression (Lázaro-Gredilla and Titsias, 2011, Saul et al., 2016) we learn the drift and diffusion functions of the FPE, which allows us to estimate the likelihood of microscale parameters given a set of empirical macroscale observations, thus linking the two scales. We also introduce a novel, adaptive sampling algorithm that makes use of a second Gaussian process to emulate the log-likelihood surface of the microscale parameters, thus reducing the computational cost of the algorithm.

In the remaining chapters we shift focus to a different problem. We tackle the so-called switching problems, where the animal's dynamics are assumed to be dependent on quantitatively different behaviours, or states, thus resulting in quantitatively different trajectories. Generally speaking, these types of models are particularly useful to model the changes in dynamics of an animal which are expected to occur over longer time-scales. Indeed, over the hours and days, we would expect an animal to show different dynamics that could be associated with different behaviours, such as an exploratory behaviour or a resting behaviour, etc.

In Chapter 4 we introduce the different approaches we have taken in order to simulate continuous-time semi-Markov chains. Simulating semi-Markov chains is a fundamental block for our method developed and explained in the subsequent Chapter 5, aimed at reconstructing the history of the behavioural transitions. In this chapter we introduce a novel,

scalable, non-Markovian model of animal movement. The model assumes the dynamics of the animal can be approximated by an integrated Ornstein-Uhlenbeck process whose parameters depend on an underlying (latent) semi-Markov process. We then employ a Monte-Carlo Expectation-Maximisation algorithm to reconstruct the latent state sequences as well as to optimise the model parameters. This method will be applied both to synthetic data as well as data representing the locations of free-roaming Merino sheep in a large (400 Ha) paddock in Patagonia.

In Chapter 6 we develop this method further to accommodate different waiting distributions for the semi-Markov process. From the sheep dataset we will derive an order parameter from which we can measure the degree of cohesiveness in the group. Indeed, higher values of the order parameter correspond to more ordered movement whereas lower values correspond to low speeds or chaotic movement. We conclude with a general discussion in Chapter 7.

Chapter 2

Background theory

In this chapter we are going to review some of the fundamental theoretical results found in the literature useful to get a better understanding of the material covered in the subsequent chapters.

2.1 Introduction

Complex systems exist in nature and throughout society. Examples of complex systems are the coordinated movement of a bird flock, the spread of political ideas or that of a disease. The common thread binding these different examples is that these emergent, global-scale phenomena all arise from fine-scale interactions occurring at the individual level. In this work we only focus on animal movement, one such example of a complex system. Understanding animal movement is important for conservation and management, especially in our modern times that are characterised by heightened anthropogenic pressures on the environment. Understanding how animals react to new changes in the environment would provide significant insights and guide the formulation of ecologically sensitive measures that mitigate adverse impacts on animal well-being.

Simulation and inference are two powerful tools available to the movement ecologists (as well as other disciplines) to study animal movement. In this introductory chapter, we will go through some of the most common simulation and inference techniques used across this domain that will be also employed in some of the following chapters. Through simulations we are able to explore the underlying dynamics of movement, and through inference we are able to recover the model parameters that best fit the movement data. We will explain in more details the tools that played a major role in this thesis: Gaussian processes (GPs) and hidden Markov models (HMMs). Many concepts introduced in Section 2.2, the material covered in Section 2.3 as well as the HMMs machinery are based on the concept of Markov chain, or at least benefit from the Markov property, hence we shall give a definition of Markov chain before diving into the core part of this chapter.

A Markov chain is a stochastic process defined by a sequence of random events $\{X_n : n \in \mathbb{N}\}$, often called states when applied to animal movement modelling, for which the Markov property holds:

$$p(X_n = x_n | X_1 = x_1, \ldots, X_{n-1} = x_{n-1}) = p(X_n = x_n | X_{n-1} = x_{n-1}), \tag{2.1}$$

that is, the probability of the chain taking a new value x_n solely depends on the value of the chain at the previous step.

A Markov chain can be formulated both in discrete-time and continuous-time. Discrete-time Markov chains (DTMCs) are characterised by a transition probability matrix

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & \ldots & p_{1n} \\ \ldots & \ldots & \ldots \\ p_{n1} & \ldots & p_{nn} \end{pmatrix}, \tag{2.2}$$

where $\forall i, j \in \{0, \ldots, n\},\ p_{ij} \geq 0$ and $\sum_j p_{ij} = 1$. Here the parameters p_{ij} represent the probabilities of moving from state i to state j. The sojourn (or residence) times in each state follow

a geometric distribution (Zucchini et al., 2009).

On the other hand, continuous-time Markov chains (CTMCs) are characterised by an infinitesimal generator matrix (IGM):

$$\mathbf{\Lambda} = \begin{pmatrix} -\lambda_{11} & \dots & \lambda_{1n} \\ \dots & \dots & \dots \\ \lambda_{n1} & \dots & -\lambda_{nn} \end{pmatrix}, \tag{2.3}$$

where $\forall i, j \in \{1, \dots, n\}$, $\lambda_{ii} = \sum_{i \neq j} \lambda_{ij}$. The parameters λ_{ij} represent the instantaneous rate at which the state i transitions to state j. As a consequence of the Markov property, the sojourn times of a state i is exponentially distributed with rate parameter λ_{ii} (Hobolth and Stone, 2009).

2.2 Animal Movement Modelling

2.2.1 Random walks

Random walk theory finds its origins in the erratic movement of individual pollen particles, famously explored by the botanist Brown in 1828, now called Brownian motion. The initial rudimentary models of random walks were uncorrelated and unbiased. In this context, "uncorrelated" means that each movement's direction is entirely unrelated to previous ones; the location after each step in the random walk relies solely on the prior step's location, that is the process is Markovian (Weiss, 1994). "Unbiased" implies that there is no favoured direction; each step's direction is entirely random.

The equation of a simple discrete-space, one-dimensional random walk is the following:

$$x_t = x_{t-1} + z_t, \tag{2.4}$$

where x is the position at time t and

$$z_t = \begin{cases} 1, \text{with probability } \frac{1}{2}, \\ -1, \text{with probability } \frac{1}{2}. \end{cases} \tag{2.5}$$

Random walk models alone however are not suitable models to describe animal movement as we know that most animals exhibit a tendency to move forward (persistence). This is what lead to the conceptualisation of correlated random walks (CRWs), in that they introduce a connection between consecutive step orientations, namely a persistence. This introduces a localised directional tendency where each step leans towards the previous one's direction. However, the initial motion's influence decreases over time, and in the long term, step orientations become uniformly distributed, as explained by Benhamou (2006).

A simple modification of Equation 2.5 leads to a simple equation of a correlated random walk model introduced by Renshaw and Henderson (1981):

$$x_t = x_{t-1} + \varepsilon_t, \tag{2.6}$$

where in this case the random variable ε_t represents the probability that the step is taken in the same direction as the previous step. For example, if ε_t follows a Bernoulli distribution then the next step will be in the same direction as the previous step with probability p and in the contrary direction with probability $1-p$.

The CRWs models have been used extensively in the animal movement literature. Contrarily to random walks, some CRW processes are not Markovian as the position at a specific time step depends on previous locations too. In Chapter 4, we will describe in greater details a continuous-time correlated random walk (CTCRW) model based on a diffusion process (Johnson et al., 2008) and show how the Markovian property is introduced so that the tools from hidden Markov models and state-space models can be leveraged.

2.2.2 Diffusion processes

A diffusion process is a stochastic model widely used to describe the random movement of particles over time. They provide a mathematical framework for characterising the spreading and movement of particles within an environment influenced by both deterministic forces and random fluctuations (Ito and McKean, 1967). Recall the simple random walk model introduced in the previous section through Eqn. 2.4 and we extend it now to multiple dimensions:

$$\boldsymbol{x}(t_i) = \boldsymbol{x}(t_{i-1}) + \boldsymbol{\varepsilon}(t_i), \tag{2.7}$$

where $\boldsymbol{x}(t_i)$ represents the location of the "walker" at time t_i and $\boldsymbol{\varepsilon}(t_i) \sim N(0, \Delta_i \mathbf{I})$, where $\Delta_i = t_i - t_{i-1}$.

Alternatively, we can write Eqn. 2.7 as a sum of step lengths beginning at the origin $\boldsymbol{x}(t_0) = \mathbf{0}$, with $t_0 = 0$ (Hooten et al., 2017a):

$$\begin{aligned} \boldsymbol{x}(t_i) &= \sum_{j=1}^{i} \boldsymbol{x}(t_j) - \boldsymbol{x}(t_{j-1}) \\ &= \sum_{j=1}^{i} \boldsymbol{\varepsilon}(t_j). \end{aligned} \tag{2.8}$$

Then, as the time difference approaches 0, the resulting model will be in continuous time:

$$\boldsymbol{x}(t_i) = \lim_{\Delta t \to 0} \sum_{j=1}^{i} \boldsymbol{\varepsilon}(t_j), \tag{2.9}$$

and for all t, the resulting sequence is the Brownian motion (Hooten et al., 2017a), which is an example of diffusion process. Using the notation used in stochastic calculus, the Brownian motion is defined as:

$$\mathbf{x}(t) = \int_0^t d\mathbf{x}(\tau), \tag{2.10}$$

where $d\mathbf{x}(t) = \lim_{\Delta t \to 0} \mathbf{x}(t) - \mathbf{x}(t - \Delta t)$.

In the context of animal movement, diffusion processes offer valuable insights into the behaviour and ecological dynamics of species. By fitting diffusion models to observed movement trajectories, it is possible to estimate important parameters like diffusion coefficients and drift rates. These parameters are key to an understanding of the underlying movement patterns, habitat preferences, and interactions between animals and their environment. Through diffusion processes it is possible to differentiate between various movement strategies, such as searching for resources or exploring new areas, which can have implications for population dynamics, conservation efforts and spatial management, thus contributing to a more comprehensive understanding of species' movement behaviours and their ecological roles.

One such example of diffusion process that is more complex than Brownian motion is the Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein, 1930). The OU process is mean-reverting, that is, the particle shows tendency to return to a point of attraction μ. The equation for a one-dimensional OU process is the following:

$$dX(t) = \beta(\mu - X(t))dt + \sigma dW(t). \tag{2.11}$$

Here, $dX(t)$ represents the change in animal position over a short time interval dt, μ is the center of attraction, σ quantifies the intensity of random movements represented by $dW(t)$, where $dW(t)$ is the Weiner process, and β governs the strength of attraction towards μ (Uhlenbeck and Ornstein, 1930). In the context of animal movement, the OU process offers insights into the interplay between deterministic behaviours and unpredictable environmental factors. For instance, consider an animal that has a favoured location it tends to return to due to resource availability or safety. At the same time, the animal's movement is influenced by random factors like wind or momentary behaviour shifts, which is accounted for by $dW(t)$. A modified version of this model will be employed in Chapter 5 to model the movement dynamics of a flock of sheep.

Other diffusion processes are employed in the context of animal movement, like the Langevin equation (Coffey and Kalmykov, 2012). The equation is expressed as:

$$m\frac{d^2x}{dt^2} = -\gamma\frac{dx}{dt} + \sqrt{2k_BT\gamma}\xi(t) + F(x) \tag{2.12}$$

where m is the mass of the particle, x represents its position, t is time, γ is the friction co-

efficient, k_B is the Boltzmann constant, T is the temperature, $\xi(t)$ is a Gaussian white noise term representing random fluctuations, and $F(x)$ is a potential force. The Langevin equation introduces more flexibility as the process is not constrained to be mean-reverting. This model will not be employed in this work.

2.2.3 Individual-based modelling

Individual-based modeling (IBM), also known as agent-based modelling, is a dynamic approach used to simulate complex systems by focusing on the behaviours and interactions of individual entities within the system. Unlike traditional aggregate-level models, IBM seeks to capture the intricate interplay between autonomous agents and their environment, allowing for a deeper understanding of emergent phenomena. By representing each individual as a distinct entity with its own set of characteristics, rules, and decision-making processes, IBM offers a powerful tool to explore the complex dynamics that arise from simple local interactions. Different IBMs therefore arise from different interaction rules. The Czirok model (Czirók et al., 1999) is a simple one-dimensional self-propelled particle (SPP) model that was used to simulate the motion of a group of locusts confined in a ring-shaped arena during an experiment. This model is analysed further in Chapter 3 so we defer the details. The two-dimensional extension of this model is known as the Vicsek model (Vicsek et al., 1995) and it was originally used to describe flocking in a noisy environment. These two models define an interaction radius within which individuals adjust their alignment according to the group alignment. The Vicsek model is described by the following equations:

$$\begin{aligned} \Theta_i(t+\Delta t) &= \langle \Theta_j \rangle_{|r_i - r_j| < r} + \eta_i(t) \\ \mathbf{r}_i(t+\Delta t) &= \mathbf{r}_i(t) + v\Delta t \begin{bmatrix} cos(\Theta_i(t)) \\ sin(\Theta_i(t)) \end{bmatrix}, \end{aligned} \tag{2.13}$$

where $\mathbf{r}_i(t)$ and $\Theta_i(t)$ are respectively the position and angle of particle i at time t, $\langle \Theta_j \rangle_{|r_i - r_j|}$ denotes the average direction of the velocities of particles (including particle i) within a circle of a given distance r, $|r_i - r_j| < r$, surrounding particle i, and $\eta_i(t)$ is the noise uniformly distributed on $[-\pi, \pi]$.

Another example of IBM is found in Kerman et al. (2012). The model describes constant-speed, two-dimensional movement within a periodic domain and three interaction zones are explicitly defined: repulsion, orientation (or alignment) and attraction. The neighbours in these zones are determined by the following equations (Kerman et al., 2012):

$$\begin{aligned} n_i^r &= \{j : ||\mathbf{r}_i - \mathbf{r}_j|| < R_r, a_{ij} = 1\}, \\ n_i^o &= \{j : ||\mathbf{r}_i - \mathbf{r}_j|| < R_o, a_{ij} = 1\}, \\ n_i^a &= \{j : ||\mathbf{r}_i - \mathbf{r}_j|| < R_a, a_{ij} = 1\}, \end{aligned} \tag{2.14}$$

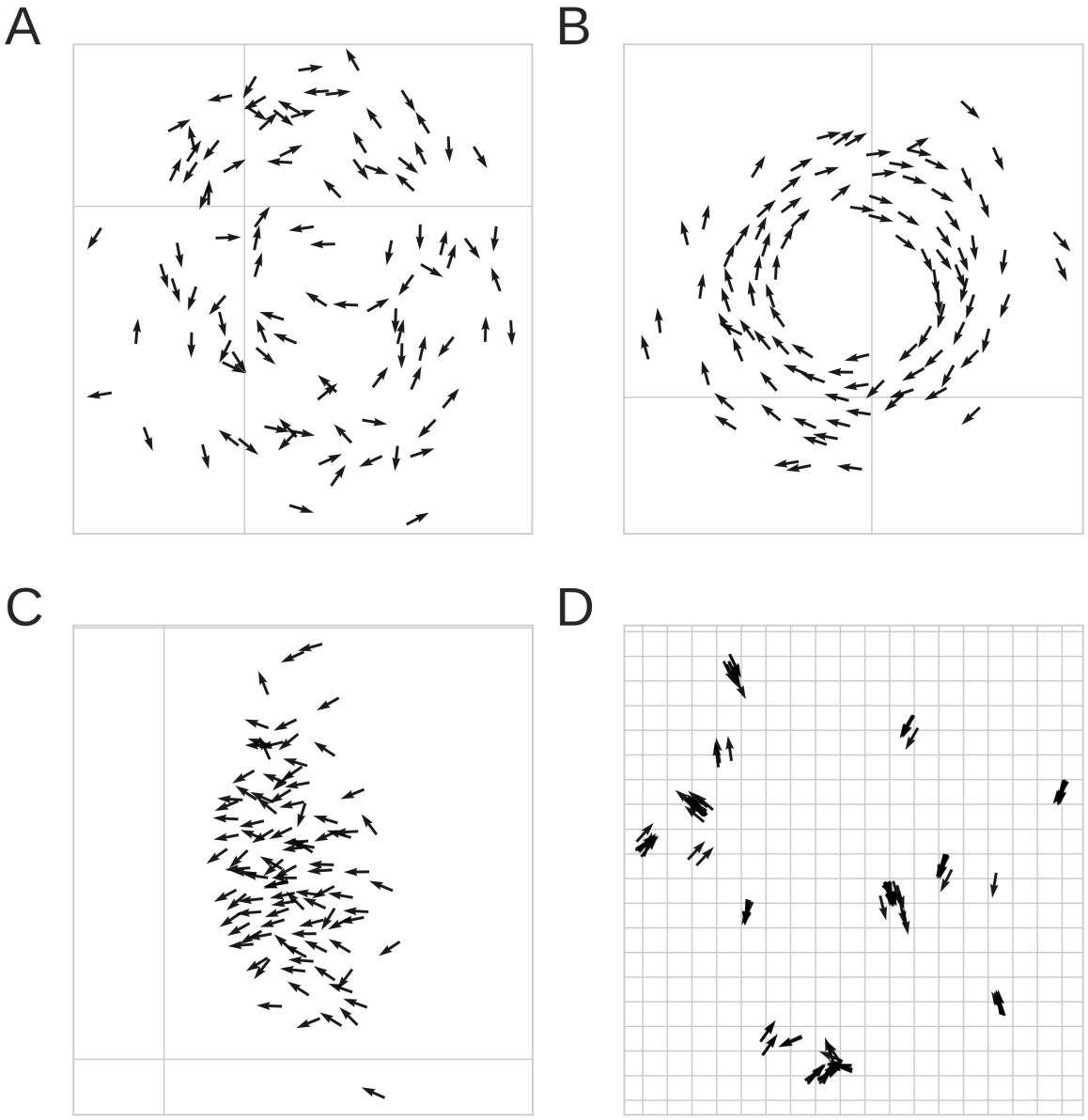


Figure 2.1: Illustration of the different behaviours arising from different combinations of the zonal interaction model parameters (Kerman et al., 2012), for different parameter combinations of the orientation radius, R_o, and the attraction radius, R_a. Panel (A): swarm ($R_o = 0, R_a = 15$). Panel (B): toroidal ($R_o = 3, R_a = 15$). Panel (C): dynamic parallel ($R_o = 10, R_a = 10$). Panel (D): concentrated parallel ($R_o = 20, R_a = 10$). The plot was taken from Gaskell et al. (2023)

where R_r, R_o and R_a are respectively the radius of repulsion, orientation and alignment, $A(t) = a_{ij(t)}$ is the sensory adjacency matrix (Kerman et al., 2012), where $a_{ij}(t) = 1$ means that agent j is visible to agent i at time t, and $\mathbf{r}_i$ is the location of agent i. The quantities n_i^r, n_o^i and n_a^i are the sets of agent i's neighbours in the regions of, respectively, repulsion, orientation and attraction (Kerman et al., 2012). The model allows for overlapping regions of repulsion, orientation and attraction.

The agent will then interact with the individuals within these three interaction zones differently, and depending on the radius of the orientation zone, the model manifests both a flock and a torus attractor (Kerman et al., 2012). This is illustrated in Fig. 2.1.

2.2.4 State-space models

State-space models (SSMs) are a wide class of time series models (Durbin and Koopman, 2012) employed in the context of estimating the dynamics of a phenomenon that cannot be observed directly. State-space models can handle measurement error by defining two processes; the first process is the observation process and is described by the observation equation which relates the observations $\mathbf{y}_t$ to the unknown true values $\mathbf{z}_t$:

$$\mathbf{y}_t = h(\mathbf{z}_t, \mathbf{v}_t), \tag{2.15}$$

where $h()$ is any function (linear or non-linear) and $\mathbf{v}_t$ is the observation noise. The second equation is the system equation and defines the update rules of the unknown states, that is, the noise-free locations:

$$\mathbf{z}_t = f(\mathbf{z}_{t-1}, \mathbf{w}_t, \mathbf{u}_t), \tag{2.16}$$

where $f()$ can be any function, $\mathbf{w}_t$ is the process noise and $\mathbf{u}_t$ is the control input, which represents an external signal or command that influences the system's behaviour.

A widely used state-space model in the movement ecology literature is the Kalman filter (Kalman, 1960). The Kalman filter was originally employed by control engineers and physical scientists in areas such as signal processing in aerospace tracking and underwater sonar. As an example, the Kalman filter could be employed in the estimation of the position and speed of a satellite ($\mathbf{z}_t$) given our distance and relative angle to the satellite ($\mathbf{y}_t$). The Kalman filter is based on two main assumptions: the state transitions (system equations) as well as the observation equation are assumed to be linear, and the observation noise and the system noise are Gaussian (Meinhold and Singpurwalla, 1983). This means that the equations of the Kalman filter become:

$$\mathbf{y}_t = \mathbf{F}_t \mathbf{z}_t + \mathbf{v}_t, \tag{2.17}$$

where the quantity $\mathbf{F}_t$ is assumed to be known and represents the observation model (how the latent state is mapped to the observations) and $\mathbf{v}_t \sim N(0, \mathbf{V}_t)$ is the observation noise, whereas

the second equation:

$$\mathbf{z}_t = \mathbf{G}_t \mathbf{z}_{t-1} + \mathbf{w}_t, \tag{2.18}$$

where $\mathbf{G}_t$ is the transition density matrix (assumed to be known) and $\mathbf{w}_t \sim N(0, \mathbf{W}_t)$. Note that contrarily to Eqn. 2.16, we have not included the control input $\mathbf{u}_t$, following the work of Meinhold and Singpurwalla (1983).

The Kalman filter may be easily understood if it is seen as inference about $\mathbf{z}_t$ using the Bayes' rule (Meinhold and Singpurwalla, 1983). For the sake of simplicity in the illustration of the algorithm, let's take into consideration scalar z_t and observed data $\mathbf{y}_t = \{y_1, \ldots, y_t\}$. Then, the Bayes' rule states:

$$p(z_t|\mathbf{y}_t) \propto p(y_t|z_t, \mathbf{y}_{t-1}) \times p(z_t|\mathbf{y}_{t-1}), \tag{2.19}$$

where $p(A|B)$ indicates the probability of event A conditional on the fact that event B has occurred. We defer the details of the Bayes' rule and Bayesian inference until Section 2.3.

Inference about z_t is carried out through a recursive algorithm. To illustrate the algorithm, we will focus on time point $t-1$, for $t = 1, 2, \ldots$, and the observed data until time $t-1$, $\mathbf{y}_{t-1} = \{y_{t-1}, \ldots, y_1\}$ (Meinhold and Singpurwalla, 1983). At time $t-1$, the distribution of z_{t-1} will follow a normal distribution with expectation and variance, respectively, $\hat{z}_{t-1}$ and $\mathbf{\Sigma}_{t-1}$:

$$(z_{t-1}|\mathbf{y}_{t-1}) \sim N(\hat{z}_{t-1}, \mathbf{\Sigma}_{t-1}). \tag{2.20}$$

The recursive scheme begins by choosing initial conditions for $\hat{z}_0$ and $\mathbf{\Sigma}_0$. Then at time t, we update the values of the expectation and variance of the state in two stages, prior to observing y_t and after observing y_t.

Before observing y_t, our best knowledge of z_t is given by Eqn. 2.16; given that we know the value of z_{t-1} from Eqn. 2.20, we conclude that, prior to observing y_t, our state knowledge is (Meinhold and Singpurwalla, 1983):

$$(z_t|\mathbf{y}_{t-1}) \sim N(\mathbf{G}_t \hat{z}_{t-1}, \mathbf{G}_t \mathbf{\Sigma}_{t-1} \mathbf{G}_t^{'} + \mathbf{W}_t). \tag{2.21}$$

Once we observe y_t, we want to calculate the probability distribution in Eqn. 2.19. This is however possible only if we know the distribution $p(y_t|z_t, \mathbf{y}_{t-1})$; to calculate it, we shall proceed as follows. Let e_t be the prediction error from time point $t-1$:

$$\begin{aligned} e_t = y_t - \hat{y}_t &= y_t - \mathbf{F}_t \mathbf{G}_t \hat{z}_{t-1} \\ &= \mathbf{F}_t (z_t - \mathbf{G}_t \hat{z}_{t-1}) + v_t, \end{aligned} \tag{2.22}$$

where we have used the relation described in the observation equation, Eqn. 2.17. Note that upon observation of y_t, e_t becomes known. Hence, we can rewrite Eqn. 2.19 as follows

(Meinhold and Singpurwalla, 1983):

$$p(z_t|e_t, \mathbf{y}_{t-1}) \propto p(e_t|z_t, \mathbf{y}_{t-1}) \times p(z_t|\mathbf{Y}_{t-1}). \tag{2.23}$$

Since $v_t \sim N(0, \mathbf{V}_t)$, it follows that

$$(e_t|z_t, \mathbf{y}_{t-1}) \sim N(\mathbf{F}_t(z_t - \mathbf{G}_t\hat{z}_{t-1}), \mathbf{V}_t) \tag{2.24}$$

and that we can use Bayes' theorem to obtain (Meinhold and Singpurwalla, 1983):

$$p(z_t|y_t, \mathbf{y}_{t-1}) = \frac{p(e_t|z_t, \mathbf{y}_{t-1}) \times p(z_t|\mathbf{Y}_{t-1})}{\int_{z_t} p(e_t|z_t, \mathbf{y}_{t-1})dz_t}. \tag{2.25}$$

We conclude this section by stating that hidden Markov models are a sublcass of state-space models whereby the state space is assumed to be discrete. Since HMMs have become a popular tool amongst the movement ecologists, and because HMMs will play an important role in Chapters 4 and 5, we dedicate a more thorough review of the subject in Section 2.6.

2.3 Bayesian statistics

Bayesian statistics is a formulation of statistics based on probability distributions. The main idea of Bayesian statistics revolves around the application of Bayes' rule. Consider the scenario where we have some observed data D and we have a model that describes the data which depends on some model parameters $\boldsymbol{\theta}$. The end goal is to leverage the data to acquire knowledge of the model parameters and this is done through the application of Bayes' rule:

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.26}$$

We define the following quantities:

1) $p(D|\boldsymbol{\theta})$ - this is the *likelihood function*. Assuming that $\boldsymbol{\theta}$ is true, the likelihood expresses how likely $\boldsymbol{\theta}$ is to have generated the observed data D.

2) $p(\boldsymbol{\theta})$ - this is the *prior distribution*. As the name suggests, it reflects our prior beliefs about the parameters $\boldsymbol{\theta}$.

3) $p(\boldsymbol{\theta}|D)$ - this is the *posterior distribution*. It describes how confident or certain we are about the values of $\boldsymbol{\theta}$ given the observed data D. In Bayesian statistics, calculating the posterior distribution is the end goal.

4) $p(D) = \int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ - this is called the *evidence*, or *marginal likelihood*. It is a normalising constant that ensures the posterior distribution is a proper probability distribution. If the probability distributions are discrete, then integration is replaced by summation.

The power of Bayesian statistics is that whenever we have access to more data, our "beliefs" are updated through the likelihood function and we become more confident on the true values of $\boldsymbol{\theta}$. The Bayesian inference framework can also be used to make predictions on unseen data, specifically we can calculate the probability distribution of possible unobserved values conditional on the observed values, known as the posterior predictive distribution:

$$p(\tilde{D}|D) = \int p(\tilde{D}, \boldsymbol{\theta}|D)d\boldsymbol{\theta} = \int p(\tilde{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta}, \tag{2.27}$$

where $\tilde{D}$ is the new unobserved data. As we can see, making predictions on observed data is based on the posterior distribution.

The evidence, however, may become intractable when considering high-dimensional distributions. In that case, the most commonly employed techniques that are used to get estimates of the posterior distribution (or more generally any distribution that is hard to sample from directly) are the Markov chain Monte Carlo (MCMC) techniques and variational inference (VI).

2.3.1 Markov chain Monte Carlo techniques

Markov chain Monte Carlo (MCMC) techniques are powerful computational methods used for generating samples from complex probability distributions that are challenging to sample directly. The idea is to generate samples by constructing a Markov chain on the state space $\mathscr{X}$ whose stationary distribution is the target distribution $\pi(\mathbf{x})$ that we want to sample from, for $\mathbf{x} \in \mathscr{X}$. Essentially, this means that we define a random walk on the state space $\mathscr{X}$ (Murphy, 2012). Applying this to the Bayesian inference problem, the target distribution of interest will be the posterior distribution $p(\boldsymbol{\theta}|D)$, whereas the state space will be the parameter space $\boldsymbol{\Theta} \ni \boldsymbol{\theta}$.

To ensure that the Markov chain has as its stationary distribution the target distribution, the *detailed balance* condition must be met. Consider a state space $\mathscr{X}$ and let $\rho(a,b) = p(X_{n+1} = b|X_n = a)$, with $a,b \in \mathscr{X}$, be the transition density of the Markov chain, with X_{n+1} and X_n being two successive realisation of the Markov chain. Then detailed balance is satisfied if there exists a probability distribution κ such that

$$\rho(a,b)\kappa(a) = \rho(b,a)\kappa(b), \ \forall a,b \in \mathscr{X}. \tag{2.28}$$

If detailed balance is satisfied, then the probability distribution κ is the stationary distribution

(Kelly and Yudovina, 2012). This condition will ensure that the constructed Markov chain will have as stationary distribution the target distribution of interest. We are constructing a Markov chain that will reach its stationary distribution, however we don't know how quickly it will converge to it, hence it is common to discard the influence of the starting values of the chain. This is called the *burn-in* phase. Furthermore, since successive Markov samples are correlated, depending on the applications it is sometimes useful to keep every nth sample so that they are independent.

We will see how we can construct such Markov chains in two of the most commonly employed MCMC algorithms, the Gibbs sampling algorithm and the Metropolis-Hastings algorithm.

Gibbs sampling

Gibbs sampling is an MCMC technique used to sample from complex multivariate joint probability distributions where the conditional distributions of each variable is known (Robert and Casella, 2005). Consider a joint distribution $p(x_1, x_2, \ldots, x_n)$. Gibbs sampling iteratively generates samples from the conditional distributions $p(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Starting with an initial estimate $\{x_1^{(0)}, \ldots, x_n^{(0)}\}$, the algorithm updates each variable sequentially according to:

$$\begin{aligned} x_1^{(t+1)} &\sim p(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_n^{(t)}) \\ x_2^{(t+1)} &\sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \ldots, x_n^{(t)}) \\ &\vdots \\ x_n^{(t+1)} &\sim p(x_n|x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{n-1}^{(t+1)}) \end{aligned} \tag{2.29}$$

where t represents the iteration number. As t approaches infinity, the samples $\{x_1^{(t)}, \ldots, x_n^{(t)}\}$ converge to the true joint distribution $p(x_1, \ldots, x_n)$. We can show this by making the following definition.

Let $x \sim_j y$ if $x_i \neq y_i$, $\forall i \neq j$ and g be the desired target distribution. Then the transition probabilities are defined as (Murphy, 2012):

$$\rho(x,y) = \begin{cases} \frac{1}{n} \frac{g(y)}{\sum_{z \sim_j x} g(z)}, & x \sim_j y \\ 0, & \text{otherwise.} \end{cases} \tag{2.30}$$

Hence,

$$g(x)\rho(x,y) = \frac{1}{n} \frac{g(x)g(y)}{\sum_{z \sim_j x} g(z)} = \frac{1}{n} \frac{g(y)g(x)}{\sum_{z \sim_j x} g(z)} = g(y)\rho(y,x), \tag{2.31}$$

that is, detailed balance is satisfied and therefore g is the stationary distribution.

Metropolis-Hastings

Another widely employed MCMC algorithm is the Metropolis-Hastings (MH) algorithm (Hastings, 1970), which will be extensively employed in Chapters 3 and 5. Given a target distribution $\pi(x) = \frac{1}{Z}\tilde{\pi}(x)$, that is impossible to sample from directly due to the intractability of the normalisation constant Z, in the MH algorithm we proceed by defining a proposal distribution, that is a distribution through which we can generate candidate samples to construct the Markov chain that has as its stationary distribution $\pi(x)$. Let $Q(x'|x)$ be the proposal distribution that expresses the probability to generate sample x' from a current sample x. At each iteration of the algorithm, we first propose a new sample through Q, then this sample is either accepted or rejected based on an acceptance probability defined as follows (Hastings, 1970):

$$\alpha(x',x) = min\left\{1, \frac{\pi(x')}{\pi(x)}\frac{Q(x|x')}{Q(x'|x)}\right\} = min\left\{1, \frac{\tilde{\pi}(x')}{\tilde{\pi}(x)}\frac{Q(x|x')}{Q(x'|x)}\right\}, \tag{2.32}$$

where $\alpha(x',x)$ is the probability of accepting x' and we can see that we avoid computing the normalising constant Z since it cancels out in the acceptance ratio. The ratio of proposal probabilities is called Hastings ratio. For symmetric proposal Q, the Hastings ratio is 1 and the acceptance probability simplifies to

$$min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}. \tag{2.33}$$

The Hastings ratio will play a crucial role in Chapter 4 and Chapter 5.

The reason why a Markov chain built through such an acceptance probability will converge in distribution to π is because $\alpha(x',x)$ is constructed in such a way that it satisfies detailed balance. To see this, let's define the transition probability matrix for this chain as follows (Hastings, 1970, Murphy, 2012):

$$P_{xx'} = p(x'|x) = \begin{cases} Q(x'|x)\alpha(x',x), \text{ if } x' \neq x, \\ Q(x'|x) + \sum_{x' \neq x} Q(x'|x)(1-\alpha(x',x)), \text{ otherwise.} \end{cases} \tag{2.34}$$

Assume without loss of generality that $\alpha(x',x) = 1$ and $\alpha(x,x') = \frac{\tilde{\pi}(x)Q(x'|x)}{\tilde{\pi}(x')Q(x|x')}$. Then (Murphy, 2012):

$$\tilde{\pi}(x')Q(x|x')\alpha(x,x') = \tilde{\pi}(x)Q(x'|x)\alpha(x',x) \tag{2.35}$$

which implies

$$\pi(x')P_{x'x} = \pi(x)P_{xx'}, \tag{2.36}$$

that is, detailed balance is satisfied. It can be shown that the Gibbs sampling algorithm is a special case of the Metropolis-Hastings algorithm, whereby all samples are accepted (Murphy, 2012).

Reversible jump MCMC

MCMC techniques are often used for the inference on mixture models. Inference on mixture models can be broadly categorised into two groups: one group tackles inference with fixed k, whereas the other group treats k as unknown. The latter is more mathematically involved. Concerning this matter, pioneering study on the topic is found in Richardson and Green (1997) and Robert et al. (2000). The theoretical framework from which they developed their work is based on that from Green (1995). There, inference on HMMs parameters employs reversible jump Markov chain Monte Carlo (RJMCMC) routines. This method was developed to overcome limitations of standard MH scheme and enable the sampler to "jump" to parameter subspaces of different dimensions, all respecting detailed balance. Generally, given a vector of parameters $\boldsymbol{\theta}$, a target posterior distribution π, a countable family of move types $m = 1, 2, ...$ and an arbitrary proposal distribution $q_m(x, x')$, a proposed sample $\boldsymbol{\theta}'$ is accepted with probability

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = min\left\{1, \frac{\pi(\boldsymbol{\theta}')q_m(\boldsymbol{\theta}', \boldsymbol{\theta})}{\pi(\boldsymbol{\theta})q_m(\boldsymbol{\theta}, \boldsymbol{\theta}')}\right\}. \tag{2.37}$$

If the move does not change the dimension of the parameter then the above expression coincides with a standard MH acceptance probability. If the move is dimension-changing, then suppose that $\boldsymbol{\theta}'$ lies in a higher-dimensional space. As suggested in Richardson and Green (1997), the move is usually implemented by drawing a vector of continuous random variables $\boldsymbol{u}$, independent of $\boldsymbol{\theta}$, and setting $\boldsymbol{\theta}'$ by using an invertible deterministic function $f(\boldsymbol{\theta}, \boldsymbol{u})$. The reverse of the move then is accomplished by using the inverse transformation, so that the proposal is deterministic. The acceptance probability then becomes

$$min\left\{1, \frac{p(\boldsymbol{\theta}'|y)r_m(\boldsymbol{\theta}')}{p(\boldsymbol{\theta}|y)r_m(\boldsymbol{\theta})q(\boldsymbol{u})}\left|\frac{\partial d\boldsymbol{\theta}'}{\partial d(\boldsymbol{\theta}, \boldsymbol{u})}\right|\right\}, \tag{2.38}$$

where $r_m(\boldsymbol{\theta})$ is the probability of choosing move type m when in $\boldsymbol{\theta}$, and $q(\boldsymbol{u})$ is the density function of u. The last term is the Jacobian associated with the change of variable from $(\boldsymbol{\theta}, \boldsymbol{u})$ to $\boldsymbol{\theta}'$.

There are four standard dimension-changing moves: *split* and *combine* moves and *birth* and *death* moves. Key is their reversibility: such moves are constructed in tandem, that is, in a reversible pair. Without going into the details, the combine move combines two adjacent components into a new component, reducing the total number of components by 1; on the contrary, from a randomly chosen components the split move creates two new components, increasing the number of components by 1. The birth and death move are involved with empty components, i.e. components to which no observation is associated. As the names suggest, the birth move creates a new empty component whereas the death move deletes an existing empty component. At every step of the sampler, split-combine and birth-death are

all implemented with probability b_k, $d_k = 1 - b_k$, with $b_{k_{max}} = 0$, $d_0 = 0$. The parameter k_{max}, chosen a priori, is fixed and represents the total number of components. The mathematically involved part is the derivation of the acceptance probability for these moves. We will omit the details, but they can be found in Green (1995) and Richardson and Green (1997). In Robert et al. (2000) an application to HMMs inference is given.

In the future chapter we will perform inference on a hidden semi-Markov model of animal movement. Given that the number of states will be kept fixed, we will not need the RJMCMC framework.

2.3.2 Variational inference

Variational inference (VI) is a different approach to MCMC techniques and is used to get approximations to an intractable probability distribution. In the context of Bayesian inference, it can be exploited to approximate posterior distributions that are intractable due to the intractability of the marginal likelihood. The idea behind variational inference is to introduce a variational distribution q from a family of tractable parameterised distributions that can approximate the true posterior. The implementation of VI revolves around finding the best such variational distribution.

Let's start from the same settings as in Section 2.3. Suppose that $p(\boldsymbol{\theta}|D)$ is the intractable posterior distribution. Let $q(\boldsymbol{\theta})$ be a distribution taken from a family of tractable, parameterised distributions $\mathscr{F}$. In order to find the best approximation to $p(\boldsymbol{\theta}|D)$, we need $q(\boldsymbol{\theta})$ to be "close" to $p(\boldsymbol{\theta}|D)$. To do so, we will use the Kullback-Leibler (KL) divergence, which is a measure of similarity between two distributions. For the distributions $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|D)$, the KL divergence is (Kullback and Leibler, 1951):

$$KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|D)) = \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|D)}\right). \tag{2.39}$$

The KL divergence is asymmetric and non-negative, with equality to 0 if and only if $q(\theta) = p(\theta|D)$.

Then, the best distribution $q(\boldsymbol{\theta})$ from the family $\mathscr{F}$ is found by solving the following optimisation task (Murphy, 2012):

$$\begin{aligned} q^*(\boldsymbol{\theta}) &= \underset{q(\boldsymbol{\theta})\in\mathscr{F}}{argmin}\ KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|D)) \\ &= \underset{q(\boldsymbol{\theta})\in\mathscr{F}}{argmin}\ \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|D)}\right). \end{aligned}$$

However, this form is still unpractical as the intractable evidence appears in the denominator of Eqn. 2.40 via the posterior distribution that we want to approximate in the first place. To overcome this issue, we can consider the unnormalised posterior distribution $\tilde{p}(\boldsymbol{\theta}|D) =$

$p(\boldsymbol{\theta}|D)p(D)$, where $p(D)$ is the evidence. Then we can optimise the following objective function (Murphy, 2012):

$$\begin{aligned} KL(q(\boldsymbol{\theta})||p(\tilde{\boldsymbol{\theta}}|D)) &= \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) ln\left(\frac{q(\boldsymbol{\theta})}{\tilde{p}(\boldsymbol{\theta}|D)}\right) \\ &= \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) ln\left(\frac{q(\boldsymbol{\theta})}{p(D)p(\boldsymbol{\theta}|D)}\right) \\ &= \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|D)}\right) - ln(p(D)) \\ &= KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})|D)) - ln(p(D)). \end{aligned}$$

Then by minimising the objective function in Eqn. 2.40, the negative log marginal likelihood will cancel out since it is constant with respect to $\boldsymbol{\theta}$. Alternatively, we can turn this into a maximisation task by optimising the negative objective function of Eqn. 2.40 (Murphy, 2012):

$$q^*(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta})\in\mathscr{F}}{argmax}\ ln(p(D)) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})|D)). \tag{2.40}$$

Because the KL divergence is nonnegative, we can see that

$$ln(p(D)) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})|D)) \leq ln(p(D)), \tag{2.41}$$

and this quantity is a lower bound for the log marginal likelihood. By pushing the lower bound to the log marginal likelihood, we can find the best approximation to the posterior distribution $p(\boldsymbol{\theta}|D)$.

2.4 Classical statistics

The classical, or frequentist, approach to statistics is based on the fundamental concept of probability as the long-term relative frequency of events in repeated, identical experiments. That is, the statement "the probability of observing the number 1 after rolling a die is $\frac{1}{6}$" means that after rolling the die infinitely many times we will observe the number 1 $\frac{1}{6}$ of the times. In this framework, a parameter estimate $\hat{\boldsymbol{\theta}}$ is computed by applying an estimator δ to some data D: $\hat{\boldsymbol{\theta}} = \delta(D)$ (Murphy, 2012). In this statistical paradigm, the parameter is viewed as fixed and the data as random. The uncertainty in the parameter estimate can then be measured by computing the sampling distribution of the estimator, which is the distribution that an estimator has when applied to multiple data sets sampled from the true but unknown distribution (more details on this can be found in Murphy (2012), Young (2005)). We will now describe two powerful methods that are used in the frequentist paradigm of statistics, the Maximum Likelihood Estimation (MLE) and the Expectation-Maximisation (EM) algorithm.

2.4.1 Maximum likelihood estimation

One of the most commonly employed methods for the inference of parameters $\boldsymbol{\theta}$ in the context of frequentist statistics is the Maximum Likelihood Estimation method. This method is based on the concept of likelihood (which also plays an important role in the Bayesian paradigm of statistics, Section 2.3). Given a set of data $\mathbf{Y} = \{y_1, \ldots, y_N\}$, where N is the size of the dataset, and a probability distribution of the data given the parameters $p(D|\boldsymbol{\theta})$, then the likelihood is defined as (Hastie et al., 2009):

$$L(\boldsymbol{\theta}, \mathbf{Y}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{\theta}), \tag{2.42}$$

where the underlying assumption is that the data $\{y_1, \ldots, y_N$ are independent and identically distributed.

The MLE then seeks the parameters that maximise the likelihood, meaning that the MLE ultimately amounts to an optimisation routine:

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{argmax}\, L(\boldsymbol{\theta}, \mathbf{Y}), \tag{2.43}$$

where $\boldsymbol{\Theta}$ is the parameter space.

2.4.2 Expectation-Maximisation algorithm

For many models, computing MLE of the parameters can be straightforward, provided that the data is "complete", meaning the data is fully observable. This however may not be the case in the presence of missing data or latent variable models, such as state-space models or hidden Markov models, as we shall see in Section 2.6. In these cases, we can employ the Expectation-Maximisation algorithm. The EM algorithm is a two-step iterative algorithm that continuously iterates between the Expectation step (E-step) and the Maximisation step (M-step). We will now give the details.

As before, let $\mathbf{Y}$ represent the observed data and let $\mathbf{Z}$ be the missing or impossible to observe data. For example, in the study of animal movement $\mathbf{Y}$ may represent the location of an animal and $\mathbf{Z}$ may represent some missing fixes due to some GPS error. In another example, $\mathbf{Z}$ may represent the "state" in which the animal is at the time of observation, meaning what behavioural activity (for example, foraging or resting) was the animal engaged with when its position $\mathbf{Y}$ was recorded. We shall denote the set $\{\mathbf{Y}, \mathbf{Z}\}$ the *complete data* and the set $\mathbf{Y}$ the *incomplete data*.

Consider the log-likelihood of the model:

$$ln\ p(\mathbf{Y}|\boldsymbol{\theta}) = ln \sum_{\mathbf{Z}} p(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}), \tag{2.44}$$

that is, if we were given knowledge of the complete data set we could use it to find the likelihood of the model. Knowledge of the missing or latent data is contained in the following distribution:

$$p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}). \tag{2.45}$$

Having set the preliminaries, we proceed as follows. Consider a function on the latent variables $q(\mathbf{Z})$ (this is a very similar approach to the VI method, in fact, the EM algorithm is a special case of the VI framework), then the log-likelihood can be decomposed in the following way (Bishop, 2006):

$$ln\ p(\mathbf{Y}|\boldsymbol{\theta}) = \mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) + KL(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})), \tag{2.46}$$

where

$$\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) ln\left(\frac{p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}\right) \tag{2.47}$$

is a lower bound for $p(\mathbf{Y}|\boldsymbol{\theta})$ (Bishop, 2006) and

$$KL(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})) = \sum_{\mathbf{Z}} q(\mathbf{Z}) ln\left(\frac{p(\boldsymbol{Z}|\boldsymbol{Y},\boldsymbol{\theta})}{q(\mathbf{Z})}\right) \tag{2.48}$$

is the Kullback-Leibler divergence between $q(\mathbf{Z})$ and the posterior distribution $p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})$.

To illustrate the EM algorithm, suppose that the current parameter is $\boldsymbol{\theta}^-$. In the E-step, the lower bound $\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}^-)$ is maximised with respect to $q(\mathbf{Z})$ while holding $\boldsymbol{\theta}^-$ fixed. This is obtained if the KL divergence vanishes (Bishop, 2006), which occurs if

$$q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-). \tag{2.49}$$

This means that during the E-step the lower bound is

$$\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-) ln\ p(\mathbf{Y},\mathbf{Z}|\boldsymbol{\theta}) + const., \tag{2.50}$$

which is the *expectation* of the complete data log-likelihood with respect to the distribution of $\mathbf{Z}$ conditioned on the incomplete data and the current parameter $\boldsymbol{\theta}^-$.

In the M-step, the lower bound is then maximised with respect to $\boldsymbol{\theta}$ to give a new value $\boldsymbol{\theta}^+$, this time holding $q(\mathbf{Z})$ fixed. This will cause the lower bound to increase, which will in turn increase the log-likelihood. Because $q(\mathbf{Z})$ is held fixed during the M-step and because it is determined using $\boldsymbol{\theta}^-$, once the M-step is complete the distribution will not equal the new distribution $p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^+)$, thus the KL divergence will be non-zero (Bishop, 2006).

In some cases, however, the conditional expectation required in the E-step of the algorithm may be intractable and therefore it is replaced by some stochastic approximation (Nielsen, 2000) or Monte Carlo approximation (Levine and Casella, 2001). We will see in

Chapter 5 how we have introduced a Monte Carlo approximation of the conditional expectation for a model of animal movement.

2.5 Gaussian processes

Gaussian processes are stochastic processes $\{\mathbf{X}_t\}_{t\geq 0}$ for which any finite collection of random variables follows a multivariate normal (MVN) distribution. For $\{X_1,\ldots,X_N\} = (x_1,\ldots,x_N)$, we have

$$(x_1,\ldots,x_N) \sim N(\boldsymbol{m},\boldsymbol{\Sigma}). \tag{2.51}$$

Gaussian processes are therefore determined by a mean function $\mathbf{m}$ and a covariance matrix $\boldsymbol{\Sigma}$, which is often defined through a kernel function $\Sigma_{ij} = k(x_i,x_j)$, that is a function of the location of the random variables. The kernel function is a measure of similarity between the locations x_i and x_j (Murphy, 2012, Bishop, 2006). There exist different kernels that because of their properties may be more suitable for some applications than others. Below we review some of the most common kernels: the absolute exponential kernel, the radial basis function (RBF) kernel and the Matern family of kernels.

The absolute exponential kernel depends on two parameters and is defined as:

$$k(x_i,x_j) = \tau^2 exp\bigg(-\frac{||x_i - x_j||}{\vartheta}\bigg), \tag{2.52}$$

where the τ^2 and ϑ kernel hyperparameters represent the scale, or amplitude, of the process and its lengthscale and $||x_i - x_j||$ is the Euclidean distance between x_i and x_j.

The RBF kernel is a similar kernel and is defined as follows (Rasmussen and Williams, 2006):

$$k(x_i,x_j) = \tau^2 exp\bigg(-\frac{||x_i - x_j||^2}{\vartheta}\bigg), \tag{2.53}$$

where the kernel hyperparameters τ^2 and ϑ represent again the scale and lengthscale of the process. This kernel will be employed in Chapter 3.

The Matern kernel class, an extension of the RBF kernel, introduces a parameter ν determining the function's smoothness; lower ν values yield rougher approximations. As $\nu \to \infty$, the kernel is equivalent to the RBF kernel. For $\nu = \frac{1}{2}$, it matches the absolute exponential kernel. Other important values include $\nu = \frac{3}{2}$ (once differentiable functions) and $\nu = \frac{5}{2}$ (twice differentiable functions). The equation is

$$k(x_i,x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\bigg(\frac{\sqrt{2\nu}}{\vartheta}||x_i,x_j||\bigg)^{\nu} K_{\nu}\bigg(\frac{\sqrt{2\nu}}{\vartheta}||x_i,x_j||\bigg), \tag{2.54}$$

where $\Gamma(\cdot)$ is the gamma function and $K_{\nu}(\cdot)$ is a modified Bessel function.

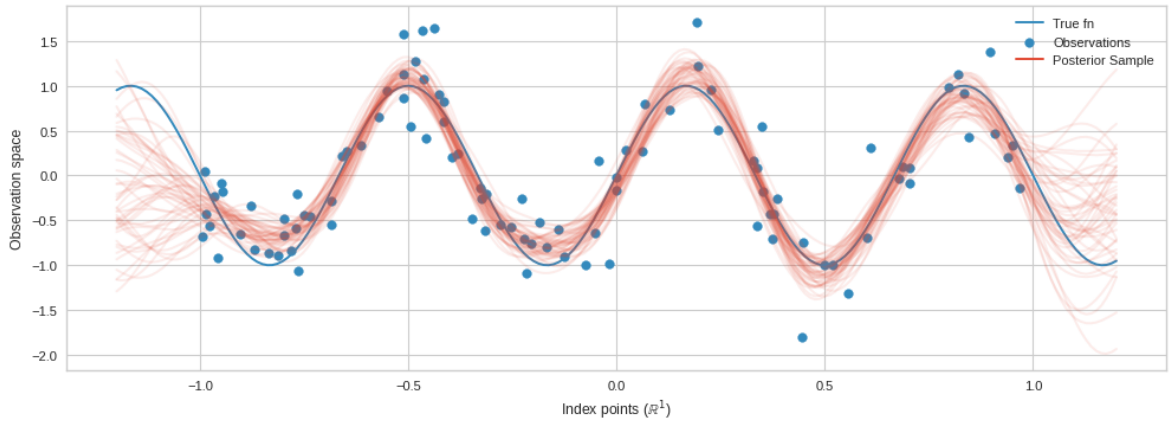


Figure 2.2: Example of GP regression on a simple 1-dimensional model. The observations were generated by a sine function with additive Gaussian noise.

2.5.1 Gaussian process regression

Gaussian processes are widely used in a regression task. The regression settings are the following. Given observed data $\mathbf{y} = \{y_1, \ldots, y_N\}$ and corresponding explanatory variables $\mathbf{x} = \{x_1, \ldots, x_N\}$, the data are assumed to be generated by an unknown function and additive Gaussian white noise:

$$y_i = f(x_i) + \varepsilon, \tag{2.55}$$

where $\varepsilon \sim N(0, \sigma^2)$ (Murphy, 2012). The latent function f is seen as a realisation of a GP, that is, we place a GP prior on the functions:

$$p(\mathbf{f}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}). \tag{2.56}$$

This is possible because we only need to evaluate the functions at an arbitrary, finite set of locations, in this case $\mathbf{x}$ (Murphy, 2012, Bishop, 2006).

Consider now some test locations $\mathbf{x}_* = \{\mathbf{x}_*^1, \ldots, \mathbf{x}_*^{N_*}\}$, that is, we want to predict the values of f at some new location: $\mathbf{f}_* = \{f(\mathbf{x}_*^1), \ldots, f(\mathbf{x}_*^{N_*})\}$. Then

$$p(\mathbf{f}_*|\mathbf{x}_*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(\boldsymbol{m}_*, \boldsymbol{\Sigma}_*), \tag{2.57}$$

where

$$\begin{aligned} \boldsymbol{m}_* &= \mathbf{K}(\mathbf{x}_*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \\ \boldsymbol{\Sigma}_* &= \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) \\ &\quad - \mathbf{K}(\mathbf{x}_*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{x}_*, \mathbf{x})^T \end{aligned} \tag{2.58}$$

Here $\mathbf{I}$ is the identity matrix and $\mathbf{K}(\mathbf{x}, \mathbf{x})$ is the covariance matrix defined at all pairs of train points (an $N \times N$ matrix), $\mathbf{K}(\mathbf{x}_*, \mathbf{x}_*)$ is the covariance functions specified at all pairs of test

points (an $N_* \times N_*$ matrix) and $\mathbf{K}(\mathbf{x}_*, \mathbf{x})$ is the covariance matrix defined at all pairs of train and test points (an $N_* \times N$ matrix). As these two equations suggest, inference of the latent function defined on the test set, $\mathbf{f}_*$, only depends on the kernel function and the data. An example of GP regression is found in Fig. 2.2, where the observations were generated by a sine function with additive Gaussian noise. The regression was performed by specifying an RBF kernel.

2.5.2 Sparse Gaussian process regression

One of the major drawbacks of GP regression is the $\mathcal{O}(N^3)$ computational cost associated with inverting the covariance matrix. An alternative method to reduce the computational cost is sparse GP regression (Snelson and Ghahramani, 2005); here the idea is to define so called inducing locations $\mathbf{z}$, with corresponding latent function values $\mathbf{f}_z$ which can summarise the training set. By doing so, the computational cost reduces to $O(N|\mathbf{z}|^2)$, where $|\mathbf{z}|$ is the number of inducing points. The key assumption is that the latent function $\mathbf{f}_*$ at any test inputs and the latent function at the training locations $\mathbf{f}$ are conditionally independent given $\mathbf{f}_z$ (Titsias, 2009), that is,

$$p(\mathbf{f}_*|\mathbf{f}, \mathbf{f}_z) = p(\mathbf{f}_*|\mathbf{f}_z).$$

Given this assumption the posterior distribution of the latent function at any test locations given $\mathbf{y}$ is

$$p(\mathbf{f}_*, \mathbf{f}_z|\mathbf{y}) = p(\mathbf{f}_*|\mathbf{f}_z)p(\mathbf{f}_z|\mathbf{y}).$$

We will give more details on this in Chapter 3, where the sparse GP regression framework will be used extensively.

2.6 Hidden Markov models

Hidden Markov models have become very popular in the animal movement community. They find their major applications in the context of switching problems, that is, a canonical challenge whereby the aim is to partition an animal's trajectory based on the associated behavioural state. By behaviour, in this context we refer to those behaviours that can be linked to different movement dynamics, for example an exploratory behaviour or a resting behaviour. As stated in Patterson et al. (2017a), care must be placed when considering many states as in some cases a state could have no biological meaning but be a statistical nuance.

Hidden Markov models can be specified either in discrete- or in continuous-time. In the first case, state transitions can only occur at the time of the observations. This implies that the time step needs to be specified a priori, so that it matches the time scale at which changes in the animals' behaviour occur (Patterson et al., 2017a). In the latter case, the transitions

can occur at any point in time and are not tied to the observation times, thus granting more flexibility to the model.

2.6.1 Discrete-time hidden Markov models

A hidden Markov model (HMM) is a class of generative models composed of two stochastic processes, a latent stochastic process that is unobserved (hidden) and an observable process (the observations). A discrete-time hidden Markov model is specified by the following quantities:

1. a set of n states $\{1,\dots,n\}$;
2. a transition probability matrix $\mathbf{P}$, with $p_{ij} \geq 0$ and $\sum_j p_{ij} = 1$, where each p_{ij} is the probability of transitioning from state i to state j;
3. a sequence of observations $\mathbf{y} = \{y_1,\dots,y_T\}$;
4. a sequence of observed likelihoods, also called emission probabilities and denoted as $e_i(y_t)$, that represent the probability of an observation y_t being generated by a state i. These values are stored in the emission probability matrix $\mathbf{E}$.

Given that each observation is associated with a latent hidden state, we introduce the state sequence as $\mathbf{z} = \{z_1,\dots,z_T\}$, where at each time point the sequence can take one value from the set of states, $z_t = i$, for $i \in \{1,\dots,n\}$. From the definition, we can see how HMMs are a special case of SSMs with discrete state space.

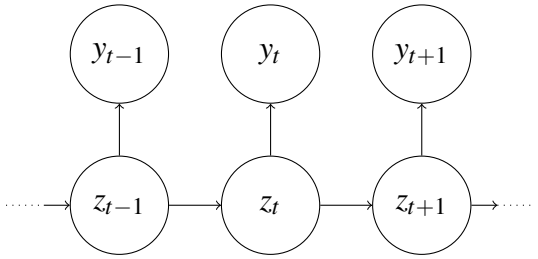


Figure 2.3: Illustration of the structure of a discrete-time hidden Markov model.

There are two underlying assumptions with HMM. The first assumption is that the state sequence forms a first-order Markov chain (Zucchini et al., 2009):

$$p(z_i|z_1,\dots,z_{i-1}) = p(z_i|z_{i-1}). \tag{2.59}$$

The second assumption is called output independence, whereby the probability of an observation is dependent only on the state that has produced the observation:

$$p(y_i|\mathbf{z},\mathbf{y}) = p(y_i|z_i). \tag{2.60}$$

In the context of animal movement modelling, the observations are often taken to be the bivariate time series of step lengths and turning angles (Morales et al., 2004), $\mathbf{y}_t = (\ell_t, \phi_t)$. In what follows, we shall continue using the notation introduced above.

There are three fundamental inference tasks associated with hidden Markov models. The first two tasks are central part of what will be covered in the subsequent chapters and we will describe them in details. Specifically, given an HMM $\mathscr{H} = (\mathbf{P}, \mathbf{E})$ and observations $\mathbf{y}$, what is the *likelihood* of the data being generated by the HMM, $p(\mathbf{y}|\mathscr{H})$? Furthermore, what is the hidden sequence $\mathbf{z}$ that can *best* explain the observations? This task is called *global decoding*. Decoding routine limited to infer the most likely state sequence at each point in time is called *local decoding*.

The third task concerns simulating from the joint posterior distribution of the states using the *forward filtering-backward sampling* algorithm, which is not tackled in this thesis and more details can be found in Zucchini et al. (2009).

The forward algorithm

The forward algorithm is used to calculate the likelihood of the data being generated by a hidden Markov model. Suppose we want to calculate the probability of the observations - this would require marginalising out the hidden state sequences from the joint probability distribution of the data and the state sequences (Zucchini et al., 2009), that is, summation over all possible state sequences:

$$p(\mathbf{y}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y}|\mathbf{z}) p(\mathbf{z}) = \sum_{\mathbf{z}} \prod_{t=1}^{T} p(y_t|z_t) \prod_{t=2}^{T} p(z_t|z_{t-1}), \tag{2.61}$$

where we have used the output independence assumption. The cost of such calculations are n^T, for n the number of states and T the length of observations. The computational cost can be reduced by employing the forward algorithm, an example of dynamic programming algorithm that stores intermediate values as it calculates the probability of the observation sequence (Zucchini et al., 2009).

The idea is the following. Let $\alpha_t(j)$ be the probability that after seeing t observations, the state sequences takes value j, given the HMM model:

$$\alpha_t(j) = p(y_1, \ldots, y_t, z_t = j|\mathscr{H}). \tag{2.62}$$

The α is called a *forward trellis*. This probability can be calculated recursively from α at previous time steps, which are stored in a table thus avoiding expensive calculations:

$$\alpha_t(j) = \sum_{i=1}^{n} \alpha_{t-1}(i) p_{ij} e_j(y_t). \tag{2.63}$$

Then the probability of the observed data given the HMM model is found by employing the forward trellis associated with the last time step and summing over all state configurations:

$$p(\mathbf{y}|\mathscr{H}) = \sum_{i=1}^{n} \alpha_T(i). \tag{2.64}$$

Through the forward algorithm, the computational cost is reduced to $O(n^2T)$.

Viterbi algorithm

The Viterbi algorithm is another example of dynamic programming algorithm based on the forward algorithm and is employed in the decoding task, that is, finding the optimal state sequence to explain the observed data. It is very similar to the forward algorithm but here summation over the previous forward trellis is replaced by maximisation. Let $v_t(j)$ be the *Viterbi trellis*, namely the probability that the hidden sequence is in state j after passing through the most probable hidden state sequence:

$$v_t(j) = \max_{z_1,\ldots,z_{t-1}} p(y_1,\ldots,y_t,z_1,\ldots,z_{t-1},z_t = j|\mathscr{H}). \tag{2.65}$$

This probability is defined recursively using the Viterbi trellis at the previous time step:

$$v_t(j) = \max_{i=1}^{n} v_{t-1}(i)p_{ij}e_j(y_t). \tag{2.66}$$

Given that the algorithm finds the optimal state sequence, at every step of the algorithm the best state is saved in the so-called Viterbi backpointers:

$$bp_t(j) = \operatorname*{argmax}_{i=1}^{n} v_{t-1}(i)p_{ij}e_j(y_t). \tag{2.67}$$

Finally, the most probable hidden state sequence along with the associated probability are found by the Viterbi trellis calculated at the last time point:

$$\begin{aligned} &\text{best path: } \operatorname*{argmax}_{i=1}^{n} v_T(i)p_{ij}e_j(y_T), \\ &\text{probability: } \max_{i=1}^{n} v_T(i)p_{ij}e_j(y_T). \end{aligned} \tag{2.68}$$

2.6.2 Continuous-time hidden Markov models

A continuous-time HMM has the same dependence structure as a discrete-time HMM but the state process is defined by a continuous-time Markov chain, that is, it is defined by an IGM as in Eqn. 2.3. The structure of a continuous-time HMM is illustrated in Fig. 2.4.

We can see that the distribution of an observation only depends on the current value of the state, as in discrete-time HMM, however in a continuous-time HMM, the times of transitions $(\tau-1,\tau,\tau+1)$ and the times of observation $(t-1,t,t+1)$ do not need to match, and both may be irregularly spaced (Glennie et al., 2023).

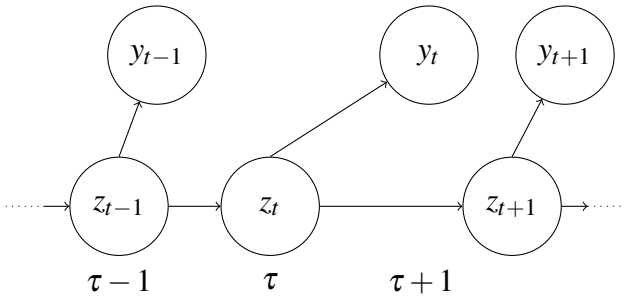


Figure 2.4: Illustration of the structure of a continuous-time hidden Markov model.

The forward algorithm and the Viterbi algorithm can also be used in continuous time (Glennie et al., 2023). One major difference between the two model formulations is related to output independence assumption, also known as the *snapshot property*. In discrete time, this is a well-understood assumption (Glennie et al., 2023); however, in continuous time, state transitions are not constrained to occur at the time of observations. Continuous-time HMMs are therefore only suitable when the distribution of each observation does not depend on the switches that have occurred between any two consecutive observations. However, when this is not the case, the snapshot property may still be a reasonable approximation in cases where observations occur at a high temporal resolution relative to the scale of state switching, that is, if the number of switches occurring within observation intervals is small (Glennie et al., 2023).

We have given an overview on the theoretical background needed to understand the material covered in the subsequent chapters. The Gaussian processes machinery as well as the variational inference approach will be crucial in Chapter 3, whereas hidden Markov models and the EM algorithm will be greatly relevant for Chapters 4-6.

Chapter 3

Inferring microscale properties of interacting systems from macroscale observations

The material of this chapter was developed during the course of two accepted publications. The first paper was accepted in the proceedings of ICSTA 2020 (Campioni et al., 2020) and was essentially the preliminary work that has lead to the second publication in the Physical Review Research journal (Campioni et al., 2021). Here we report the second publication.

3.1 Abstract

Emergent dynamics of complex systems are observed throughout nature and society. The coordinated motion of bird flocks, the spread of opinions, fashions and fads, or the dynamics of an epidemic, are all examples of complex macroscale phenomena that arise from fine-scale interactions at the individual level. In many scenarios, observations of the system can only be made at the macroscale, while we are interested in creating and fitting models of the microscale dynamics. This creates a challenge for inference as a formal mathematical link between the micro and macro scale is rarely available. Here, we develop an inferential framework that bypasses the need for a formal link between scales and instead uses sparse Gaussian process regression to learn the drift and diffusion terms of an empirical Fokker-Planck equation which describes the time evolution of the probability density of a macroscale variable. This gives us access to the likelihood of the microscale properties of the physical system and a second Gaussian process is then used to emulate the log-likelihood surface, allowing us to define a fast, adaptive MCMC sampler which iteratively refines the emulator when needed. We illustrate the performance of our method by applying it to a simple model of collective motion.

3.2 Introduction

Complex systems are characterised by multiscale dynamics, with a high-dimensional microstate that describes the state of the individual components, and a reduced dimension macrostate that emerges from interactions at the lower level (Anderson, 1972, Sethna, 2006). Connecting these two scales is the canonical challenge of complex systems science (Shalizi, 2006, Prokopenko et al., 2009). In certain cases a formal mathematical derivation of equations describing the macrostate may be obtained based on the properties of the lower level components (Demirel et al., 2014, Bellomo et al., 2015, Toner and Tu, 1998), however this often requires simplifying assumptions that cannot be justified in most scenarios. In the absence of a formal mathematical link between scales, the inference of microscale dynamics from macroscale observations is challenging. While forward simulations of complex computer models are able to link microscale parameters with coarse grained observables, the inverse problem of statistical inference remains largely intractable (Nguyen et al., 2017). This is due to the unavailability of the probability density, or likelihood, for an observation. Several simulation-based approaches have been proposed that approximate the intractable likelihood but these are often computationally expensive and lack a formal quantification of uncertainty (Cranmer et al., 2020b, Wood, 2010, Wilkinson, 2014).

Here, we approach the problem of multiscale inference by assuming the existence of an empirical Fokker-Planck equation (FPE) that describes the changes with time of the probabil-

ity density of a macroscale variable. We consider a scenario where we are able to efficiently run forward simulations of the model but do not have access to a likelihood function that provides the probability density of the empirical observations for a given parameter set i.e. we cannot derive the drift and diffusion functions of the FPE from the microscale parameters. Instead through the application of sparse Gaussian process (GP) regression (Lázaro-Gredilla and Titsias, 2011, Saul et al., 2016) we learn the drift and diffusion functions of the FPE from simulations. This allows us to estimate the likelihood of microscale parameters given a set of empirical macroscale observations.

Our approach presents several advances over existing methods for inference in complex systems. Firstly, we incorporate several concepts from equation-free modelling (Kevrekidis et al., 2003) into the inference process. Equation-free modelling offers an efficient numerical method for investigating the macroscale dynamics of microscale models. By integrating aspects of this approach with sparse Gaussian process regression of the drift and diffusion functions we are able to formally quantify the uncertainty inherent in simulations of stochastic microscale models. This allows us to connect the equation-free framework with an MCMC sampler that directs simulation effort to refining regions of parameter space with high likelihood.

A conceptual overview of our framework is provided in Fig. 3.1. This figure illustrates the three main components of our framework. Firstly, we employ a microscale simulator to generate the macroscale dynamics of our physical model for a given microscale parameter set. Secondly, we use sparse GP regression to link the two scales and learn the macroscale drift and diffusion functions from simulation output. This allows us to estimate the likelihood of the microscale parameters given the observed data, and further provides a formal quantification of the uncertainty in the estimate. Finally, the likelihood estimate and its associated uncertainty are passed to an adaptive MCMC algorithm that samples from the posterior distribution of the microscale parameters. The adaptive MCMC sampler employs a second, independent implementation of sparse GP regression to emulate the log-likelihood surface of the microscale parameters and uses the emulated surface when uncertainty is low but triggers further forward simulations when uncertainty is high.

The remainder of this paper is structured as follows. In Section 3.3, our simulation model is introduced. Section 3.4 describes the theoretical foundation for our framework which is explained in detail in Section 3.5. In Section 3.6 we give details about the parameter settings we adopt to produce the results presented in Section 3.7. Lastly, we discuss the effectiveness and potential applications of our method in Section 3.8.

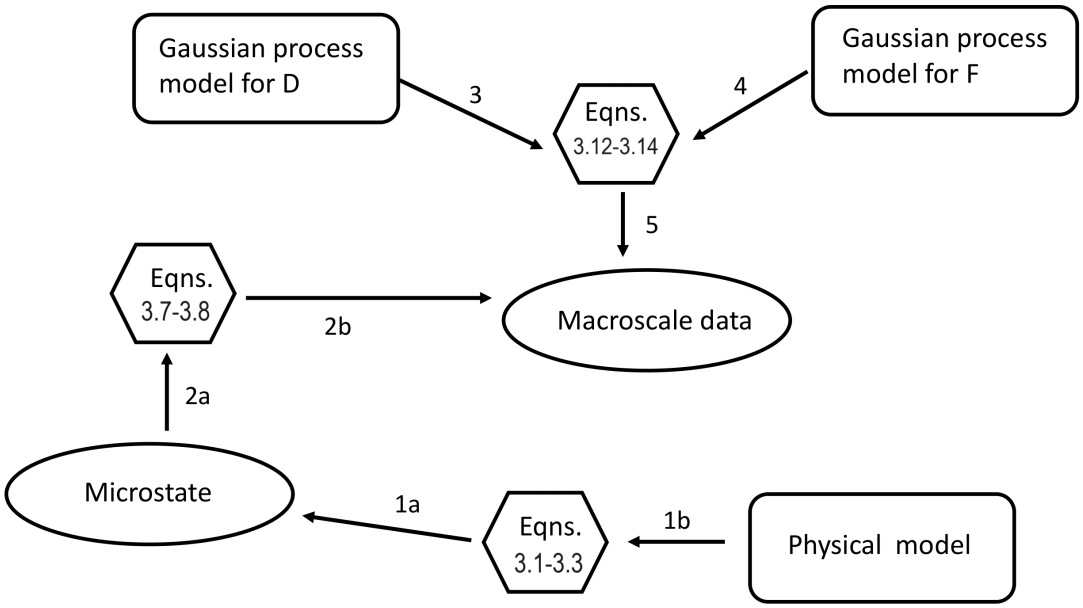


Figure 3.1: Graphical summary of the proposed inference scheme. If the microstates, corresponding to the particles' positions $x_i(t)$ and velocities $u_i(t)$ in Eqn. (3.1), were observable, we could infer the parameters of the physical model directly from the corresponding microstate data (arrows 1a and 1b). However, such high-resolution data is usually not available, and the challenge therefore is to infer the physical model parameters from macroscale features - the distribution of average velocities U in our case (arrows 2a and 2b). This distribution is in principle defined by the physical model and its parameters via Eqn. (3.7), giving rise to the likelihood of the physical model given the macroscale data (Eqn. (3.8)). However, the mathematical expression of this physical model likelihood depends on two functions - the diffusion function D and the drift function F - which are not analytically tractable. We therefore approximate these functions by two Gaussian process models fitted to simulated macroscale output, based on Eqns. (3.12–3.14) (arrows 3-5). Note that the inference of these Gaussian processes is based on the probability of the simulation output given the Gaussian process, which is independent of the physical model. Inserting the Gaussian process approximations of F and D back into Eqn. (3.7) then leads to an approximation of the physical model likelihood (Eqn. (3.8)), which is used for inference of the physical model parameters. Note that in order to make the inference computationally efficient, the physical model likelihood is approximated (or emulated) by another Gaussian processes, which is not included in the present figure.

3.3 The model

We demonstrate our framework on a simple model of collective animal movement adapted from Buhl et al. (2006), Czirók et al. (1999). The model is a one-dimensional, self-propelled particles (SPPs) model with the following equations describing the evolution of the positions $x_i(t)$ and velocities $u_i(t)$ of an individual that moves along a line of dimensionless length with periodic boundary conditions,

$$\begin{aligned} x_i(t+\Delta t) &= x_i(t) \ + \Delta t \ v_0 \ u_i(t), \\ u_i(t+\Delta t) &= u_i(t) + \alpha(G(\bar{u}_i(t,\delta)) - u_i(t)) + \xi_i, \end{aligned} \tag{3.1}$$

where v_0 is a constant scaling of each particle's velocity, the quantity $\bar{u}_i(t,\delta)$ is the average velocity of all individuals, excluding individual i, within a metric interaction range of length δ, the parameter α represents the relative weight that an individual assigns to its own velocity and those of its neighbours when updating its velocity, ξ_i is a random noise term taken from a normal distribution $\mathcal{N}(0,\eta^2\Delta t)$, and the function G represents a social interaction term which causes an individual to adopt a similar velocity to its observed neighbours,

$$G(z) = \begin{cases} (z+1)/2, & z > 0 \\ (z-1)/2, & z < 0. \end{cases} \tag{3.2}$$

The model was developed for the study of locust moving in an annular arena (Buhl et al., 2006, Yates et al., 2009) and is characterised by a double-well potential with intermittent switches occurring between metastable states representing clockwise and counterclockwise motion (see Fig. 3.2a for an example time series). The dynamics of the model are governed by three parameters; the interaction radius, the strength of the social force, and the noise level which we define as our microscale parameter vector,

$$\boldsymbol{\theta} = (\alpha, \delta, \eta). \tag{3.3}$$

Ideally, we would like to infer these parameters from detailed measurements or observations of the microstates $\{x_i(t), u_i(t)\}$. However, such high-resolution data are rarely available. In the present paper we therefore pursue an approach that focuses on the emergent macroscale properties of the system using equation-free modelling, to be explained in Section 3.5.

As in Yates et al. (2009) our coarse macroscale variable is taken to be the global average velocity

$$U(t) = \frac{1}{N}\sum_{i=1}^{N} u_i(t). \tag{3.4}$$

Yates et al. (2009) show that in the case of infinite interaction radius δ the evolution of U can be described by a stochastic differential equation (SDE) of the following form

$$dU = F(U,\boldsymbol{\theta})dt + \sqrt{D(U,\boldsymbol{\theta})}dW_t, \tag{3.5}$$

where dW_t is a Wiener process, $F(U,\boldsymbol{\theta})$ is the drift function and $D(U,\boldsymbol{\theta})$ is the diffusion function, both of which are available in closed form. Following Yates et al. (2009) we assume that for finite interaction radius δ, an FPE of the same form can be assumed to exist. However, in that case, $F(U,\boldsymbol{\theta})$ and $D(U,\boldsymbol{\theta})$ are no longer available analytically and have to be empirically inferred from the data.

Eqn. (3.5) gives rise to an associated FPE (Gardiner, 2009) that describes the evolution of $\rho(U,t)$ the probability density function of $U(t)$,

$$\frac{\partial \rho}{\partial t} = \frac{1}{2}\frac{\partial^2(D(U,\boldsymbol{\theta})\rho)}{\partial U^2} - \frac{\partial(F(U,\boldsymbol{\theta})\rho)}{\partial U}. \tag{3.6}$$

For known drift and diffusion functions, the stationary probability density (SPD) $\rho_s(U|\boldsymbol{\theta})$ can be calculated as (Risken, 2012)

$$\rho_s(U|\boldsymbol{\theta}) = \frac{1}{Z}\exp\left(2\int_0^U \frac{F(s,\boldsymbol{\theta})}{D(s,\boldsymbol{\theta})}ds - \ln\left(D(U,\boldsymbol{\theta})\right)\right) \tag{3.7}$$

where Z is a normalising constant.

Hence if the drift and diffusion functions can be derived from the microscale parameters, Eqn. (3.7) gives us access to

$$\mathscr{L}(\boldsymbol{\theta};U_{data}) \;:=\; \rho_s(U_{data}|\boldsymbol{\theta}) \tag{3.8}$$

the likelihood of model parameters $\boldsymbol{\theta}$ given empirical observations of the group average velocity $U_{data} = \{U_1, U_2, \ldots, U_n\}$. With this likelihood, one can pursue parameter inference in a classical sense via maximum likelihood estimation or within a Bayesian framework by sampling from the $\boldsymbol{\theta}$ posterior distribution, after defining an appropriate prior distribution.

However in most scenarios, the drift and diffusion functions are intractable and cannot be derived from microscale parameters. A variety of likelihood-free methods have been developed recently, see e.g. King et al. (2016), Cranmer et al. (2020a), Owen et al. (2015), but they are intrinsically approximate and depend on various heuristics and intuition. To overcome this difficulty, we propose to employ sparse Gaussian process regression to learn these functions from fine-scale simulations of the model.

3.4 Background

In this section we summarise the inferential machinery which forms the foundation of our framework. Specifically, we review the concepts of Gaussian process regression (Rasmussen and Williams, 2006) and sparse Gaussian process regression (Snelson and Ghahramani, 2005, Opper and Archambeau, 2009, Titsias, 2009). We use the letter x to denote the input or explanatory variable of the function of interest, and the letter y to denote the response or output variable. Depending on the application, x may represent the macroscale velocity U, with y representing the corresponding outputs of $F(U,\boldsymbol{\theta})$ and $D(U,\boldsymbol{\theta})$, or x may represent the microscale parameter vector $\boldsymbol{\theta}$, with y representing the corresponding log-likelihood. Please note that x is not to be confused with the spatial coordinate in Eqn. (3.1).

Gaussian processes are stochastic processes for which any finite collection of random variables follows a multivariate normal (MVN) distribution. Gaussian processes are therefore determined by a mean function $\boldsymbol{m}$ and a covariance matrix $\boldsymbol{K}$, often defined as a covariance kernel $K_{ij} = k(x_i, x_j)$ that is a function of the location of the random variables. In Gaussian process regression we seek to learn a latent function f based on a set of N observations $\mathbf{y} = \{y_i\}_{i=1}^N$ at locations $\mathbf{x} = \{x_i\}_{i=1}^N$, where

$$y_i = f(x_i) + \nu, \tag{3.9}$$

and ν is an additive Gaussian white noise term. The latent function f is a realisation of a Gaussian process and is modelled with a GP prior,

$$p(\mathbf{f}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}).$$

GP regression may also be extended to consider the case of heteroscedastic noise where the variance of the observation noise ν is itself a function of x (Goldberg et al., 1998). GPs inherit all properties from MVNs; hence, performing GP regression when the likelihood is also Gaussian involves calculating the conditional distribution of a joint MVN. Given a set of training observations $\mathbf{y}$ at locations $\mathbf{x}$, and assuming a zero-mean process, it follows that the posterior distribution of $\mathbf{f}^*$ at a set of N^* test locations $\mathbf{x}^*$ is given by (Murphy, 2012)

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

where

$$\begin{aligned}
\boldsymbol{\mu}^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I})^{-1}\mathbf{y} \\
\boldsymbol{\Sigma}^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) \\
&\quad - \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I})^{-1}\mathbf{K}(\mathbf{x}^*, \mathbf{x})^T
\end{aligned}$$

Here $\mathbf{I}$ is the identity matrix and $\mathbf{K}(\mathbf{x}^*, \mathbf{x})$ is the covariance matrix defined at all pairs of train and test points (i.e. an $N^* \times N$ matrix) with similar definitions for the other covariance terms. As these two equations suggest, inference of the latent function defined on the test set, $\mathbf{f}^*$, only depends on the kernel function and the data. The kernel function defines the correlation structure of the GP. In our work we employ one of the most commonly used kernels, the Exponentiated Quadratic kernel, also known as the Radial Basis Function (RBF) kernel,

$$k(x_i, x_j) = \tau^2 exp\left(- \frac{||x_i - x_j||^2}{\vartheta} \right).$$

The τ^2 and ϑ kernel hyperparameters represent the scale, or amplitude, of the process and its lengthscale and $||\cdot, \cdot||$ is the Euclidean norm. For a review of alternative kernels, see Rasmussen and Williams (2006).

One of the major drawbacks of GP regression is the $\mathcal{O}(N^3)$ computational cost associated with inverting the covariance matrix. The idea behind sparse GP regression (Snelson and Ghahramani, 2005) is to define so called inducing locations $\mathbf{z}$, with corresponding latent function values $\mathbf{f}_z$ which can summarise the training set. The key assumption is that the latent function $\mathbf{f}^*$ at any test inputs and the latent function at the training locations $\mathbf{f}$ are conditionally independent given $\mathbf{f}_z$ (Titsias, 2009), i.e.

$$p(\mathbf{f}^* | \mathbf{f}, \mathbf{f}_z) = p(\mathbf{f}^* | \mathbf{f}_z).$$

Given this assumption the posterior distribution of the latent function at any test locations given $\mathbf{y}$ is

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

Variational inference proceeds by introducing a variational approximation to this posterior,

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

where $\phi(\mathbf{f}_z)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_q$ and covariance $\boldsymbol{\Sigma}_q$. To determine the optimal variational parameters, $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$, we can maximise a lower bound on the marginal log-likelihood, which is equivalent to minimising the Kullback-Leibler (KL) divergence between the true posterior and the variational distribution. Following Saul et al. (2016), the

evidence lower bound can be obtained via Jensen's inequality,

$$\begin{aligned}\log p(\mathbf{y}) &= \log \int\int p(\mathbf{y}|\mathbf{f},\mathbf{f}_z)p(\mathbf{f},\mathbf{f}_z)d\mathbf{f}d\mathbf{f}_z \\ &= \log \int\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f},\mathbf{f}_z)\frac{q(\mathbf{f},\mathbf{f}_z)}{q(\mathbf{f},\mathbf{f}_z)}d\mathbf{f}d\mathbf{f}_z \\ &\geq \int\int \log\left(\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f},\mathbf{f}_z)}{q(\mathbf{f},\mathbf{f}_z)}\right)q(\mathbf{f},\mathbf{f}_z)d\mathbf{f}d\mathbf{f}_z \\ &\geq \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} - \mathbf{KL}\left(\phi(\mathbf{f}_z)\|p(\mathbf{f}_z)\right) \end{aligned} \tag{3.10}$$

where $q(\mathbf{f}) = \int q(\mathbf{f},\mathbf{f}_z)d\mathbf{f}_z$ and the **KL** term denotes the KL divergence between the prior distribution over $\mathbf{f}_z$ and the variational posterior.

Since $\phi(\mathbf{f}_z)$ is a multivariate Gaussian, $q(\mathbf{f})$ and the KL term in Eqn. (3.10) are available in closed form. As the likelihood factorises across the data, i.e.

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i|f_i),$$

the integral $\int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f}$ can be decomposed into N one dimensional integrals that are tractable when the likelihood is Gaussian (Hensman et al., 2013).

In Saul et al. (2016) the sparse variational method is extended to incorporate likelihoods that depend on multiple latent functions, this is termed a chained, or multi-latent GP. Considering the case when the likelihood depends on two latent functions, $\mathbf{f}$ and $\mathbf{g}$, the lower bound on the marginal log-likelihood is now (Saul et al., 2016)

$$\begin{aligned}\log p(\mathbf{y}) \geq &\int\int \log p(\mathbf{y}|\mathbf{f},\mathbf{g})q(\mathbf{f})q(\mathbf{g})d\mathbf{f}d\mathbf{g} \\ &- \mathbf{KL}\left(\phi(\mathbf{f}_z)\|p(\mathbf{f}_z)\right) - \mathbf{KL}\left(\phi(\mathbf{g}_z)\|p(\mathbf{g}_z)\right)\end{aligned} \tag{3.11}$$

where $q(\mathbf{g}) = \int p(\mathbf{g}|\mathbf{g}_z)\phi(\mathbf{g}_z)d\mathbf{g}_z$ and we have introduced two Gaussian distributions that are variational approximations to the posterior at the inducing point locations, $\phi(\mathbf{f}_z) \sim \mathcal{N}(\boldsymbol{\mu}_q^f, \boldsymbol{\Sigma}_q^f)$ and $\phi(\mathbf{g}_z) \sim \mathcal{N}(\boldsymbol{\mu}_q^g, \boldsymbol{\Sigma}_q^g)$.

In this work we consider the case where the second latent GP determines an input dependent heteroscedastic noise term, such that

$$\begin{aligned} y_i &\sim \mathcal{N}(f(x_i), e^{g(x_i)}) \\ p(\mathbf{f}|\mathbf{x}) &\sim \mathcal{N}(\boldsymbol{m}_f, \boldsymbol{K}_f) \\ p(\mathbf{g}|\mathbf{x}) &\sim \mathcal{N}(\boldsymbol{m}_g, \boldsymbol{K}_g). \end{aligned}$$

Hence, the latent function f determines the mean of y_i at location x_i, while the latent function g is exponentiated so that it is constrained positive and is then the location dependent variance

in y_i. In this case the integral in Eqn. (3.11) is analytic (Lázaro-Gredilla and Titsias, 2011) and a closed-form lower bound can be obtained. This lower bound can then be maximised using stochastic optimisation (Hensman et al., 2013) in order to find the optimal variational distributions $\phi(\mathbf{f}_z)$ and $\phi(\mathbf{g}_z)$.

3.5 Methods

3.5.1 Microscale simulations

Simulations of the microscale model are implemented in Python using the machine learning library TensorFlow (Abadi et al., 2016) and run in parallel on a GPU. Using this approach, we are able to run multiple independent instances of the model for each set of values of the parameter vector $\boldsymbol{\theta}$.

In equation-free modelling (Kevrekidis et al., 2003) short bursts of simulations are run and used to learn about the macroscale dynamics. The approach involves moving from the microscale to the macroscale, termed restriction, and moving from the macroscale to the microscale, termed lifting. The first process, restriction, is straightforward and for our model involves applying Eqn. (3.4) to calculate the macrostate variable from the microstate. The lifting process is more involved and requires mapping a macrostate to a specific microstate. Initialising a microstate at random with a defined macrostate will introduce a lifting error as not all configurations with the same macrostate variable are equally likely.

In our application we seek to obtain values of $U_{t+\Delta t} - U_t$ from simulations that are distributed evenly across the domain of U. If we allow the microstate to evolve and record U_t throughout the simulations we will inevitably end up with simulation outputs focused in regions of high probability density, and few measurements from areas of low probability density, which in our model corresponds to $U_t \simeq 0$.

To overcome this issue, we simulate the dynamics for an initial period of time and then successively perturb the microstates of each parallel simulation to a desired set of macrostate locations. We then run several time steps of the simulations from the perturbed microstates and record the output to use in estimating the drift and diffusion functions. For example, if we are running 1000 simulations in parallel, we run the simulations for an initial number of timesteps. Next, we define a desired set of 1000 macrostate variables U_t that are uniformly distributed across the domain. We then map each desired macrostate to the microscale simulation with the closest macrostate variable. Each simulation is subsequently perturbed by altering the velocity of each individual, so that the macrostate of the simulation matches the desired macrostate, and run forward for a number of time steps. By repeating these steps, we are able to accelerate the coverage of the whole configuration space without having to wait for the system to evolve to particular locations.

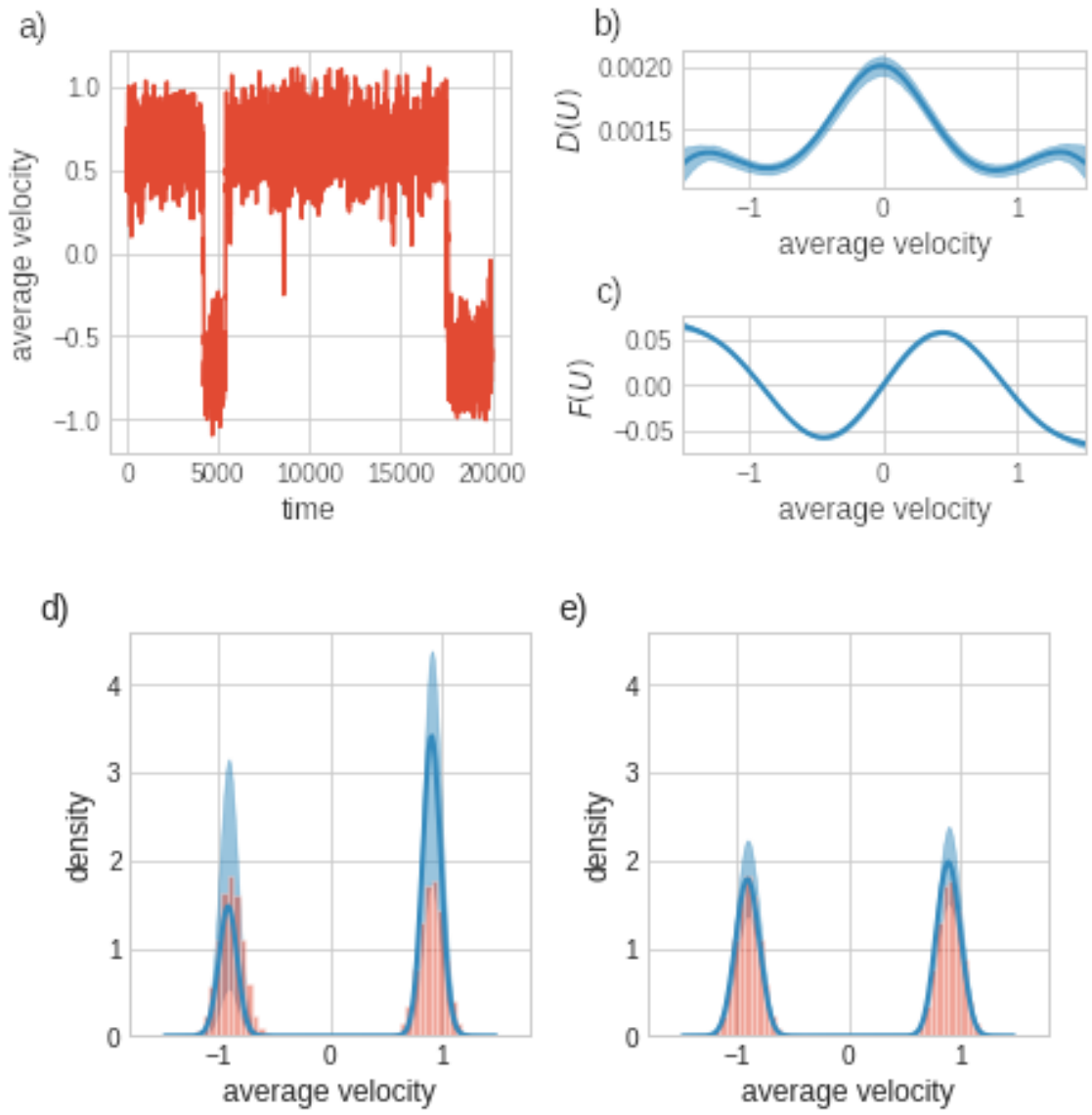


Figure 3.2: a) Sample time series from the simulation model for 20 individuals moving along a line of dimensionless length 36 with parameters $v_0 = 1$ and $\Delta t = 1$. The switches between the two metastable states $U = 1$, $U = -1$ represent cohesive movement in clockwise or counterclockwise direction. b) Inferred diffusion function $D(U)$ (dark blue line) and 95% posterior credible interval (light blue shaded region) learnt from simulation results using sparse GP regression, as a function of the average velocity U. c) Idem for the drift function $F(U)$. d) Stationary probability density mean and uncertainty calculated based on 50,000 simulation outputs. e) Stationary probability density from 500,000 simulation outputs. Note, by increasing the number of simulations the uncertainty has greatly reduced.

3.5.2 Inferring the drift and diffusion functions

As closed-form expressions for the drift and diffusion functions are unavailable in our application, we employ sparse Gaussian process regression to learn these functions from fine-scale simulations of the model. For notational simplicity, we will omit the explicit dependence on $\boldsymbol{\theta}$ and from here on use $F(U)$, $D(U)$ and $\rho_s(U)$ to indicate respectively the drift and diffusion functions and the SPD.

Following Batz et al. (2018), we define an analogous, discretised version of Eqn. (3.5) as a stochastic difference equation

$$U_{t+\Delta t} - U_t = F(U_t)\Delta t + \varepsilon\sqrt{D(U_t)\Delta t}, \tag{3.12}$$

where Δt is a discrete time step and $\varepsilon \sim \mathcal{N}(0,1)$. By comparing Eqns. (3.9) and (3.12) we can observe that learning the drift and diffusion functions is an example of heteroscedastic Gaussian process regression where

$$y = \frac{U_{t+\Delta t} - U_t}{\Delta t}, \tag{3.13}$$

$F(U_t)$ is the latent function, and the variance of the heteroscedastic noise term is

$$\sigma^2 = \frac{D(U_t)}{\Delta t}. \tag{3.14}$$

It is possible to learn the drift and diffusion functions sequentially from measurements of $U_{t+\Delta t} - U_t$ as in Campioni et al. (2020), Batz et al. (2018) by employing the following equation to infer $D(U)$,

$$D(U) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} E[(U_{t+\Delta t} - U_t)^2 | U_t = U]. \tag{3.15}$$

However, this approach leads to a systematic bias for finite Δt (Ragwitz and Kantz, 2001). To overcome this issue, we employ sparse GP regression within a variational framework (Saul et al., 2016). This allows the drift and the diffusion functions to be learnt simultaneously from simulations of $U_{t+\Delta t} - U_t$ and enables us to deal with large numbers of simulation outputs ($N \sim 10^5$). Note that inferring the drift and diffusion functions involves maximising a lower bound on a second, different likelihood that depends on simulation output and not empirical observations. This second likelihood is a standard GP likelihood as defined in Eqn. (3.10) and is separate from the physical model likelihood defined in Eqn. (3.8). Hence, the variational approximation to the posterior for the drift and diffusion functions, along with the kernel parameters and inducing point locations, are optimised by maximising a variational lower bound on the marginal GP log-likelihood extended to multiple latent functions (Saul et al., 2016) given in Eqn. (3.11).

Our implementation uses the GPflow library (Matthews et al., 2017), a package for building Gaussian process models using TensorFlow (Abadi et al., 2016, Dillon et al., 2017). We specify Gaussian process priors on the latent drift and diffusion functions $F(U)$ and $D(U)$ with separate independent RBF kernels.

Once optimised, we obtain the diffusion function posterior distribution, shown in Fig. 3.2b, and the drift function posterior, shown in Fig. 3.2c. These distributions capture the inherent uncertainty in the functions due to the finite number of microscale simulations that can be performed. By drawing multiple samples from the posterior drift and diffusion functions we are able to propagate this uncertainty into the stationary probability density function of the macroscale observations U_{data} by calculating Eqn. (3.7) for each sample using numerical quadrature. Thus, given a sequence of empirical macroscale observations, we are able to calculate an estimate of the data likelihood $\mathscr{L}(\boldsymbol{\theta};U_{data})$ as well as formally quantifying the uncertainty in the estimate.

Figs. 3.2d and 3.2e show $\rho_s(U)$, the steady state probability density of the global average velocity U calculated for different numbers of simulation outputs. As we would expect, increasing the number of simulation outputs decreases the uncertainty in $\rho_s(U)$ and will subsequently lead to reduced uncertainty in the likelihood $\mathscr{L}(\boldsymbol{\theta};U_{data})$.

The uncertainty in the likelihood is propagated from the uncertainty in the drift and diffusion functions; as our model is stochastic, this variability is intrinsic. Indeed, in the context of our framework, where simulations are stochastic and closed-form expressions for $F(U)$ and $D(U)$ are unavailable, this variability will be present for any finite number of simulation outputs.

3.5.3 The sampling algorithm

The stationary probability density defined by Eqn. (3.7) is of central importance in our framework as it provides the tools to access the likelihood of microscale parameters $\boldsymbol{\theta}$ given observations of our macroscale variable U. We perform inference of $\boldsymbol{\theta}$ in a Bayesian framework and aim to sample from the posterior distribution of the microscale parameters through an adaptive MCMC sampler based on the method proposed in Conrad et al. (2016).

In a standard Metropolis-Hastings (MH) framework (Hastings, 1970), a candidate sample $\boldsymbol{\theta}^+$ is generated from an initial point in parameter space $\boldsymbol{\theta}^-$ through a specified transition kernel. The sampler evaluates the likelihood L and priors π of the two points and then either accepts or rejects the candidate location via an acceptance function. For a symmetric transition kernel, a commonly used acceptance function is

$$a = min\left\{1, \frac{L(\boldsymbol{\theta}^+)\pi(\boldsymbol{\theta}^+)}{L(\boldsymbol{\theta}^-)\pi(\boldsymbol{\theta}^-)}\right\} \tag{3.16}$$

and the candidate location is accepted with probability a. A naive implementation of the MH sampler would involve a computationally expensive forward simulation at each step and would base acceptance on a point estimate of the data likelihood. Instead, we accelerate the sampling using a sparse GP emulator (Gardner et al., 2018) that approximates the log-likelihood surface. Note, we therefore introduce sparse Gaussian process regression at two points in our framework, firstly to learn a drift and diffusion function from microscale simulations, then secondly to emulate the log-likelihood surface when running the sampler.

For our sampling procedure, we need to choose a design for the locations of a set of points to initialise the emulator. The aim is pick input parameters to cover the whole parameter domain efficiently. Naive designs include specifying a regular grid of parameter values, which suffers from the curse of dimensionality, or drawing samples from a uniform distribution in the parameter domain, which is inefficient due to random clustering of points and large gaps. As our initial set of points needs to span a high-dimensional microscale parameter space, we employ a space-filling design, which is a computationally efficient way to place points in a high-dimensional space such that there is a measure of uniformity in how they fill the space, i.e. they do not leave large gaps. In our work, we use a Sobol sequence (Santner et al., 2003) to create an initial log-likelihood map over a pre-specified region. For each set of initial values, we run forward simulations that yield a distribution over the drift and diffusion functions. We then generate k samples of the log-likelihood given empirical observations U_{data} by sampling the drift and diffusion functions and inserting them into Eqn. (3.7), effectively yielding k log-likelihood evaluations for the same parameter values. The initial points form a training set for the emulator used by the sampler.

Starting from the highest log-likelihood initial location, the sampler then uses the emulated log-likelihood surface as a surrogate for the full forward simulations (Gramacy, 2020). We employ sparse GP regression again using the GPflow package (Matthews et al., 2017). We maximise the lower bound on the marginal GP log-likelihood from Eqn. (3.10) to optimise the variational approximation and the GP hyperparameters. As we have intrinsic stochasticity in the simulation outputs and k samples of the log-likelihood at each simulation location, this is mathematically equivalent to fitting a Gaussian process with observation noise, or non-zero nugget term (Rasmussen and Williams, 2006). By letting the GP learn the nugget parameter and covariance kernel lengthscale, the sampler is able to accurately estimate the log-likelihood surface at a candidate location along with its associated uncertainty.

At each step of the sampler, a candidate location $\boldsymbol{\theta}^{+}$ is generated. The sampler assesses the uncertainty in the log-likelihood surface at the proposal location $\boldsymbol{\theta}^{+}$ and the current location $\boldsymbol{\theta}^{-}$. If the uncertainty is sufficiently low then the candidate is accepted according to Eqn. (3.16), otherwise further forward simulations are triggered to refine the emulator (Conrad et al., 2016).

The refinement criterion is based on the uncertainty associated with the log-likelihood

surface. Given proposed and current locations with log-likelihoods l^+ and l^- and associated uncertainties σ^+, σ^-, we define an uncertainty indicator $P^\pm$ as

$$P^\pm = \begin{cases} P(\zeta^+ \leq \zeta^-) & \text{if} \quad l^+ \leq l^- \\ P(\zeta^+ > \zeta^-) & \text{otherwise,} \end{cases}$$

where ζ^+ and ζ^- are random variables drawn from $\mathcal{N}(l^+, \sigma^{+2})$ and $\mathcal{N}(l^-, \sigma^{-2})$ respectively.

The uncertainty in the log-likelihood is therefore quantified by comparing the expected relationship between two random samples from the emulator surface with the mean values. $P^\pm$ ranges from 0.5, meaning there is no discernible difference between the two locations, to 1, indicating complete certainty in the relationship between l^+ and l^-.

The refinement probability is then defined as

$$\gamma = 2\bar{\gamma}(s)(1 - P^\pm)$$

where s is the number of steps since the last refinement and $\bar{\gamma}(s)$ is a logistic function with a specified slope and midpoint that acts as a memory and prevents refinement at every step. As there is intrinsic uncertainty in the likelihood (unlike in Conrad et al. (2016)) forward simulations could in principle be triggered at every step, hence our refinement probability takes on smaller values whenever a microscale simulation has just been triggered and increases with the number of steps taken since refinement. If refinement is triggered, forward simulations are run at the location with largest uncertainty. Along with triggered refinement, we also include random refinement at each step for ensuring asymptotic convergence and note that our uncertainty indicator is invariant to relabelling of $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ meaning the refinement process does not impact the reversibility of the transition kernel (Conrad et al., 2016). When refinement occurs, the log-likelihood samples are added to the emulator GP training set and further optimisation is performed.

Our final comment concerns the theoretical guarantee of convergence to the true posterior distribution. A related proof is provided in Conrad et al. (2016), but for deterministic systems. Our system is stochastic, and the likelihood of Eqns. (3.7–3.8), estimated by fitting GPs for $F(U)$ and $D(U)$ to finite numbers of forward simulations from Eqn. (3.1), is itself subject to uncertainty. However, convergence to the true posterior distribution is guaranteed by combining the proof in Conrad et al. (2016) with the following three well-established facts (all subject to adequate regulatory conditions): (i) that a neural network with a sufficiently large number of hidden nodes is a universal approximator and, thus, unbiased (Cybenko, 1989, Hornik, 1991); (ii) that a Gaussian process is the limiting case of a neural network with an infinite number of hidden nodes (Neal, 1996); and that (iii) replacing the true likelihood by an unbiased estimate does not affect the limiting distribution of an MCMC sampling scheme

(Andrieu and Roberts, 2009, Andrieu et al., 2010).

3.6 Empirical study

The drift and diffusion functions inference is performed using sparse GP regression and we specify $M = 20$ inducing points, which we found to provide an appropriate trade-off between accuracy and computational costs.

For the inference of model parameters $\boldsymbol{\theta}$, we first create synthetic data by running multiple parallel simulations of the model with the given parameter set. After relaxation of transients, we randomly select 200 simulated macroscale outputs as our empirical data. Denoting the synthetic data set as $U_{data} = \{U_1, U_2, \ldots, U_{200}\}$, a sample of the log-likelihood of the parameter set given the data is given by

$$l^k = \sum_{n=1}^{200} \log \rho_s^k(U_n)$$

where $\rho_s^k(U_n)$ is the stationary probability density from Eqn. (3.7), calculated from the k-th sample from the posterior of the drift and diffusion functions.

Next, we employ sparse GP regression to build our surrogate log-likelihood surface. We arrange the M inducing points in a fixed uniform grid and set $M = 16$ per parameter dimension; this reduction in the number of inducing points reflects the additional computational burden arising from performing GP regression in more dimensions.

Lastly, for our MCMC scheme we choose uniform priors $\pi \sim U(0, 10)$ on all parameters to infer and we set the random refinement criterion to 10^{-4}.

3.7 Results

We demonstrate the performance of our method for both two-dimensional and three-dimensional inference using three parameter sets and show that we can accurately infer the interaction radius δ of the model, along with the interaction strength α and, in 3-dimensions, the level of noise η.

The first parameter set is defined on a group of $N = 20$ individuals with parameter values $\alpha = 0.3,\ \delta = 2$ and a fixed (assumed known) value of $\eta = 0.25$. The second parameter set is specified on a group of $N = 100$ individuals with parameters $\alpha = 0.8,\ \delta = 1,\ \eta = 1$ (again η is held fixed). For the third parameter set we infer all three parameters for a group size of $N = 30$. The true parameters are $\alpha = 0.6$, $\delta = 1.5$ and $\eta = 0.5$.

We then use our framework to infer the posterior distributions for α and δ, which are shown in Figs. 3.3a and 3.3b and Figs. 3.4a and 3.4b as well as for all three parameters α, δ and η, shown in Figs. 3.5a, 3.5b and 3.5d. The refined, surrogate log-likelihood surface for

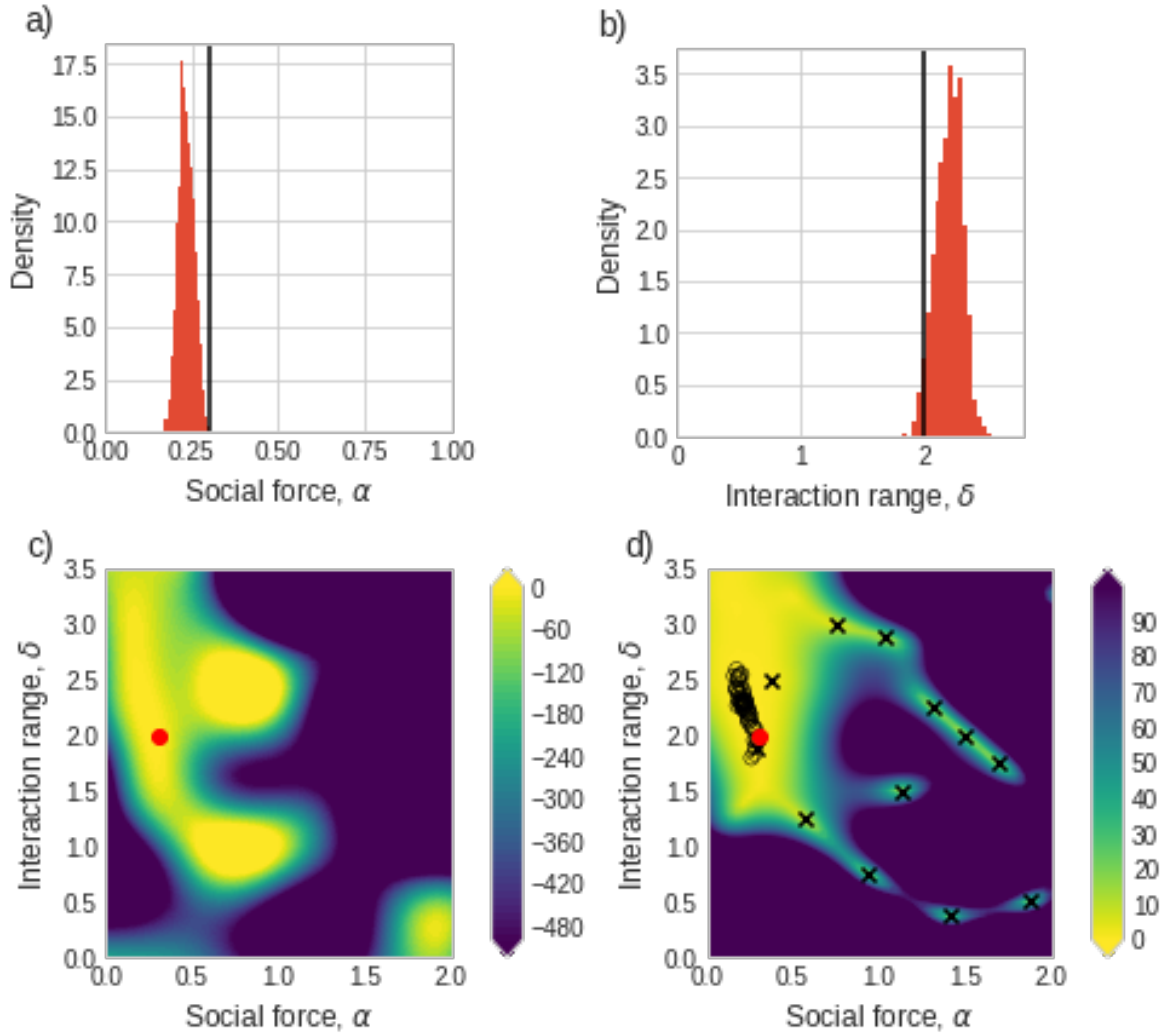


Figure 3.3: Inference results for $N = 20$, $\alpha = 0.3$, $\delta = 2$. Results shown for 10,000 samples after burn-in of 40,000 (Geweke's diagnostic (Geweke, 1991) was used to test convergence: highest absolute z-score was 0.73 for α and 0.89 for δ). a) Posterior distribution of the weighting given to social cues (α). Vertical bar represents the true parameter value. b) Posterior distribution for the interaction range (δ). Vertical bar represents the true parameter value. c) Refined surrogate log-likelihood; the true parameter value is indicated by the red, full circle. d) Variance associated with the refined surrogate log-likelihood. Black crosses show the first initial points from our space-filling design; black open circles show every tenth refinement. The more explored regions have lower uncertainty.

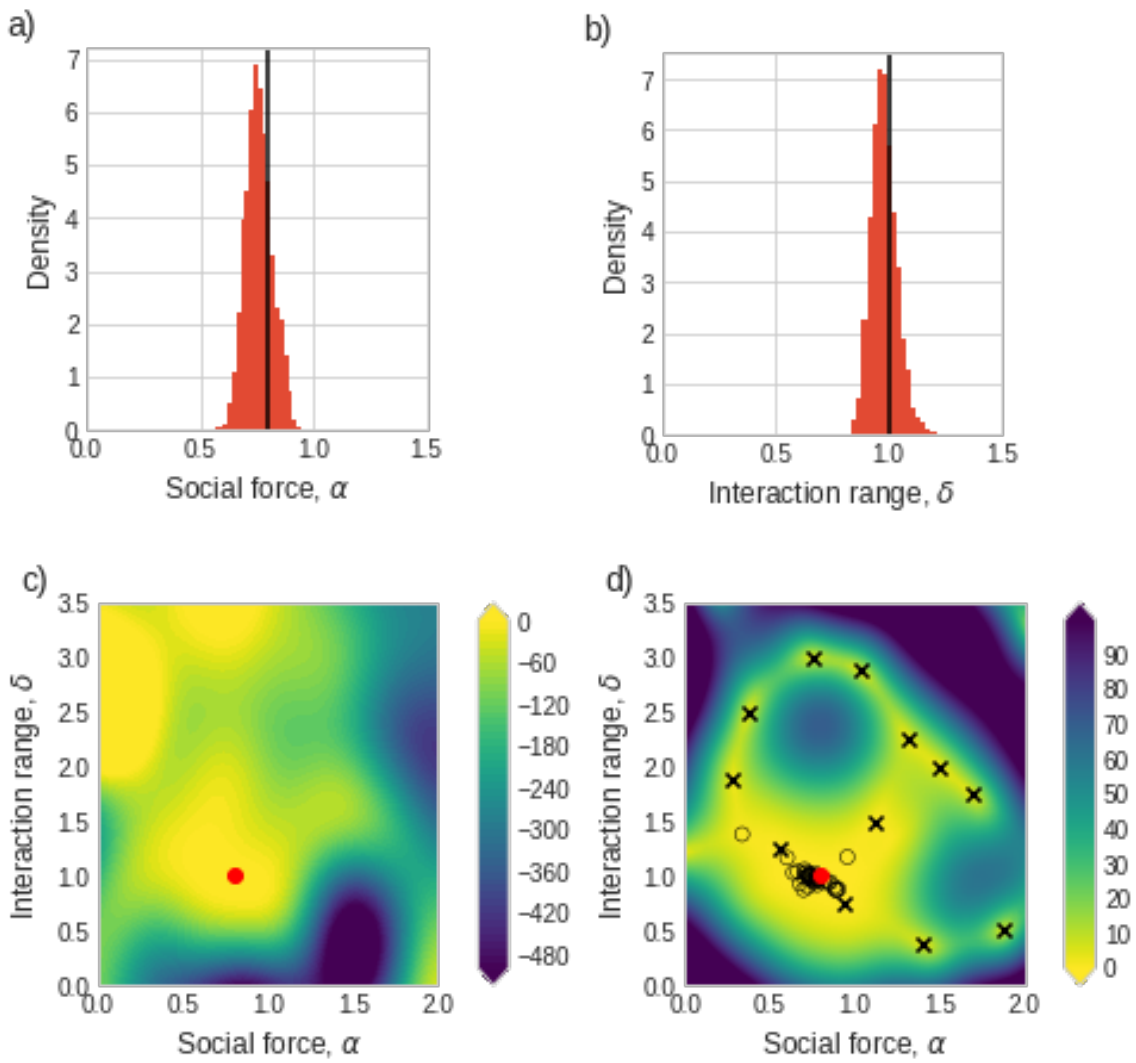


Figure 3.4: Inference results for $N = 100$, $\alpha = 0.8$, $\delta = 1$. Results shown for 10,000 samples after burn-in of 20,000 (Geweke's diagnostic (Geweke, 1991) was used to test convergence: highest absolute z-score was 0.43 for α and 0.34 for δ). a) Posterior distribution of the weighting given to social cues (α). Vertical bar represents the true parameter value. b) Posterior distribution for the interaction range (δ). Vertical bar represents the true parameter value. c) Refined surrogate log-likelihood; the true parameter value is indicated by the red, full circle. d) Variance associated with the refined surrogate log-likelihood. Black crosses show the first initial points from our space-filling design; black open circles show every tenth refinement. For larger population size there is lower uncertainty in the simulation output so less refinement is required.

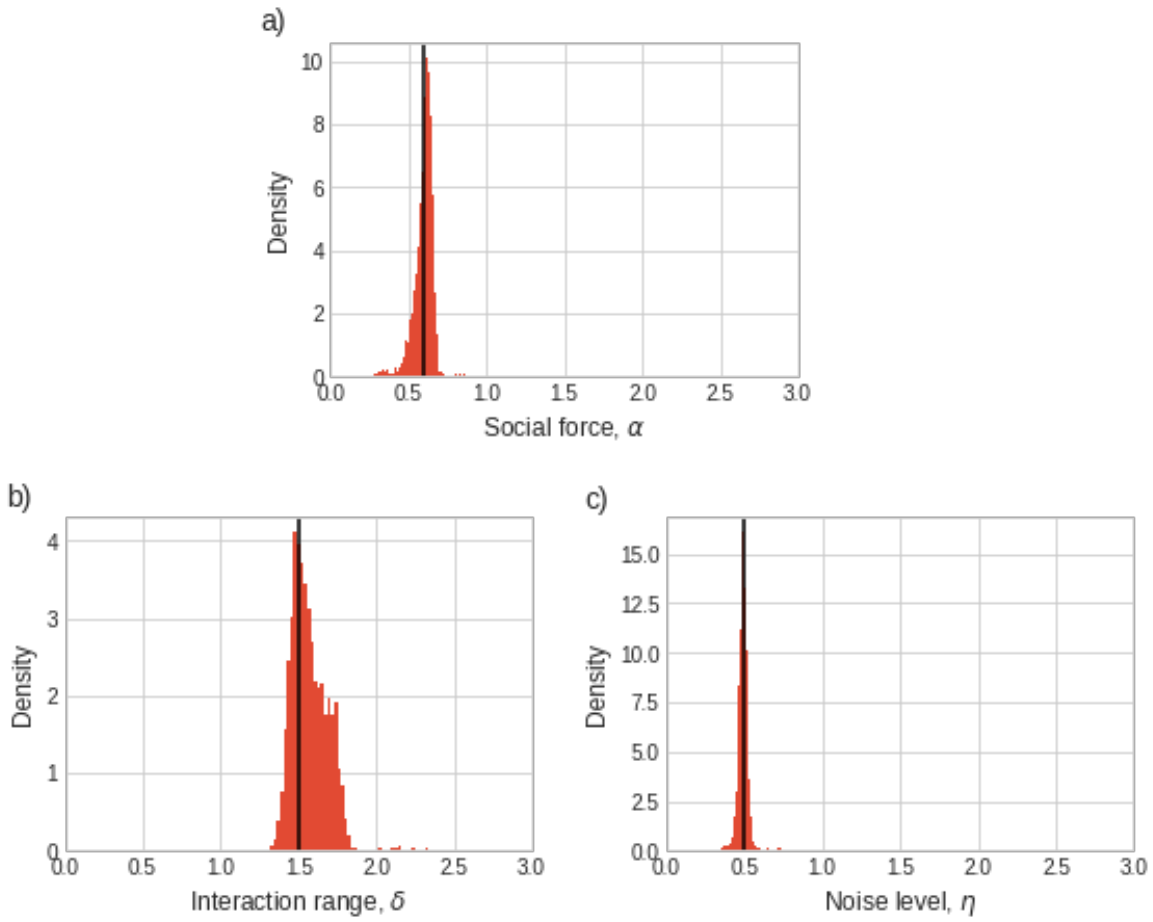


Figure 3.5: Inference results for $N = 30$, $\alpha = 0.6$, $\delta = 1.5$ and $\eta = 0.5$. Results shown for 20,000 samples after burn-in of 30,000 (Geweke's diagnostic (Geweke, 1991) was used to test convergence: highest absolute z-score was 0.67 for α, 0.59 for δ and 0.39 for η). a) Posterior distribution of the weighting given to social cues (α). b) Posterior distribution for the interaction range (δ). c) Posterior distribution for the noise term (η). The vertical bars show the true parameter values.

the parameter sets is shown in Figs. 3.3c and 3.4c and the associated uncertainty in Figs. 3.3d and 3.4d. By incorporating the uncertainty in the log-likelihood into the emulator the sampler is able to run forward simulations only where needed and focus on refinement in regions of high likelihood.

3.8 Discussion

We have presented a new statistical framework for inference of microscale parameters from macroscale measurements of interacting systems. By employing sparse Gaussian process regression we have by-passed the need for a formal link between scales and obtained approximations of the probability density of macroscale observations, simultaneously calculating

the associated uncertainty caused by the use of a finite number of microscale simulations. This allows us to construct a fast, adaptive MCMC sampler that employs a second Gaussian process to emulate the log-likelihood surface.

While Gaussian process regression has been shown to be an effective method for learning drift and diffusion functions of stochastic differential equations (Batz et al., 2018), our approach presents a novel application of multi-latent Gaussian processes (Saul et al., 2016) in this context. Previous work in this area (Batz et al., 2018, Campioni et al., 2020) has employed a separate Gaussian process to learn the diffusion function directly from simulation outputs as specified in Eqn. (3.15), or by using a parametric function for the diffusion (Batz et al., 2018). As stated earlier, the first approach leads to a bias in the posterior distribution of the diffusion function due to finite sampling rates, while the second approach neglects uncertainty. As the diffusion function appears in the denominator of the stationary probability density defined in Eqn. (3.7), both these effects will significantly impair the estimate of the likelihood of microscale parameters given macroscale observations. We have overcome these issues by employing a multi-latent variational approach (Saul et al., 2016) that learns the drift and diffusion functions simultaneously and is able to quantify uncertainty in the diffusion as well as providing an unbiased estimator of both functions.

For effective sampling from the posterior, we employ an adaptive Metropolis-Hastings algorithm that is based on Conrad et al. (2016) with several modifications. Notably, we replace the local Gaussian process approximation with a sparse Gaussian process that allows us to use multiple samples from the SPD posterior for each parameter set where microscale simulations are run. By passing these multiple samples into the algorithm, the emulator GP is able to learn an effective observation noise in the simulator that arises due to the stochastic nature of the microscale model.

While we have applied our framework to a simple 1-dimensional simulation model, our approach can be applied to any multiscale system that can be modelled at the microscale but can only be easily observed at the macroscale. Collective animal movement presents one example of such a system, where individual trajectories are often difficult to observe while microscale models are straightforward to simulate (Vicsek and Zafeiris, 2012). However, many models of complex systems, such as models of voter behaviour (Sood et al., 2008), opinion dynamics (Torney et al., 2013) or epidemics (Pokharel and Deardon, 2016), share these characteristics and their microscale dynamics could be inferred from static observations using our proposed method. As for the physical sciences, our method can find applications in the context of ferromagnetism e.g. inverse Ising problem, where the aim is to infer the coupling strength between spins given observed spin correlations, magnetisations or other data (Nguyen et al., 2017), as well as fluid dynamics e.g. the inverse problem of identifying unknown flow conditions from an observed response of the free surface (Sellier, 2016) or the non-invasive estimation of physiological parameters determining the systemic and pulmonary

blood flow (Colebank et al., 2019, Paun et al., 2020). Modelling molecular, cellular and auto-catalytic pattern formation (Meinhardt, 1982) is another application area of the method proposed in the present work.

Future work will extend the framework to higher-dimensional macroscale systems, for which the presented ideas of using GPs to approximate the unavailable stationary probability density of interacting systems should in principle hold.

Chapter 4

State sequence proposal mechanisms for hidden Markov and semi-Markov models

In this chapter, we present various methods aimed at simulating semi-Markov chains. We conceptualised three different proposal mechanisms for semi-Markov chains and we discuss their validity, advantages and drawbacks below. The proposal mechanism will be a fundamental block in the algorithm described in the next chapter.

4.1 Introduction

Hidden Markov models (HMMs) have come to be a reliable and versatile method employed in the behavioural ecology and animal movement community. This is due to the mixture components nature of such models (Zucchini et al., 2009, Robert et al., 2000) through which modelling behaviours becomes relatively simple (with all due modifications that each study may require). For telemetry data covering long time-scales, qualitatively different dynamics will emerge due to changes in the animal behaviour (for example, a change from a resting behaviour to an exploratory behaviour). Thus, it is convenient to assume that observations are dependent on a latent (hidden) behavioural process, called state process, which is described by a Markov process. From here on, states will be used as proxies for biological behaviours. By augmenting the framework with a behaviour-dependent observational process and by leveraging the machinery introduced back in Chapter 2, it is now possible to answer inferential questions concerning the movement parameters, the behavioural process parameters and consequently the history of changes in behaviour.

As far as discrete-time HMMs are concerned, significant contributions in this domain are attributed to Morales et al. (2004). In their publication, they assumed that the movement path of one individual was composed of a pre-specified number k of random walks so that different behavioural patterns could be captured by the quantitatively different random walks (contrary to what was described in Chapter 2 where the movement path of one individual was modelled via one random walk). Each random walk was characterised by an ordered series of step lengths and turning angles $[r_t, \phi_t]$; step lengths and turning angles follow, respectively, a Weibull distribution and a wrapped Cauchy distribution. Here the movement parameters are represented by the Weibull and wrapped Cauchy parameters of each random walk, (a_i, b_i) and (μ_i, ρ_i), where the subscript $i = 0, \ldots, k$ represent a different state. Let y denote the full dataset, then the model likelihood as formulated in Morales et al. (2004) is

$$p(y|a,b,\mu,\rho) = \prod_{t=0}^{T} W(r_t|a_{i_t}, b_{i_t}) C(\phi_t|\mu_{i_t}, \rho_{i_t}), \tag{4.1}$$

where W and C indicate, respectively, the Weibull distribution and the wrapped Cauchy distribution and the notation i_t indicates that the state varies in time. They then estimated the movement parameters by employing MCMC sampling techniques to explore different combinations of $(a_i, b_i, \mu_i, \rho_i)$ and selected the ones associated with the highest likelihood.

Since then, the animal movement community has extensively applied discrete-time HMMs to analyse animal telemetry data. These models split observations into two data streams - step lengths and turning angles - and through applications of the forward algorithm and the Viterbi algorithm (Chapter 2) the model is fit and the most likely behavioural sequence is found. Great effort has been dedicated to the creation of user-friendly packages that perform

inference under HMM assumptions, such as the R packages "moveHMM" (Michelot et al., 2016) and its latest extension "momentuHMM" (McClintock and Michelot, 2018), each able to tackle the inferential problems just discussed and much more (for example, prediction of behavioural sequences at future times).

Working with discrete time is advantageous as the mathematics involved is simpler and easier to implement, however, this formulation comes with limitations. In particular, HMMs are not well-suited for situations involving irregularly sampled data and non-negligible measurement error (Michelot and Blackwell, 2019, Hooten et al., 2017b, Patterson et al., 2017a). At the expense of some computational efficiency, using a continuous-time model offers an effective solution to the problem. In this chapter, we are mainly interested in the application of diffusion processes to model in continuous time the dynamics underlying the animal movement and employing *semi-Markov chains* to model the behavioural process (more on those in the subsequent section).

An example of using diffusion processes to model telemetry data can be found in Blackwell (1997, 2003). Here locations are modelled via a d-dimensional Ornstein-Uhlenbeck (OU) process. Let $\boldsymbol{Y}_t = \{Y_{1,t}, \ldots, Y_{d,t}\}$, then the distribution of its location at time $t+s$ given location at time s is

$$\boldsymbol{Y}_{t+s} | \boldsymbol{Y}_s = \boldsymbol{y}_s \sim N(\boldsymbol{\mu} + e^{\boldsymbol{B}t}(\boldsymbol{y}_s - \boldsymbol{\mu}), \boldsymbol{\Sigma} - e^{\boldsymbol{B}t}\boldsymbol{\Sigma}e^{\boldsymbol{B}'t}), \tag{4.2}$$

where $\boldsymbol{\mu}$ is a d-vector and $\boldsymbol{\Sigma}$ and $\boldsymbol{B}$ are $d \times d$ matrices; the matrix $\mathbf{B}$ controls the strength and form of the centralising tendency. Since it is assumed that $\mathbf{B}$ is stable, that is, $e^{\boldsymbol{B}t} \to 0$ as $t \to \infty$, often regarded as part of the definition of OU process (Blackwell, 1997), then the limiting distribution of the location is a normal distribution with mean location vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$\mathbf{Y}_t \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{4.3}$$

Throughout the chapter, we will restrict our focus on scalar OU model parameters. While this process is Markovian, it does not however account for persistence in speed and direction of the movement; given that animals have inertia and therefore move at a similar rate over successive time steps, a more natural representation for animal movement involves continuous-time correlated random walks (CTCRWs), as introduced by Johnson et al. (2008), in that they account for autocorrelation of movement.

In their publication, they presented a novel modelling framework that integrates a CTCRW model with a state-space model. For the CTCRW model, the velocity $\boldsymbol{v}(t)$ is modelled via a 2-dimensional OU process defined as (Johnson et al., 2008)

$$v_c(t+\Delta) = \gamma_c + e^{-\beta\Delta}(v_c(t) - \gamma_c) + \zeta_c(\Delta), \tag{4.4}$$

where the subscript $c = 1, 2$ represents the coordinate axes, γ_c is the mean velocity, β is an

autocorrelation (scalar) parameter and ζ is a zero mean normal error with variance

$$\sigma^2(1-e^{-2\beta\Delta})/2\beta. \tag{4.5}$$

The location process is then derived from the velocity process by integration. This formulation of the model is also referred to as an integrated OU process, to emphasise that observations are modelled through velocities. Hence the location $\boldsymbol{\mu}$ is

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu}(0) + \int_0^t \mathbf{v}(u)du. \tag{4.6}$$

Johnson et al. (2008) integrate the CTCRW model in a state-space formulation so that measurement error is included and standard methods such as the Kalman filter can be applied to recover the OU process parameters. A general state-space model requires two equations to be defined - the observation equation and the system equation (Chapter 2). For observation $\boldsymbol{y}_{t_i} = [y_{1t_i}, y_{2t_i}]$ and true location $\boldsymbol{\mu}(t) = [\mu_x(t), \mu_y(t)]$, the observation equation is straightforward (Johnson et al., 2008)

$$y_c(t) = \mu_c(t) + \varepsilon_c, \quad \varepsilon_c \sim N(0, H_c), \tag{4.7}$$

where H_c is the measurement error variance.

On the other hand, the system (true) process presents a problem: because the location is an integrated process, the location process lacks the Markov property as it depends on all previous velocities, contrarily to the velocity process which is Markovian (by definition, Eqn. 4.4). Consequently, the true location process is constructed by combining the velocity process with the location process to form a Markov process. By using Eqn. 4.4, the true location process is defined in terms of the true location $\boldsymbol{\mu}(t)$ and the velocity $\boldsymbol{v}(t)$ (Johnson et al., 2008):

$$\mu_c(t+\Delta) = \mu_c(t) + v_c(t)\left(\frac{1-e^{-\beta\Delta}}{\beta}\right) + \xi_c, \tag{4.8}$$

where Δ is the time interval, ξ_c are zero-mean normal errors with variance (Johnson et al., 2008)

$$\frac{\sigma^2}{\beta^2}\left(\Delta - \frac{2}{\beta}(1-e^{-\beta\Delta}) + \frac{1}{2\beta}(1-e^{-2\beta\Delta})\right). \tag{4.9}$$

For SSM specification, the covariance between the true location error ξ_c and the velocity

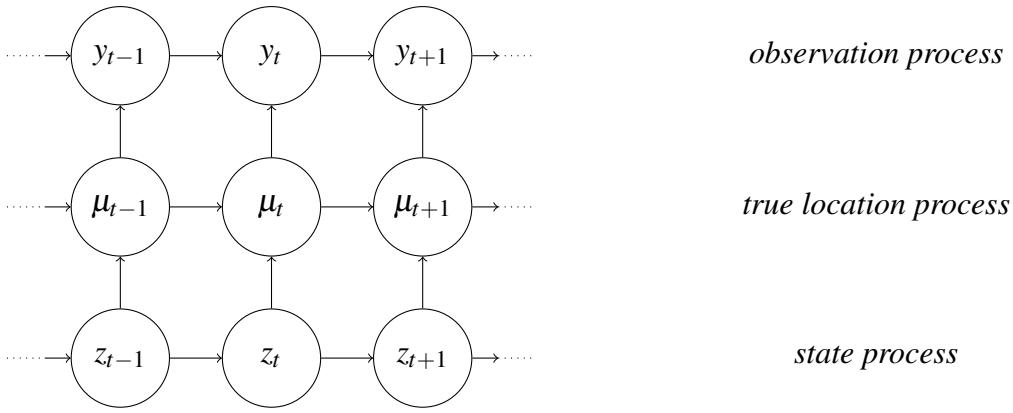


Figure 4.1: Illustration of the structure of the three-layer process introduced in Michelot and Blackwell (2019). Note that because the transitions occur in continuous-time, they are not restricted to occur at the time of observations.

error ζ_c is needed (Johnson et al., 2008):

$$C[\xi_c, \zeta_c] = \frac{\sigma^2}{2\beta^2}\left(1 - 2e^{-\beta\Delta} + e^{-2\beta\Delta}\right). \tag{4.10}$$

This state-space model presents two layers, the hidden one being the true location combined with the velocity process and the observed one being the location together with measurement error. This framework has been extended to incorporate a multi-state model by Michelot and Blackwell (2019). A third, hidden layer representing a state process is added and the observations are now assumed to be state-dependent. The state process is governed by a continuous-time Markov chain characterised by an infinitesimal generator matrix (IGM) $\mathbf{\Lambda}$; switches can occur at any point in time but each state can only take one discrete value $0, \ldots, k$ per time. A graph of the structure is illustrated in Fig. 4.1

Because this is a state-space model, the likelihood conditional on the hidden state sequences (Chapter 2) is readily available using the Kalman filter. However, the OU process parameters are state-dependent and knowledge of state needs to be inferred. Michelot and Blackwell (2019) adopt the following inferential strategy, based on that found in Blackwell (2003). They employ a Metropolis-within-Gibbs sampling scheme (Blackwell, 2003, Michelot and Blackwell, 2019); each iteration of the algorithm consists of an update for three groups: the first update concerns the underlying state sequence, which is used in the second update for the OU parameters β_i and σ_i, for $i = 0, \ldots, k$, and the final update concerns the transition rates of the IGM. The first two groups make use of the likelihood of the model, given by the Kalman filter, conditioned on the state sequence $\mathbf{S}$ and the OU parameters $\boldsymbol{\theta}$, $p(\boldsymbol{y}|\boldsymbol{\theta}, S)$. For the first group, a new state sequence $\mathbf{S}^+$ is generated from the current state sequence $\mathbf{S}^-$ and either accepted or rejected based on the ratio of the likelihoods (Blackwell,

2003):

$$min\left\{1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}, \mathbf{S}^{+})}{p(\boldsymbol{y}|\boldsymbol{\theta}, \mathbf{S}^{-})}\right\}; \tag{4.11}$$

the second group uses the accepted state sequence $\mathbf{S}^{*}$ and through a proposal density $Q(\cdot|\cdot)$ new OU parameters $\boldsymbol{\theta}^{+}$ are proposed. These are either accepted or rejected based on the following Metropolis-Hastings acceptance rate:

$$min\left\{1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}^{+}, \mathbf{S}^{*})p(\boldsymbol{\theta}^{+})Q(\boldsymbol{\theta}^{-}|\boldsymbol{\theta}^{+})}{p(\boldsymbol{y}|\boldsymbol{\theta}^{-}, \mathbf{S}^{*})p(\boldsymbol{\theta}^{-})Q(\boldsymbol{\theta}^{+}|\boldsymbol{\theta}^{-})}\right\}, \tag{4.12}$$

where $p(\boldsymbol{\theta})$ is the OU parameters prior.

The third group makes use of conjugate priors as in Blackwell (2003) to recover the transition rates, leveraging the fact that the residence times in each state are exponentially distributed.

To propose new state sequences, Michelot and Blackwell (2019) employ the endpoint-conditioned methods from Hobolth and Stone (2009). Such methods are usually used to modify existing sequences by generating new Markov chains between known initial and end points while the remaining part of the sequence is unaltered. In Hobolth and Stone (2009), three algorithms are analysed that ensure that transitions occur between the selected initial and end points. The first algorithm is the modified rejection sampling. This follows a simple algorithm called the forward algorithm, which involves sampling switch times from an exponential distribution and allowing constant state sequences. Specifically, given a chain $\{X(t) : 0 \leq t \leq T\}$ conditional on $X(0) = a$ and $X(T) = b$ and its corresponding IG matrix $\boldsymbol{\Lambda}$, a switch time $\tau \sim Exp(1/\lambda_a)$ is sampled and:

$$\begin{cases} \text{if } \tau \geq T,\ X(t) = a\ \forall t \in [0, T] \\ \text{if } \tau < T,\ X(\tau) = c \text{ with probability} \lambda_{ac}/\lambda_a;\ \text{repeat.} \end{cases} \tag{4.13}$$

The modified rejection sampling algorithm ensures that whenever the ending state is different to the initial state, a least one switch occurs with the following density

$$f(\tau) = \frac{\lambda_a e^{-\tau\lambda_a}}{1 - e^{-T\lambda_a}}, \tag{4.14}$$

so as to avoid large sample rejection rate due to the forward sampling scheme being bound to fail for small time intervals (Hobolth and Stone, 2009).

The second algorithm is called direct sampling and it is based on the assumption that the infinitesimal generator matrix admits an eigenvalue decomposition. From this assumption, it is possible to recover the probabilities of switching to a specific state i, as well as the time of

the switch, in the case of a switch occurring between $X(0)$ and $X(T)$. We omit the expressions and invite the reader to consult the original manuscript (Hobolth and Stone, 2009).

The third algorithm is called uniformisation and it begins by sampling the number of changes uniformly on the time interval $[0,T]$. If the number is either 0 or 1 with $X(0) = X(T)$ then it is going to be a constant path; if it's 1 with $X(0) \neq X(T)$, the waiting time is sampled from a uniform distribution on $[0,T]$. For any number n of switch points $n \geq 2$, n points are sampled uniformly on $[0,T]$ and a discrete Markov chain is employed to simulate the transitions between the sampled switch points conditioned on $X(0) = a,\ X(T) = b$. The transition probability matrix characterising the discrete Markov chain is obtained through the infinitesimal generator matrix via the following equation (Hobolth and Stone, 2009):

$$\boldsymbol{P} = \mathbf{I} + \frac{1}{\lambda_{max}}\boldsymbol{\Lambda}, \tag{4.15}$$

where $\lambda_{max} = max_c\ \lambda_c$, for state c.

4.2 Introducing non-Markovian switching times

Thus far we have given an overview on some of the methods found in the literature that are used in the context of switching behaviour problems in continuous-time. Specifically, we have restricted our focus on continuous-time correlated random walk models that model telemetry data as an integrated OU process whose parameters are dependent on an underlying continuous-time behavioural process modelled via a hidden Markov model.

However, the Markov property implies underlying assumptions on the behavioural process that are non-realistic. In the absence of external variables that influence the switching behaviour, the residence (or sojourn, dwell) time in each behavioural state follows an exponential distribution (or geometric, in the discrete-time counterpart), for which short and frequent state changes are favoured. This implies that the behavioural process inherits the "memoryless" property of the exponential distribution. Let s be the time spent in a state and let t be the further time that the chain will remain in the same state. Then if X is the random variable representing the time at which the chain leaves the current state, the "memoryless" property implies

$$p(X > t+s | X > s) = p(X > t), \tag{4.16}$$

meaning the sojourn times are independent of the amount of time spent in a state. Thus, when using HMMs it is assumed that the amount of time an animal will remain in a behaviour of, say, resting does not depend on how long the animal has already rested for.

It is also worth adding that current approaches described above are not scalable to large datasets due to the computational burden of standard schemes such as Markov chain Monte

Carlo sampling for which evaluation of the full dataset is required at every step of the sampler. These methods therefore become prohibitively expensive to apply to long-term, high-frequency telemetry datasets.

Therefore, our goal has been to propose an alternative method to analyse telemetry data that can overcome the intrinsic limitations of standard HMMs models. Similarly to Michelot and Blackwell (2019), in our framework telemetry data are modelled via two processes. The first process is an integrated OU process (Johnson et al., 2008, Michelot and Blackwell, 2019) that models the evolution of the location dynamics; the second process is an underlying continuous-time process included to extend the framework to a multi-state model. However, contrary to Michelot and Blackwell (2019), hidden states are not modelled via a continuous-time Markov chain but we instead employ a continuous-time semi-Markov chain that enables us to control the sojourn times. In particular, in our study we model the residence times with a gamma distribution parameterised via a shape parameter α and a mean residence time parameter m, $\Gamma\left(\alpha, \frac{\alpha}{m}\right)$. We believe that this choice makes the model more biologically realistic for two reasons. Firstly, we have now effectively introduced a memory in the process: the more an animal spends time in a behavioural state, the more likely it is to change. Secondly, as shown in Fig. 4.3, shorter sojourn times are now more likely to be rejected.

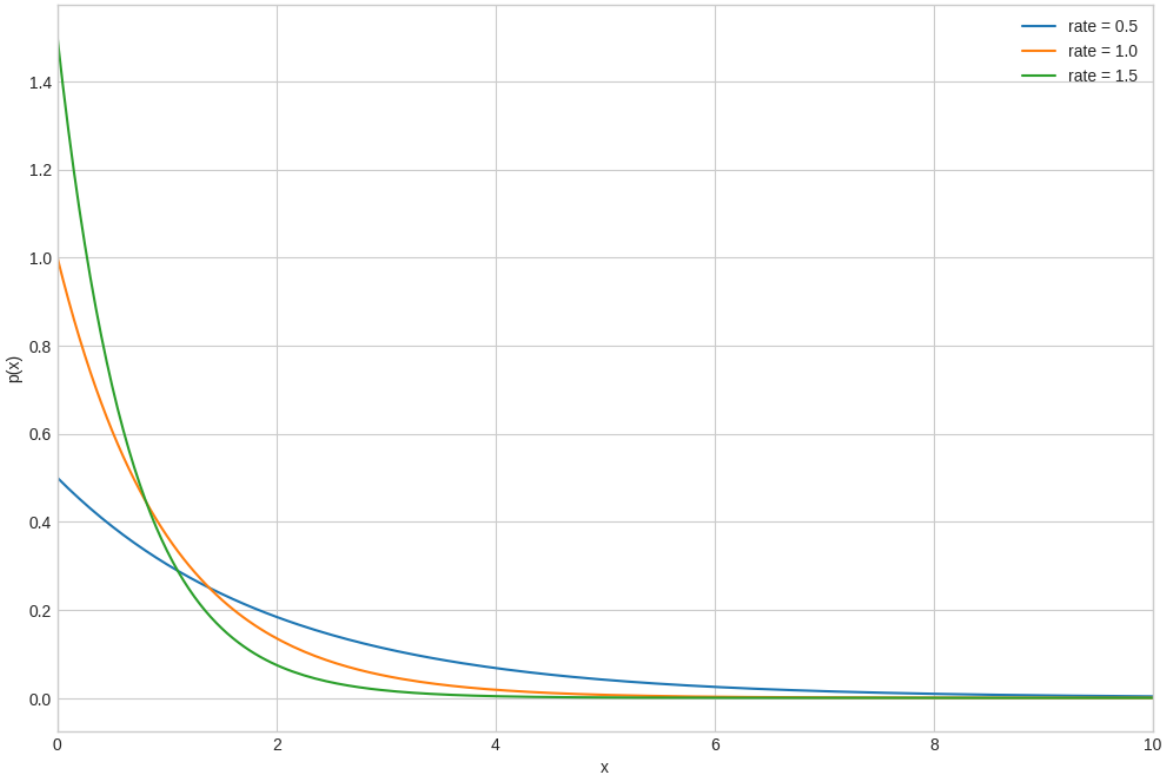


Figure 4.2: Exponential distribution for different rate values. Note how quickly the function decays regardless of the rate values.

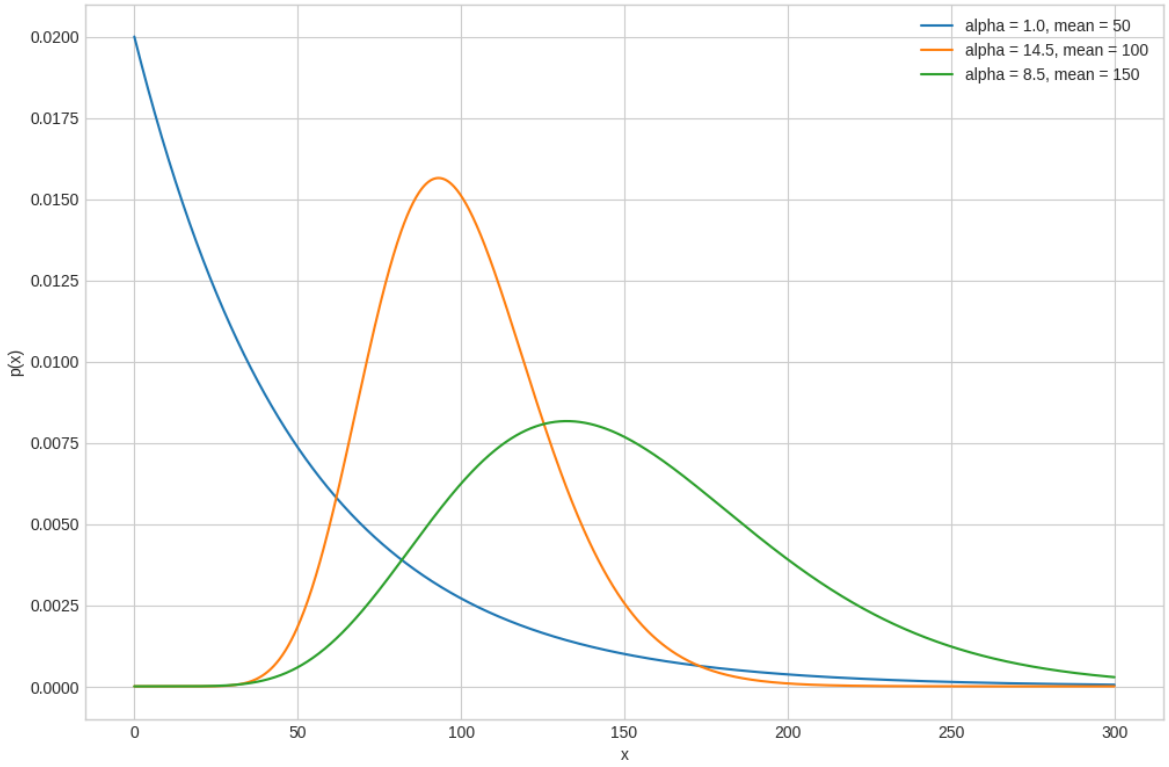


Figure 4.3: Gamma distribution for different shape and mean values. For $\alpha = 1$, the gamma distribution is an exponential distribution with rate $\frac{\alpha}{m}$.

While our method offers a different inferential algorithm that differentiates it from the methods explained above, which is based on a Monte Carlo Expectation-Maximisation (MCEM) algorithm, we defer the details until Chapter 5 and here we focus instead on the semi-Markov chains proposal mechanism. Indeed, we cannot use the endpoint-conditioned methods outlined in Hobolth and Stone (2009) as they rely on exponentially distributed residence times. The scope of this chapter is to illustrate the different algorithms that we have created to simulate continuous-time semi-Markov chains - this algorithm is the fundamental block for the methodology that we introduce in the next chapter. The main challenges that we have faced during this project concern both the computational efficiency of the algorithm and the requirements to satisfy detailed-balance. To make the algorithm efficient, we needed to introduce a degree of similarity between successive samples in order to avoid exploration high rejection rates. As for the second challenge, detailed-balance is ensured either by using a symmetric proposal mechanism or by including the Hastings factor in the acceptance rate. As we will see throughout this chapter, these two conditions were not trivial to meet. We have studied three different continuous-time semi-Markov chains generating mechanisms: below is a description of each method.

4.3 Semi-Markov state sequence proposal mechanisms

As stated before, the dwell-time distribution of a semi-Markov chain is arbitrarily defined and it is not restricted to follow an exponential distribution as in the case for Markov chains. Specifically, a semi-Markov chain is approximated by an embedded Markov chain $\{X_n; n \geq 0\}$ with a finite or countably infinite state space, a transition probability matrix $\boldsymbol{P}$ and a sequence $\{U_n; n \geq 1\}$ of holding intervals between state transitions (Zucchini et al., 2009). The times at which state transitions occur are then given, for $n \geq 1$, as

$$S_n = \sum_{j=1}^{n} U_j. \tag{4.17}$$

The semi-Markov process is then the continuous-time process $\{X(t); t \geq 0\}$ where, for each $n \geq 0$, $X(t) = X_n$ for $t \in [S_n \leq X_n < S_{n+1}]$.

This approximation of semi-Markov chains through an embedded Markov chain is particularly useful in the application of hidden semi-Markov models (HSMMs), in that it allows to employ the well-established methods for HMMs (Zucchini et al., 2009) (Chapter 2). In particular, the (approximate) likelihood of a sequence of observations $x_1, \ldots, x_T$ is given by (Zucchini et al., 2009):

$$\mathscr{L}_T = \delta \mathbf{\Omega}(x_1)\mathbf{P}\mathbf{\Omega}(x_2)\mathbf{P}\ldots\mathbf{P}\mathbf{\Omega}(x_{T-1})\mathbf{P}\mathbf{\Omega}(x_T)\mathbf{1}', \tag{4.18}$$

where δ is the initial distribution of the approximating Markov chain and

$$\mathbf{\Omega}(x) = diag(p_1(x), \underbrace{\ldots}_{k_1 \text{ times}}, p_1(x), \ldots, p_n(x), \underbrace{\ldots}_{k_n \text{ times}}, p_n(x)), \tag{4.19}$$

where, for $i = 1, \ldots, n$, p_i is the state-dependent distribution and $k_i > 0$ is the number of successive observations in the same state (for example, given the sequence 111223333 we have $k_1 = 3$, $k_2 = 2$ and $k_3 = 4$).

A question of interest regards the initial distribution δ of the embedded Markov chain. A standard approach is to assume that the first time point of the considered time series corresponds to a switchpoint, so that it is easier to model the distribution of the first dwell time (Zucchini et al., 2009). While this assumption is not expected to significantly impact parameter estimation, except for short series, its validity varies across applications. More importantly, the compelled state transition at the series' outset poses a challenge to the overall stationarity of the HSMM (Zucchini et al., 2009).

However, we can circumvent the assumption of a state switch at the series' outset by allowing non-zero initial state probabilities δ. Furthermore, we could fit stationary HSMMs by taking the initial distribution δ to be the solution to the linear equation $\delta = \delta\mathbf{P}$ (for more

specific details, refer to Zucchini et al. (2009)).

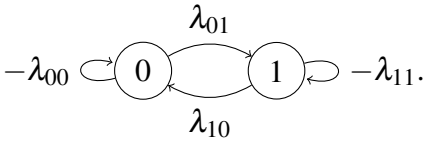
Having said that, we are now going to describe the semi-Markov chain sequence generating methods that we have conceptualised. As aforementioned, for every method we need to take into account detailed-balance and reversibility of any move. From a computational efficiency prospective, it is also important to introduce a degree of similarity between any two consecutive samples, as in Hobolth and Stone (2009). Throughout the chapter, we assume as prior belief that the residence time in each state follows a gamma distribution $\Gamma\left(\alpha, \frac{\alpha}{m}\right)$, where α is the shape parameter and m is the mean residence time. However, the residence time distribution can be chosen by the practitioner.

4.3.1 Virtual state method

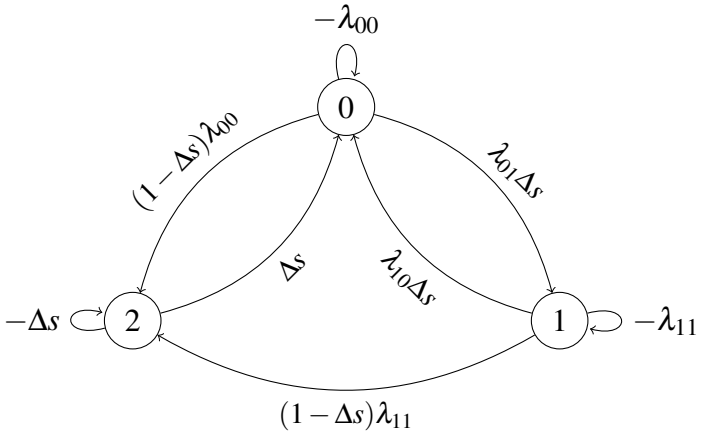
In this first method, given an $n-$state model we introduce a virtual state that copies previous states so that a degree of similarity is introduced to avoid exploration of the whole sample space. Given that we rely on the TensorFlow package HMM, and hence as with all packages the output will be a Markov chain, introducing the virtual state also allows us to increase the duration of the sojourn times in the effort to create semi-Markov chains. Introducing a virtual state is achieved by specifying an infinitesimal generator matrix $\mathbf{\Lambda}$ at every point in the time series and augmenting it with a virtual state that does not affect the total number of observable states in the original state-space model configuration. Before expanding this point further and diving into the mathematical details, it may be better to give an example.

Let's assume the observations can be described by a two-state model where we label the states as 0 and 1. We begin our sampling routine by constructing an initial state sequence over a given time series T by defining its associated 2×2 IGM $\mathbf{\Lambda}$ at every point in T. We then employ the HMM TensorFlow package to generate the first state sequence sample, $\mathbf{S}^0$. In order to generate a candidate state sequence sample $\mathbf{S}^*$, from $\mathbf{S}^0$, we augment $\mathbf{\Lambda}$ as follows. It is important to note that the augmentation process is applied at every time point.

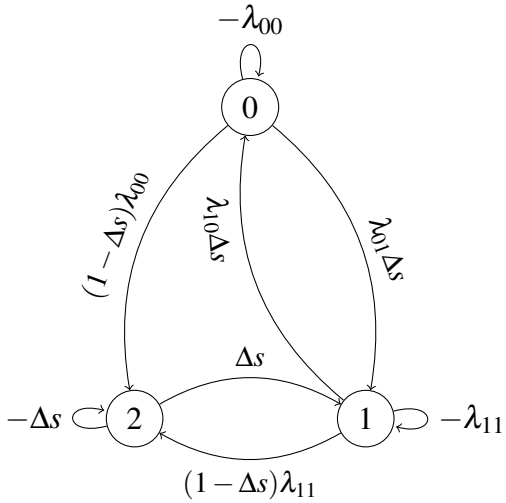
Suppose that at time $t = k$ the observations are in state 0 and that the corresponding IGM for $\mathbf{S}^0$ is described by the following transition rates diagram:



We denote the additional virtual state as 2 and we define the new transition rates so that at time k it can be transitioned to from any state but it can only transition to state 0 with rate Δs. The corresponding transition rates diagram will look like this:



Similarly, if the observations lie in state 1 at time $t = j$ the corresponding augmented IG matrix will be:



Having specified an augmented $\mathbf{\Lambda}$ over the whole time domain, we can employ the HMM package to generate $\mathbf{S}^*$. If the new chain is in the virtual state at time t, then we copy the state value from $\mathbf{S}^0$ at time t. This copying mechanism is what ensures a degree of similarity between samples. We illustrate the generating state mechanism in Fig. 4.4.

For the mathematical details, the augmentation process can be described as a two-step process: the first step is to specify the rates into the virtual state, the second step is to add the rates out of the virtual state. Given a general $n \times n$ IGM

$$\mathbf{\Lambda} = \begin{pmatrix} -\lambda_{00} & \dots & \lambda_{0(n-1)} \\ \dots & \dots & \dots \\ \lambda_{n0} & \dots & -\lambda_{(n-1)(n-1)} \end{pmatrix}, \tag{4.20}$$

where $\forall i \in 0, \dots, n-1,\ \lambda_{ii} = \sum_{i \neq j} \lambda_{ij}$ with associated state sequence $\mathbf{S}^0$, a step size variable δ transformed to be in the interval $(0,1)$, $\Delta s = 1 - exp(-\delta)$, and a time point t we modify the existing rates through the following equation:

$$\mathbf{\Lambda} \circ (\boldsymbol{I}(1-\Delta s) + \Delta s), \tag{4.21}$$

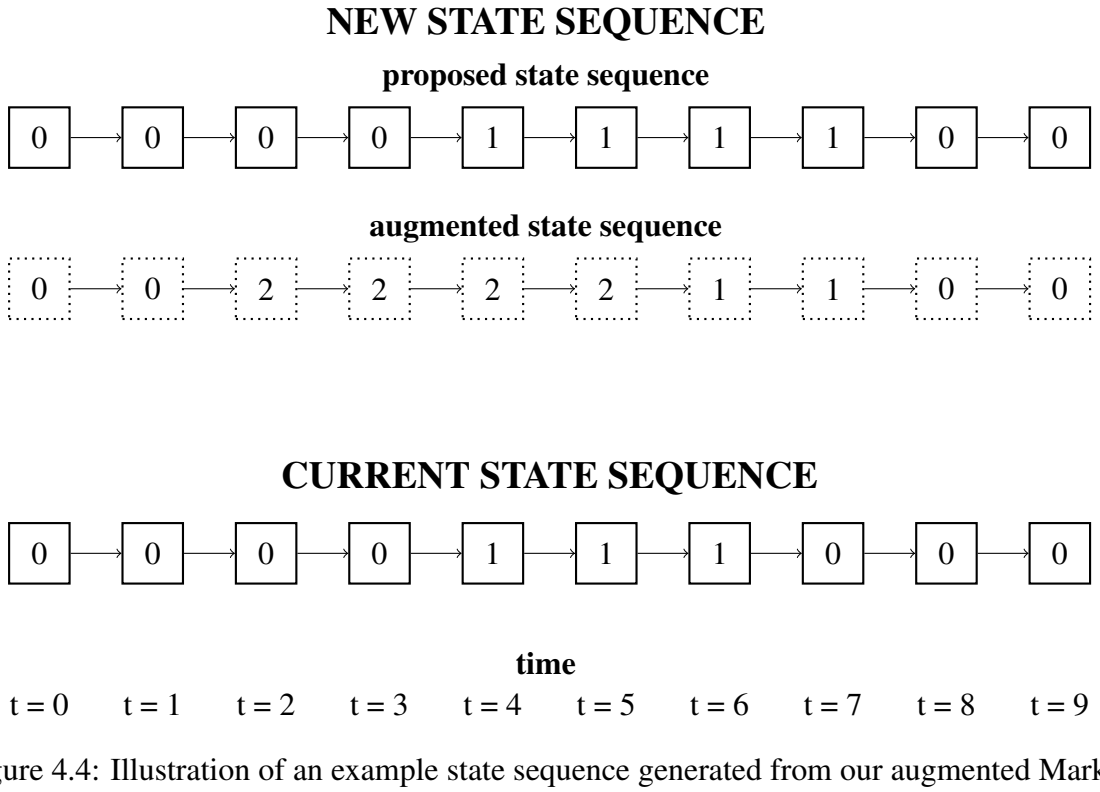


Figure 4.4: Illustration of an example state sequence generated from our augmented Markov chain. The virtual state copies values from the original state sequence.

where the symbol $\circ$ stands for element-wise multiplication. We define the rates into the virtual state as a $n \times 1$ vector

$$\begin{pmatrix} \lambda_{00}(1-\Delta s) \\ \dots \\ \lambda_{(n-1)(n-1)}(1-\Delta s) \end{pmatrix} \tag{4.22}$$

which is then appended to $\mathbf{\Lambda}$; this is the first step and we have created an $n \times (n+1)$ matrix.

The rates out of the virtual state are Δs and are stored in a $1 \times n+1$ vector

$$\begin{pmatrix} c_0 & \dots & c_{n-1} & -\Delta s \end{pmatrix}, \tag{4.23}$$

where, for $i = 0, \dots, n-1$,

$$c_i = \begin{cases} \Delta s, \text{ if } \mathbf{S}^0(t) = i, \\ 0, \text{ otherwise.} \end{cases} \tag{4.24}$$

Once the vector is concatenated to $\mathbf{\Lambda}$, we obtain the full $(n+1) \times (n+1)$ matrix.

Now that an overview of the method has been given, there are two caveats to discuss. The first thing to notice is that the HMM package is built on discrete time and therefore must be given a probability transition matrix $\boldsymbol{P}$ to work with. Given that we have defined a continuous-time model, we employ a first order approximation given by Bogdan Doytchinov

and Rachel Irby (2010):

$$\boldsymbol{P} = \boldsymbol{I} + \boldsymbol{dt}\boldsymbol{\Lambda}, \tag{4.25}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{dt}$ is the vector of time differences.

The second matter concerns the Hastings ratio. Our proposal mechanism is not symmetric, therefore the Hastings ratio must be included in the acceptance ratio. The proof is the following example.

Suppose we start from an initial state sequence $\mathbf{S}^- = [0\ 1\ 0]$ with associated IGM $\boldsymbol{\Lambda} = \begin{pmatrix} -\frac{1}{m} & \frac{1}{m} \\ \frac{1}{m} & -\frac{1}{m} \end{pmatrix}$. We begin our augmentation routine by first defining an initial distribution $\pi_0 = (\frac{\Delta s}{2}, \frac{\Delta s}{2}, 1-\Delta s)$; then the IG matrices for time $t = 1, 2$ are

$$\begin{aligned} t = 1,\ & \begin{pmatrix} -\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & -\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ 0 & \Delta s & -\Delta s \end{pmatrix}, \\ t = 2,\ & \begin{pmatrix} -\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & -\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ \Delta s & 0 & -\Delta s \end{pmatrix}. \end{aligned} \tag{4.26}$$

By employing Eqn. 4.25 with constant $dt = 1$, the approximate probability transition matrices are

$$\begin{aligned} t = 1,\ & \begin{pmatrix} 1-\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & 1-\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ 0 & \Delta s & 1-\Delta s \end{pmatrix}, \\ t = 2,\ & \begin{pmatrix} 1-\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & 1-\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ \Delta s & 0 & 1-\Delta s \end{pmatrix}. \end{aligned} \tag{4.27}$$

Let's assume that the generated sequence is $\mathbf{S}^* = [1\ 1\ 0]$. From the IG matrices, we can read

that the transition rates, from which the transition probabilities can be extracted, are:

$$\begin{aligned} P(\mathbf{S}^- \to \mathbf{S}^*) = P([0\ 1\ 0] \to [1\ 1\ 0]) + P([0\ 1\ 0] \to [1\ 2\ 2]) + \\ P([0\ 1\ 0] \to [1\ 1\ 2]) + P([0\ 1\ 0] \to [1\ 2\ 0]) = \\ \frac{\Delta s}{2}\left(1-\frac{1}{m}\right)\frac{1}{m}\Delta s + \frac{\Delta s}{2}\frac{1}{m}(1-\Delta s)\,(1-\Delta s) + \\ \frac{\Delta s}{2}\left(1-\frac{1}{m}\right)\frac{1}{m}(1-\Delta s) + \frac{\Delta s}{2}\frac{1}{m}(1-\Delta s)\,\Delta s = \\ \frac{\Delta s}{2m}\left(2-\Delta s-\frac{1}{m}\right). \end{aligned} \tag{4.28}$$

Now we'll calculate the backward rates. Starting from $\mathbf{S}^* = [1\ 1\ 0]$, and assuming the same initial distribution, the probability transition matrices for time $t = 1,\ 2$ are the same as before:

$$\begin{aligned} t = 1,\ \begin{pmatrix} 1-\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & 1-\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ 0 & \Delta s & 1-\Delta s \end{pmatrix}, \\ t = 2,\ \begin{pmatrix} 1-\frac{1}{m} & \frac{1}{m}\Delta s & \frac{1}{m}(1-\Delta s) \\ \frac{1}{m}\Delta s & 1-\frac{1}{m} & \frac{1}{m}(1-\Delta s) \\ \Delta s & 0 & 1-\Delta s \end{pmatrix}. \end{aligned} \tag{4.29}$$

Then

$$\begin{aligned} P(\mathbf{S}^* \to \mathbf{S}^-) = P([1\ 1\ 0] \to [0\ 1\ 0]) + P([1\ 1\ 0] \to [0\ 2\ 2]) + \\ P([1\ 1\ 0] \to [0\ 1\ 2]) + P([1\ 1\ 0] \to [0\ 2\ 0]) = \\ \frac{\Delta s}{2}\frac{1}{m}\Delta s\frac{1}{m}(1-\Delta s) + \frac{\Delta s}{2}\frac{1}{m}(1-\Delta s)(1-\Delta s) + \\ \frac{\Delta s}{2}\frac{1}{m}\Delta s\frac{1}{m}(1-\Delta s) + \frac{\Delta s}{2}\frac{1}{m}(1-\Delta s)\,\Delta s = \\ \frac{\Delta s}{2}\left(\Delta s\left(\frac{2(1-\Delta s)}{m}-1\right)+1\right). \end{aligned} \tag{4.30}$$

As we can see, the probability of generating a sequence $\mathbf{S}^*$ given an existing sequence $\mathbf{S}^-$ is not the same as the backward probability of generating $\mathbf{S}^-$ given $\mathbf{S}^*$, thus the proposal is not symmetric. Given the challenge that the calculation of the Hastings factor for this proposal posed, we decided to take a different approach.

4.3.2 Reversible rescaling method

The second method takes a different approach to the first one. As one of the reasons that discouraged us from pursuing the first approach was the complex Hastings factor calculation,

we defined a proposal mechanism that is symmetric that generates state sequences based on three reversible moves.

1) ***Random sequence***. Through this move, the residence times in each state are sampled from a gamma distribution with variable parameters $\Gamma\left(\alpha, \frac{\alpha}{m}\right)$.

2) ***Random segment stretch***. Here, a randomly chosen segment, defined as the length between two consecutive switches, is multiplied by a random number sampled from $N(0, \varepsilon)$. The length of the time series is kept fixed.

3) ***Shuffle***. Through this move, all segments are shuffled, meaning that each state is set to a different state randomly.

Note that the main feature of the second move is its reversibility, whereas the third move is necessary for ergodicity. This is a much more simplistic method than the previous one but its strength relies on the use of gamma-distributed residence times for each state.

To prove the reversibility of the second move, consider $z \sim N(0, \varepsilon)$. Then from a current segment length ℓ, a new segment length ℓ' is proposed via

$$\ell' = \ell \; exp(z), \tag{4.31}$$

whereas ℓ is proposed from ℓ' via:

$$\ell = \ell' \; exp(-z). \tag{4.32}$$

From symmetry of the normal distribution we have that $p(z) = p(-z)$, thus we conclude that the probabilities associated with the proposals in Eqns. 4.31 and 4.32 are:

$$p(\ell|\ell') = p(-z) = p(z) = p(\ell'|\ell), \tag{4.33}$$

therefore the Hastings factor for the second move is 1. Given that the first and third moves are both random, we can state that the this proposal generating mechanism is symmetric and therefore the Hastings factor needs not be taken into account for the acceptance rate.

4.3.3 Reversible mutation method

This latest method is an extension of the reversible rescaling method and presents three moves that alter state sequences in the following manner. Firstly, we generate an initial state sequence randomly by sampling the residence times for each state from a gamma distribution $\Gamma\left(\alpha, \frac{\alpha}{m}\right)$. This is reversible as the switch points are randomly sampled. Then the three moves

are the following.

1) ***Move.*** A transition point is selected at random and shifted either to the left or to the right.

2) ***Add.*** New transition points are added.

3) ***Remove.*** Transition points are removed.

The first move is analogous to the *random segment stretch* move from the reversible rescaling method. We restrict the new transition to occur at the times of observations. Having selected a switchpoint r at random with probability $\frac{1}{f}$, where f is the total number of switchpoints, and identifying the immediately before and after switchpoints as, respectively, a and b, the new transition point r' is selected with discrete uniform probability $\frac{1}{f} \cdot \frac{1}{n_{obs}^{a,b}}$, where $n_{obs}^{a,b}$ is the number of observations between a and b. Hence, for any interval $[a,b]$, for $a,\ b$ transition points,

$$p(r'|r) = \frac{1}{f} \cdot \frac{1}{n_{obs}^{a,b}} = p(r|r'), \tag{4.34}$$

that is, move 1 is symmetric and the Hastings factor is 1. We use a discrete uniform probability to account for irregularly spaced data, so that every time of observation has equal probability of being selected. If a continuous uniform distribution were to be used, denser observations would become more likely to be chosen.

Before exploring move 2 and move 3, we shall make the following definition. Let's consider two ordered transition points, say a and b, so that the state before a is the state after b. Then any two transition points that satisfy this condition and do not contain in between any other nested transition points satisfying the same condition are called a "pair".

Now, for the *remove* move, we begin by selecting a time point r continuously uniformly on the total time interval $[0,T]$, $r \sim U(0,T)$. Then we choose a pair Q such that:

$$\text{dist(Q,r)} \leq \text{dist(P,r)},\ \forall \text{ pairs P}, \tag{4.35}$$

where the distance function is:

$$\text{dist(P,r)} = (x_P - r)^2 + (y_P - r)^2, \tag{4.36}$$

for x_P and y_P, respectively, the startpoint and endpoint of P.

The *add* move is constructed so that it constitutes the *remove* reverse move. As for the previous move, we select a time point r from $r \sim U(0,T)$ and the startpoint and endpoint of the segment containing r are found. We sample the number of states to add, k, from the

interval $[1, n-1]$, for n total number of states. Let L be the segment length; we need to sample $k+1$ switchpoints and place them on the segment. Therefore, $k+1$ switchpoints are selected with probability

$$\prod_{i=0}^{k}(L-i)^{-1}. \tag{4.37}$$

Furthermore, the probability of selecting a specific state is given by

$$\prod_{i=0}^{k-1}((k-1)-i)^{-1}. \tag{4.38}$$

By including these probabilities in the Hastings factor, we are now satisfying detailed-balance and have built a valid sampler.

4.4 Simulation study

We will now give a demonstration of the performance of each method. We apply the methods on a simple mixture model where at every time point the position is sampled from a normal distribution $N(0,\sigma)$. We define two very distinct states corresponding respectively to higher and lower values of σ: $\boldsymbol{\sigma} = [1,5]$; the duration of each state is sampled from a gamma distribution with concentration $\alpha = 10$ and mean $m = 250$. We optimise the movement parameter $\boldsymbol{\sigma}$ as well as the gamma tuning parameters.

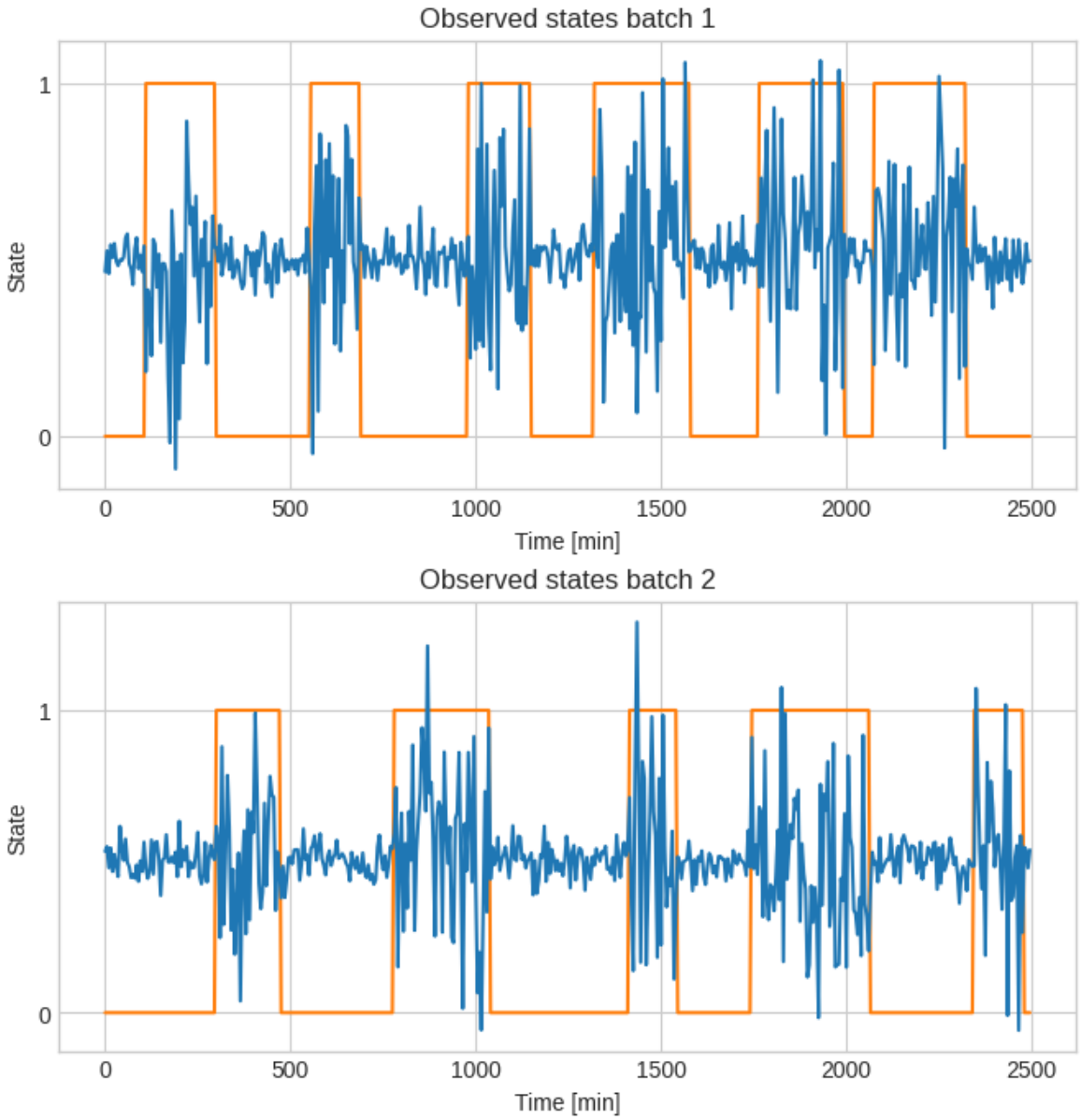


Figure 4.5: Trajectories from a simple mixture model. The states are the orange lines.

4.4.1 Virtual state method

In Fig. 4.6 we show the optimised model parameters and in Fig. 4.7 we report the reconstructed state sequence for the virtual state method. The sampler was run for 5000 steps and we only show the last 1000 samples in the plot. Note how the concentration parameter is optimised to 1 and the sampler was unable to reconstruct the true state sequence.

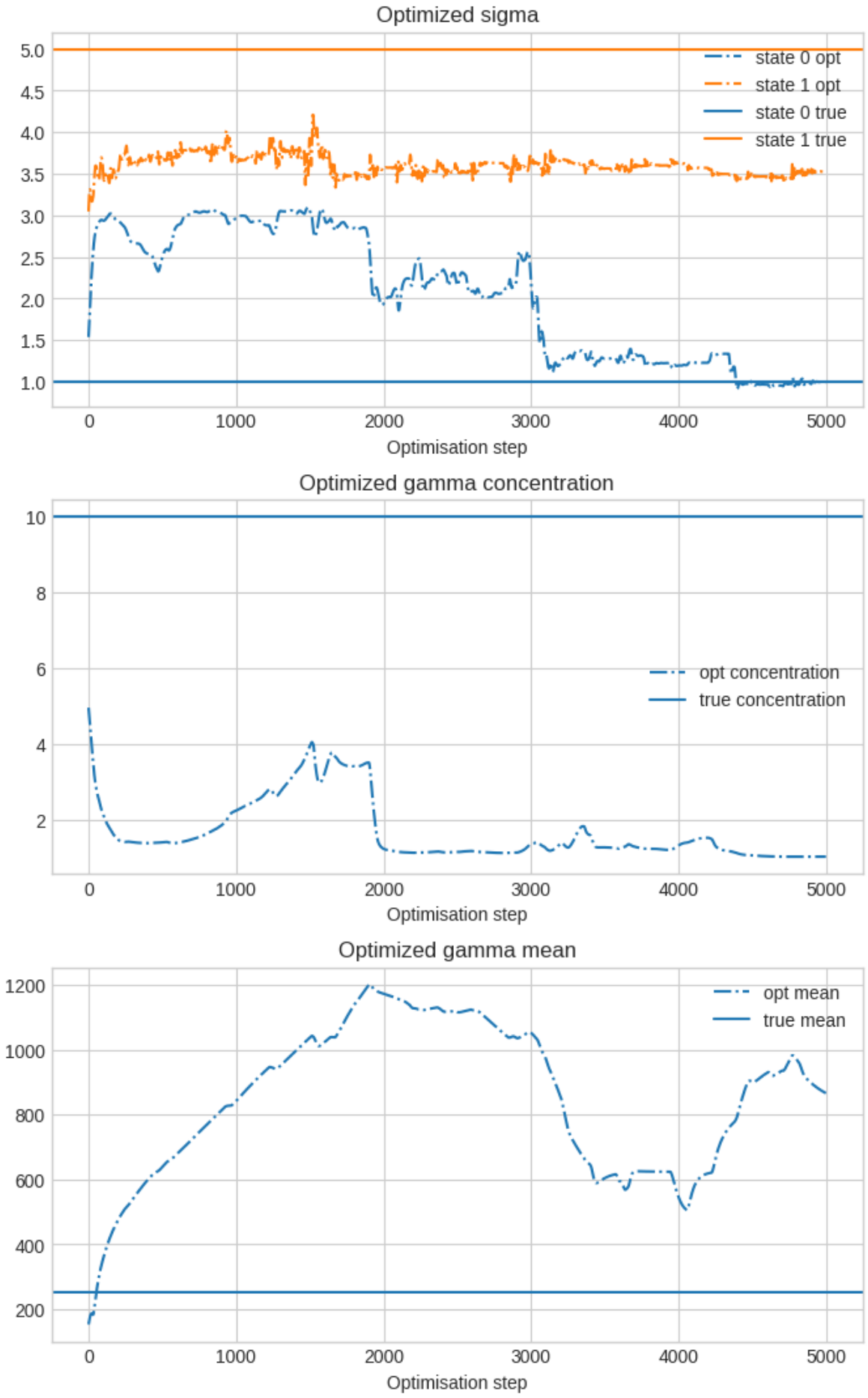


Figure 4.6: Optimised parameters from the virtual state proposal.

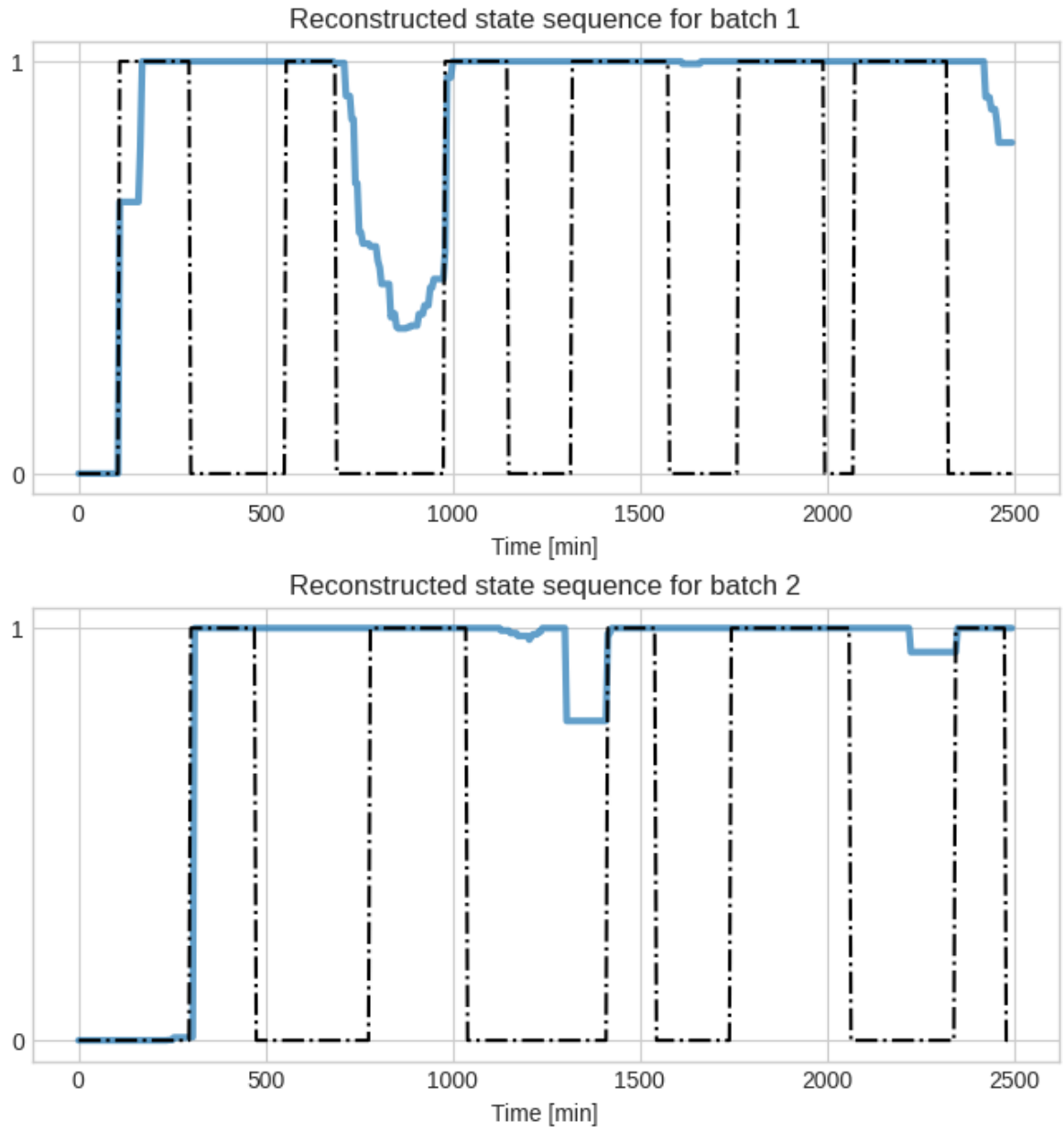


Figure 4.7: Reconstructed state sequences from the virtual state proposal.

4.4.2 Reversible rescaling method

In Figs. 4.8 and 4.9 we report the results from the reversible rescaling method. The chain was run for 5000 steps and only the last 1000 samples are shown in the reconstructed state sequence plot. The reconstructed state sequence was stuck in a sub-optimal configuration.

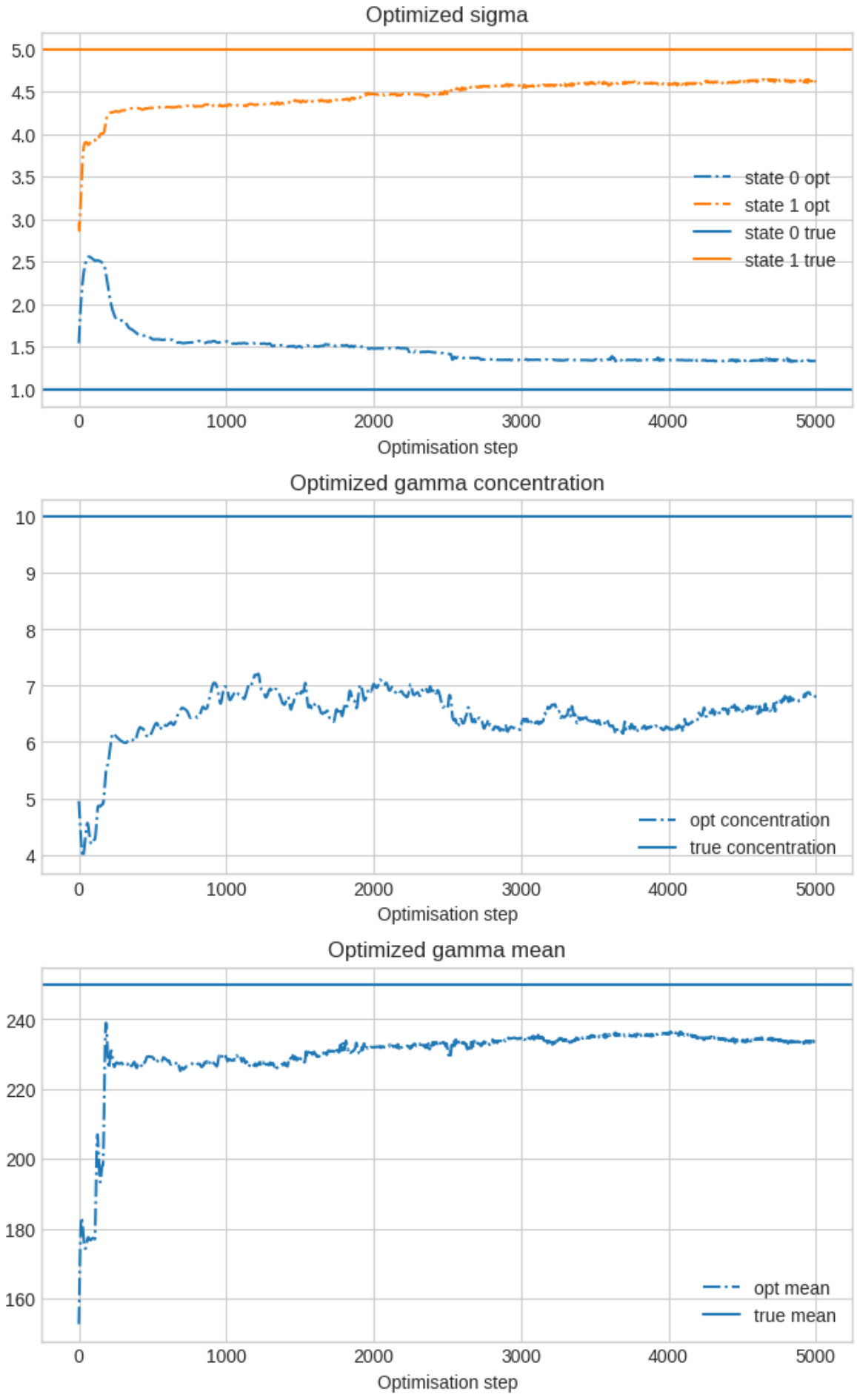


Figure 4.8: Optimised parameters from the reversible rescaling method proposal.

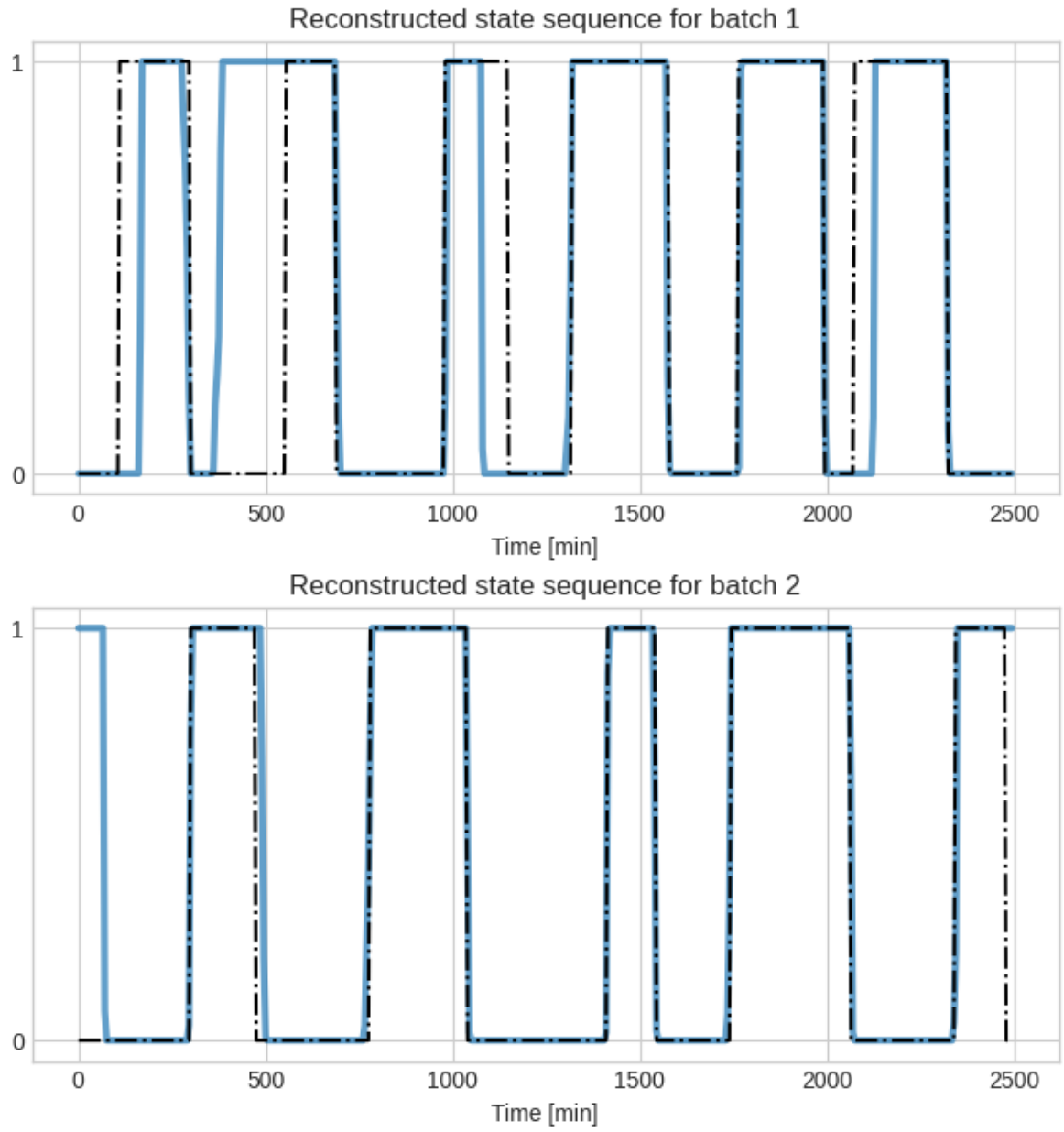


Figure 4.9: Reconstructed state sequences from the reversible rescaling method proposal.

4.4.3 Reversible mutation method

In Figs. 4.10 and 4.11 we show the results from the latest method. Then chain was run for 5000 steps and only the last 1000 samples are shown. As we can see from comparing the reconstructed state sequence to the true state sequence, the reversible mutation method was the most successful method.

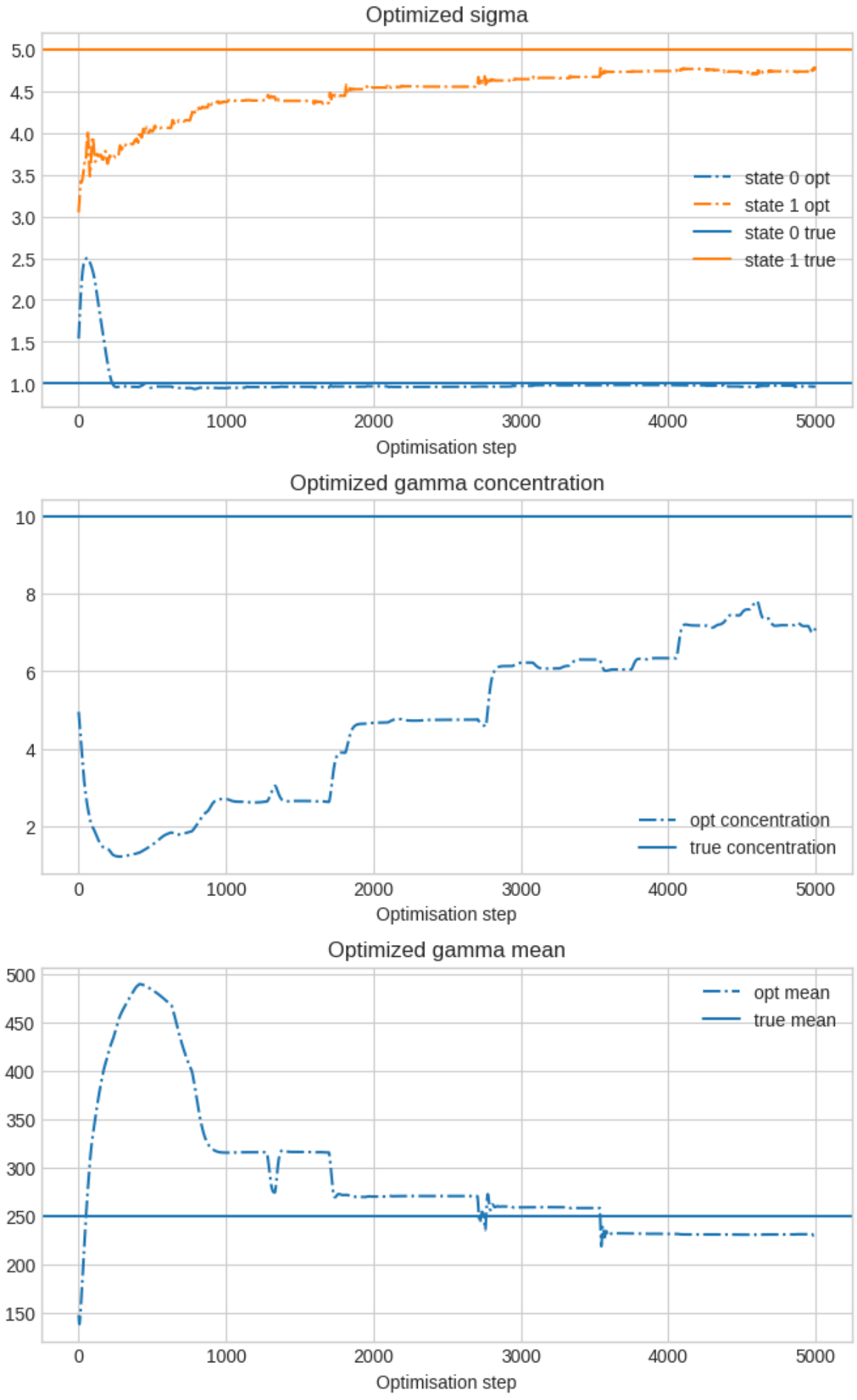


Figure 4.10: Optimised parameters from the reversible mutation proposal.

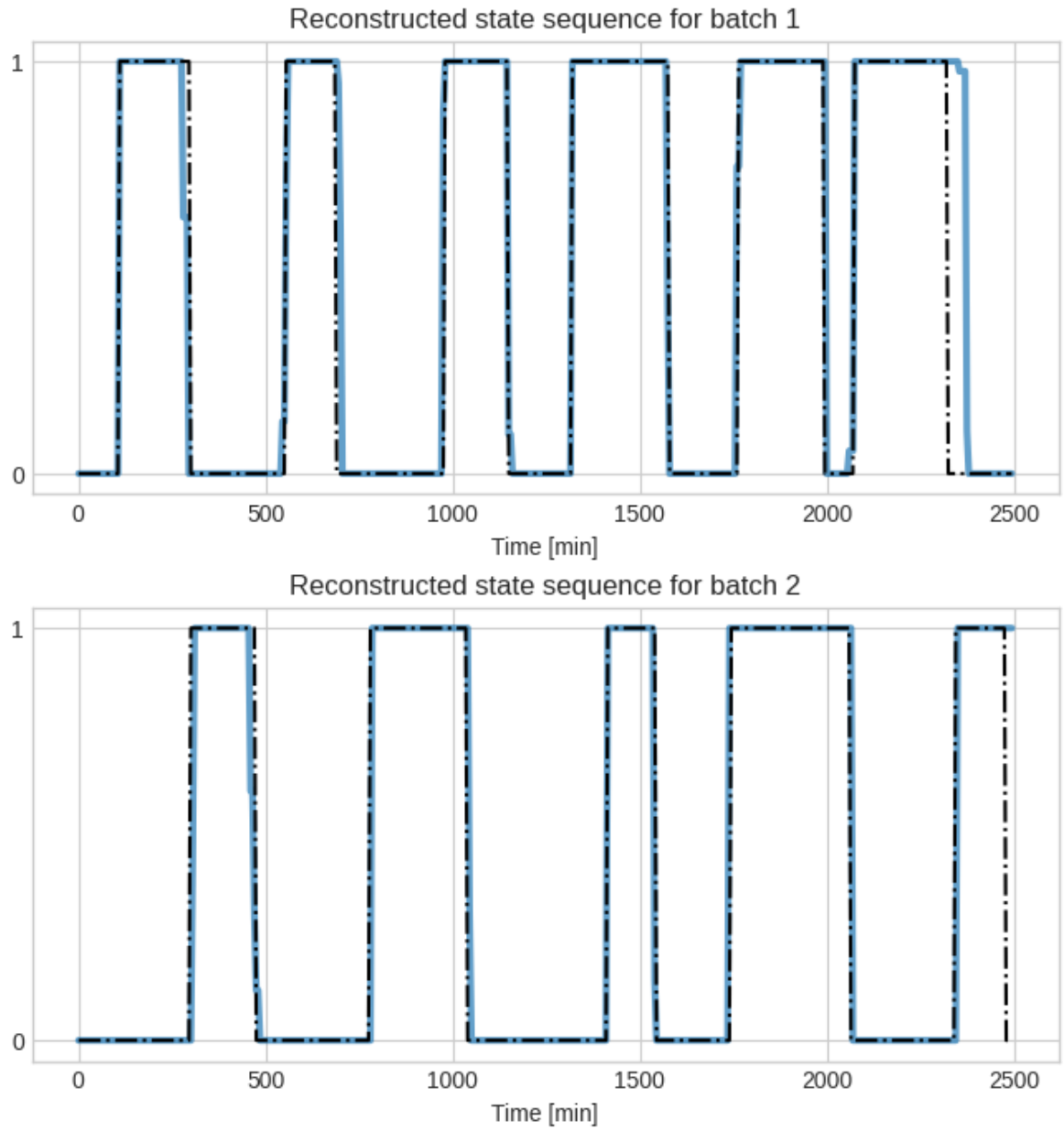


Figure 4.11: Reconstructed state sequences from the reversible mutation proposal.

4.5 Discussion

We have presented three different generating mechanisms to construct semi-Markov chains. All of the methods above come with advantages and disadvantages. Our objective was to exit the well established HMM framework and go beyond the use of exponentially distributed residence times. In a sense, our aim was to generalise the HMM framework by constructing a hidden semi-Markov model that employs a gamma distribution instead. From this perspective, it's understandably easy to identify the major drawback associated with the virtual state method.

Indeed, the main issue with the virtual state method was the reliance on packages whose built-in functions employ an exponential distribution, given that the residence times in each state in a continuous-time HMM follow this distribution. This is in direct contrast with our prior assumption that the duration of each state follows a gamma distribution. Although we augmented the state-space model with a virtual state that copied previous states to increase the duration of the sojourn times, we found that this discordance lead to problematic results. Indeed, the reconstruction of the most likely state sequence was never successful. In the application of this method to data representing the location of sheep, we found that new samples would feature short segments, a feature inherited from the exponential distribution. Although we included a gamma prior to promote rejection of short segments, the sampler couldn't quite recover the true state sequence. This was most likely due to the measurement error and weak signal from the dataset, and as a consequence the sampler tried to associate each observation to a different state. On the contrary, when the method was applied to synthetic data generated by a simple mixture model showing a strong signal, the increase of the sojourn times lead to rejection of short segments and acceptance of too large residence times, with a mean residence time fluctuating between 500 and 1200 minutes, as shown in Figs. 4.6 and 4.7.

Furthermore, this leads to a second issue concerning the optimisation of the gamma prior. Indeed, in the maximisation step of our MCEM framework, the concentration gamma prior parameter α was always optimised to 1, which is the limiting case of an exponential distribution. Hence, this encouraged acceptance of short segments, nullifying the use of our gamma prior. These issues lead us to the conceptualisation of the reversible rescaling method.

For the reversible rescaling method, the choice for such specific moves was driven by the desire to build a simple model that can efficiently reconstruct the most probable hidden state sequence and that satisfies the following two criteria: it does not rely on existing HMM packages leveraging exponential distributions; the proposal mechanism is mathematically simple, so as to avoid calculation of a possibly expensive Hastings factor. In that sense, the second method was successful. However, two issues were found. Firstly, the gamma parameters optimisation did not work effectively. Secondly, the *random segment stretch* move did not allow to stretch a segment length to 0. This created issues as the sampler couldn't explore all sequences configurations and it was in some cases found to be stuck in sequences that were not matching the true sequence. Because of this, we did not investigate as to why the optimisation did not work properly and we opted to create a new method.

The reversible mutation method was born as an effort to go beyond the limitations intrinsic to the reversible rescaling method, hence the add/remove moves. These two moves are defined in tandem but contrary to the birth and death moves described in the Chapter 2, they do not affect the total number of states. However, this came at the expense of sacrificing the mathematical simplicity of the second method and ensuring detailed-balance was found to be a challenging task given the probability of generating sequences for the add and remove

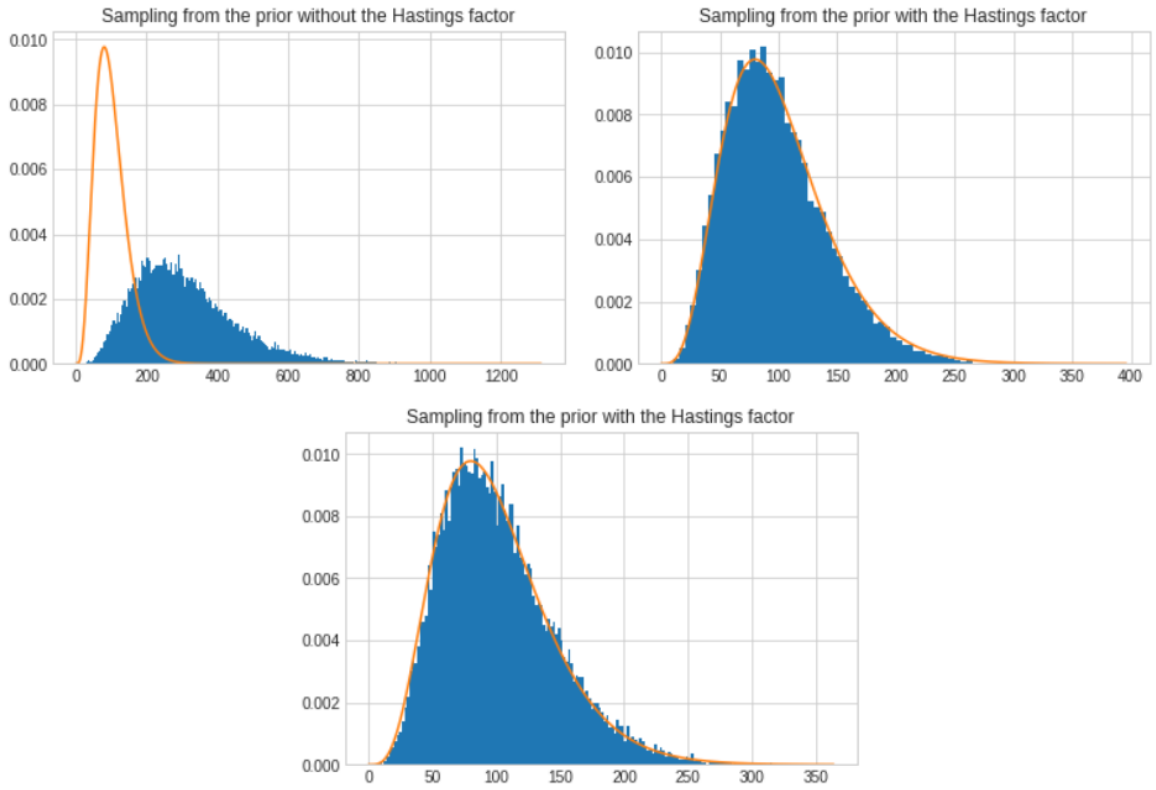


Figure 4.12: Importance of the Hastings factor. The orange line represents the prior function that we have sampled from. On the left - the sampler was run without the Hastings factor. On the right - the Hastings factor was included in the acceptance probability. Down below - the Hastings factor was included for a 10-state model.

move were not trivial. As we show in Fig. 4.12, calculation of the Hastings factor was essential for constructing a valid sampler. Here we simply sampled from the prior to verify that the Hastings factor was correct. For the first two plots we used 2 states and for the last plot 10 states were used. Once the Hastings factor was included, the sampler was found to outperform the previous methods (Figs. 4.10, 4.11).

Another drawback of the reversible mutation method is the add move is likely to create short state segments, although these sequences will be rejected by the prior. In Chapter 5, we used a modified reversible mutation proposal where at every add or remove step, only one behavioural segment was either added or removed. This was done to simplify the Hastings factor on the one hand, and to avoid generating multiple short segments on the other hand.

Chapter 5

Scalable non-Markovian state switching models for animal movement

The following material is currently under review for publication in the Methods in Ecology and Evolution journal.

5.1 Abstract

Observed animal movement trajectories are often the result of a latent process whereby an animal transitions between discrete behavioural states such as foraging or resting. The standard approach for analysing multi-state movement data is to employ hidden Markov models (HMMs) and these models have been used in a wide array of animal movement studies. Recent developments have enabled HMMs to be applied to irregularly sampled data, as well as providing uncertainty quantification in the inferred latent states. However, all such models rely on the unrealistic underlying assumption that sojourn times in each behavioural state are exponentially distributed, meaning there is always a constant probability of leaving a state.

Here, we propose a hidden semi-Markov model where movement is modelled as a continuous-time integrated Ornstein-Uhlenbeck process and behavioural state transitions are governed by an arbitrary distribution of sojourn times. We employ a Monte Carlo Expectation-Maximisation (MCEM) algorithm to reconstruct the hidden state sequences as well as to optimise the parameters of the movement and state switching dynamics. We apply our framework both to synthetic data and to telemetry data from free-roaming Merino sheep in Patagonia.

Our method efficiently optimises all parameters of the model, is scalable to large datasets, and provides a posterior distribution of latent state sequences. Due to our novel inference methodology we are able to employ a gamma distribution for sojourn times, leading to a more realistic model of animal behaviour since the probability of leaving a state depends on the amount of time spent in that state.

Our proposed method builds upon multistate state-space models from the literature but it is more flexible than standard hidden Markov models approaches in that it allows the user to choose the distribution of the residence time in each state on a case-by-case basis. By employing highly optimised machine learning libraries, this methodology is a suitable tool to efficiently deal with high-volume datasets and will facilitate the analysis of high-resolution telemetry data which have become more available to movement ecologists in recent years.

5.2 Introduction

The technological advances in the field of movement ecology in recent years have led to an increasing amount of high-resolution animal tracking data being collected (Nathan et al., 2022, Cagnacci et al., 2010). As a result, there has been an array of new statistical methodologies developed to tackle the challenges associated with the analysis of such large data sets. A common task in the study of animal movement is identifying changes in movement patterns that can be associated with different behavioural states. Here the canonical challenge is to segment the data into "states" based on quantitatively different statistical properties of segments of the full movement trajectory and to reconstruct the sequence of state switches.

The term state is used as a proxy for a distinct animal behaviour; while movement data provides high-frequency locations, different behaviours are expected to show as a consequence of changes in the animals' activities over longer time-scales, for example a change from an exploratory behaviour to a resting behaviour.

In the recent years, the task of identifying underlying behavioural states has been tackled by employing hidden Markov models (HMMs) (Zucchini et al., 2009), whereby the data are assumed to be generated by a movement process that is dependent on a latent (therefore hidden) behavioural process represented by a Markov chain. HMMs are formulated either in discrete time or in continuous time and both formulations have their advantages and disadvantages. Discrete-time HMMs are currently the preferred model due to their mathematical simplicity and easily interpreted parameters. The standard discrete-time formulation requires the transformation of trajectories into a sequence of steps and turns in polar coordinates (Morales et al., 2004) and modelling the dynamics as an ordered series of step lengths and turn angles drawn from defined probability distributions. Extensive effort has been put in the development of user-friendly packages for the researcher designed to tackle switching problems for telemetry data analysis (Michelot et al., 2016, McClintock and Michelot, 2018). However, discrete-time models are not ideal tools for the analysis of irregularly-spaced data or in the presence of non-negligible measurement error (Patterson et al., 2017b, Hooten et al., 2017b, Michelot and Blackwell, 2019). In particular, they require partitioning of the time domain into temporally-regular time steps that must be specified a priori and that are assumed to match the scale at which behavioural decisions are made (McClintock et al., 2014, Turchin, 1998).

At the expense of some mathematical simplicity, a natural extension of discrete-time HMMs are models where movement is defined using a continuous-time location process and the observation process is represented by adding measurement error to the true, noise-free locations. In a recent work by Michelot and Blackwell (2019), this approach was extended to a multi-state framework to address switching problems by introducing a hidden process representing the behavioural process, on which the location dynamics are dependent. The location process is described by a continuous-time correlated random walk (CTCRW) to account for autocorrelation of movement; specifically, the location is modeled through an integrated Ornstein-Uhlenbeck (OU) process (Michelot and Blackwell, 2019, Johnson et al., 2008), whereas the behavioural process is described by a continuous-time Markov chain. Continuous-time models don't require specification of a time scale a priori and can therefore handle irregularly-spaced data as well as different temporal scales (Patterson et al., 2017b). This can overcome issues arising from missing data or limitations within the telemetry devices.

While HMMs have seen an increase in popularity in the animal movement community and have been widely used in switching problems, the Markov property implies underlying

assumptions on the behavioural process that are non-realistic. In the absence of external variables that influence the switching behaviour, the residence (or sojourn, dwell) time in each behavioural state follows an exponential distribution (or geometric, in the discrete-time counterpart), for which short and frequent state changes are favoured. This implies that the behavioural process inherits the "memoryless" property of the exponential distribution, meaning the sojourn times are independent of the amount of time spent in a state. Thus, when using HMMs it is assumed that the amount of time an animal will remain in a behaviour of, say, resting does not depend on how long the animal has already rested for. Another drawback of using the exponential distribution lies in the application of HMMs in an unsupervised context, whereby the number of states is not defined a priori and is learnt during model fitting. If the distribution of the residence times is not exponential (or geometric), in other words if the underlying structure of the model is non-Markovian, as we would typically expect in most animal movement studies and has been shown empirically in the case of the distribution of foraging times of beaked whales (Langrock et al., 2013), using model selection to define the number of states in HMMs is unreliable and expected to favour models having more behavioural states than there are in reality (Pohle et al., 2017). This can be avoided by including a non-Markovian structure into the model by using hidden semi-Markov models which allow for the informed specification of a dwell-time distribution thus relaxing the assumption of exponentially distributed waiting times (Ruiz-Suarez et al., 2022). For a comprehensive examination of the pitfalls associated with HMMs, see Glennie et al. (2023).

It is also worth adding that current approaches are not scalable to large datasets due to the computational burden of standard schemes such as Markov chain Monte Carlo (MCMC) sampling for which evaluation of the full dataset is required at every step of the sampler. These methods therefore become prohibitively expensive to apply to long-term, high-frequency telemetry datasets.

Here we propose a novel method scalable to large datasets that can be employed in the context of switching problems and addresses the intrinsic assumptions of standard HMMs by introducing a memory in the process. In particular, similarly to Michelot and Blackwell (2019) we employ a multi-state, CTCRW model where the location process is described by an integrated OU process, however a continuous-time semi-Markov chain is employed to describe the behavioural process, thus allowing us to have control over the residence times. We employ a gamma distribution to model the residence times, nonetheless arbitrary distributions, or mixtures of distributions may be used within our framework. By employing the gamma distribution, we favour longer sojourn times and effectively introduce a memory in the switching process by ensuring that the more time spent in a state, the more likely to switch behaviour.

Our framework is intended to solve two inferential tasks; the first task aims at reconstructing the most probable hidden state sequence given observed trajectories, whereas the second

task addresses the recovering of both the behavioural and the location process parameters. This is done by employing a Monte Carlo Expectation-Maximisation (MCEM) algorithm that performs the two tasks sequentially, the Expectation step and the Maximisation step. In the E-step, a Metropolis-Hastings (MH) scheme is used to sample over the distribution of the latent state sequences to approximate the expectation of the model likelihood. The validity of the sampler is ensured by introducing a novel state sequence proposal mechanism based on three different reversible mutation steps that are designed to ensure detailed-balance is satisfied. In the M-step, the expectation is optimised with respect to the model parameters which are then updated and used in the subsequent iteration of the algorithm.

The remainder of the paper is structured as follows: in Section 5.3.1 we present the model formulation for both the movement and the switching processes; in Section 5.3.3 we explain the inferential algorithm and a pseudocode is given. In Section 5.4 we show the results of the method applied to both synthetic data and data of Merino sheep free roaming in Patagonia. We conclude with a discussion of our method.

5.3 Materials and Methods

5.3.1 State-switching movement model

In this paper, we employ a three-layer state-space model where the first layer is the underlying, hidden switching state/behaviour process, the second layer is the location process which depends on the first layer and the third and final layer is the observation process which accounts for measurement error. Here we focus on a continuous-time formulation of the switching process with a discrete state space. In a standard implementation of an HMM, the switching process is assumed to be described by a Markov chain, for which state changes are dependent only on the state at the previous time point. For an n-state model, the continuous-time Markov chain governing the state switches is described by an infinitesimal generator matrix (IGM)

$$\mathbf{\Lambda} = \begin{pmatrix} -\lambda_{11} & \dots & \lambda_{1n} \\ \dots & \dots & \dots \\ \lambda_{n1} & \dots & -\lambda_{nn} \end{pmatrix}, \tag{5.1}$$

where $\forall i, j \in \{1, \dots, n\},\ \lambda_{ii} = \sum_{i \neq j} \lambda_{ij}$. The parameters λ_{ij} represent the rate at which the state i transitions to state j. As a consequence of the Markov property, the sojourn time of a state is exponentially distributed with rate parameter λ_{ii}. Our objective is to relax this assumption and present a more flexible behavioural process that can take into account biological factors that reflect on the duration of specific animals' activities. Therefore, we make use of a continuous-time semi-Markov chain and we choose to model the sojourn time with

a gamma distribution parameterised as follows

$$\text{Gamma}\left(\alpha, \frac{\alpha}{m}\right), \tag{5.2}$$

where α is the shape parameter and m is the mean parameter, representing the mean residence time in a state.

The location process is modeled via an integrated OU process (Michelot and Blackwell, 2019, Johnson et al., 2008), meaning that the velocity is modeled with an OU process and the position is derived by integration of the velocity model. By equipping the location process with an explicit state dependency, the location $\boldsymbol{\mu}(t)$ and the velocity $\mathbf{v}(t)$ are updated according to the following 2-dimensional stochastic differential equations (SDEs) (Michelot and Blackwell, 2019),

$$\begin{aligned} d\boldsymbol{\mu}(t) &= \boldsymbol{v}(t)dt, \\ d\boldsymbol{v}(t) &= \beta_i(\boldsymbol{\gamma}_i - \boldsymbol{v}(t))dt + \sigma_i d\boldsymbol{w}(t), \end{aligned} \tag{5.3}$$

where the parameter β_i measures the persistence in the speed and direction of the movement, σ_i measures the variability in velocity, $\boldsymbol{w}(t)$ is a Wiener process and $\boldsymbol{\gamma}_i$ is the mean velocity parameter which we take to be zero in all that follows. The subscript i represents the dependency on state i, meaning different movement characteristics are associated with different state values. For example, lower values of β result into longer-term persistence in the direction and speed of the movement. Thus, we introduce the vector of model parameters

$$\boldsymbol{\theta} = [\boldsymbol{\beta}, \boldsymbol{\sigma}, \alpha, m, \omega], \tag{5.4}$$

where

$$\begin{aligned} \boldsymbol{\beta} &= [\beta_1, ..., \beta_n], \\ \boldsymbol{\sigma} &= [\sigma_1, ..., \sigma_n]. \end{aligned} \tag{5.5}$$

The last layer is represented by the observation process, i.e. the observed locations; this is obtained straightforwardly by augmenting the location process with noise:

$$\begin{aligned} \mathbf{y}(t) &= \boldsymbol{\mu}(t) + \boldsymbol{\chi}(t) \\ \chi_c &\sim N(0, \omega^2), \end{aligned} \tag{5.6}$$

where $c = x, y$ represents coordinate axes.

5.3.2 Conditional log-likelihood

Let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_d\}$ be a trajectory, that is, a set of observed locations, where d is the dimension of the set. We refer to $\mathbf{Y}$ as the incomplete data set. If knowledge of the hidden states $\mathbf{Z} = \{z_1, \ldots, z_d\}$ was available, with $z_t \in \{1, \ldots, n\}$, then we could access the complete data set $\{\mathbf{Y}, \mathbf{Z}\}$. The hidden states provide a segmentation of the locations into M contiguous blocks of behaviours. Let $\{\mathbf{Y}^{i,1}, \ldots, \mathbf{Y}^{j,M}\}$ be the partitioned trajectory, where superscript i indicates state dependency. Assuming independency of the segments, it follows that the conditional likelihood of the trajectory given the hidden states is

$$p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}) = \prod_{k=1}^{M} p(\mathbf{Y}^{i,k}|\sigma_i, \beta_i, \omega). \tag{5.7}$$

Since locations are modelled via an integrated OU process, for each segment the likelihood conditional on the behavioural state i

$$p(\mathbf{Y}^{i,k}|\beta_i, \sigma_i, \omega) \tag{5.8}$$

follows a multivariate normal distribution $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}_0$ is the initial locations and the covariance matrix is given by (Gardiner, 2009, Paun et al., 2022)

$$\Sigma_{st} = \frac{\sigma_i^2}{2\beta_i^3}\left(2\beta_i min(s,t) - 1 - e^{-\beta_i|t-s|} + e^{-\beta_i s} + e^{-\beta_i t}\right) + \omega\delta_{st}, \tag{5.9}$$

where s and t are time points of the observations and δ_{st} is the Kronecker delta. In application, we use the conditional log-likelihood

$$\ell_{cond} = ln\ p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}). \tag{5.10}$$

Unlike the incomplete data log-likelihood, the conditional likelihood is tractable. This will be exploited in our MCEM algorithm to approximately maximise the incomplete log-likelihood, as explained in the next section.

5.3.3 Monte Carlo EM algorithm

We are faced with the task of reconstructing the history of the hidden behavioural states as well as recovering the state-dependent movement parameters of the model together with the measurement error parameters from the telemetry data. For this task, we employ a Monte Carlo Expectation-Maximisation algorithm (Wei and Tanner, 1990) that enables us to combine a sampler with an efficient stochastic gradient descent (Robbins and Monro, 1951) algorithm.

The Expectation-Maximisation (EM) algorithm is employed for finding maximum likeli-

hood solutions for probabilistic models having latent variables (Dempster et al., 1977, Bishop, 2006). The algorithm is a two-step iterative algorithm; in the first step, the E-step, the expectation of the complete data log-likelihood is taken with respect to the posterior distribution of the hidden state sequences, given the current model parameters.

In the second step, the M-step, the expectation is optimised with respect to the model parameters which are then updated and used during the next E-step. This process is repeated until convergence. In our settings, the goal is to find maximum likelihood estimates for the parameter set $\hat{\boldsymbol{\theta}}$ as well as the posterior distribution of the hidden state sequence given the trajectories and $\hat{\boldsymbol{\theta}}$, $p(\mathbf{Z}|\mathbf{Y},\hat{\boldsymbol{\theta}})$. To obtain $\hat{\boldsymbol{\theta}}$, we need to maximise the incomplete data log-likelihood

$$ln\ p(\mathbf{Y}|\boldsymbol{\theta}) = ln\left(\sum_{\mathbf{Z}} p(\mathbf{Y}|\mathbf{Z},\boldsymbol{\theta})p(\mathbf{Z}|\boldsymbol{\theta})\right) \tag{5.11}$$

which is generally intractable due to the fact that the number of possible state sequences increases exponentially with the trajectory length. To estimate $ln\ p(\mathbf{Y}|\boldsymbol{\theta})$, we proceed as follows.

Given any proper probability distribution over the latent variables $q(\mathbf{Z})$, the following decomposition holds (Bishop, 2006)

$$ln\ p(\mathbf{Y}|\boldsymbol{\theta}) = \mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) + KL(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})), \tag{5.12}$$

where the quantity

$$\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) ln\left(\frac{p(\mathbf{Y},\mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}\right) \tag{5.13}$$

acts as a lower bound for the incomplete data log-likelihood and

$$KL(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) ln\left(\frac{p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})}{q(\mathbf{Z})}\right) \tag{5.14}$$

is the Kullback-Leibler divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})$, where $KL \geq 0$ and $KL = 0$ if and only if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})$. Considering a general E-step and supposing the current parameter value is $\boldsymbol{\theta}^-$, the arbitrary $q(\mathbf{Z})$ distribution is set to the posterior distribution of the latent variables, giving (Bishop, 2006)

$$\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-) ln\ (p(\mathbf{Y}|\mathbf{Z},\boldsymbol{\theta})p(\mathbf{Z}|\boldsymbol{\theta})) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-) ln\ p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-) \tag{5.15}$$

that is, the expectation of the complete data log-likelihood with respect to the posterior distribution of the latent variables given the current parameter estimate $\boldsymbol{\theta}^-$. In the M-step, this quantity is maximised with respect to $\boldsymbol{\theta}^-$ to give a new parameter value $\boldsymbol{\theta}^+$ which is used in the next E-M cycle. Since the M-step maximises the lower bound (Equation 5.13) and the E-step pushes the lower bound to the incomplete data log-likelihood because of Equation

5.14, every single cycle is guaranteed to increase the log-likelihood in Equation 5.12 and it can be shown (Dempster et al., 1977) that this converges to a zero-gradient point in the log-likelihood.

In our framework, we replace Equation 5.15 with a numerical approximation using a Monte Carlo estimate

$$\mathscr{L}(q(\mathbf{Z}),\boldsymbol{\theta}) \approx \frac{1}{S}\sum_{i=1}^{S} ln\ (p(\mathbf{Y}|\mathbf{Z}_i,\boldsymbol{\theta})p(\mathbf{Z}_i|\boldsymbol{\theta})) + const \tag{5.16}$$

where the second term in Equation (5.15) is constant with respect to $\boldsymbol{\theta}$, $\mathbf{Z}_i \sim p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta}^-)$ and S is the number of Monte Carlo samples. As this approximation uses samples from the posterior distribution of $\mathbf{Z}$ we are able to direct the Monte Carlo sum towards regions of higher likelihood; this is akin to importance sampling (refer to Bishop (2006), Section 23.4) and results in substantially reduced computational cost. This wouldn't be feasible by using Equation 5.11 directly, via the Monte Carlo approximation

$$ln\ p(\mathbf{Y}|\boldsymbol{\theta}) \approx ln\left(\frac{1}{S}\sum_{i=1}^{S} p(\mathbf{Y}|\mathbf{Z}_i,\boldsymbol{\theta})\right) \tag{5.17}$$

with $\mathbf{Z}_i \sim p(\mathbf{Z}|\boldsymbol{\theta})$, as a finite sample from the prior distribution is very unlikely to include state sequences $\mathbf{Z}_i$ for which the argument of the Monte Carlo sum has significant weight, leading to slow convergence and high estimation variance.

To efficiently optimise the parameter set $\boldsymbol{\theta}$, we gather the observations into equally-sized batches with each batch containing a pre-specified number of trajectories, where this number could represent the number of trajectories of different animals, the number of trajectories of the same animal over days or a mixture of both: $\mathscr{B} = \{\mathbf{Y}_1,\ldots,\mathbf{Y}_b\}$, hence we assume movement parameters are shared across individuals. Then during the E-step we implement a Metropolis-Hastings step to sample in parallel from the posterior distributions $p(\mathbf{Z}|\mathbf{Y}_j,\boldsymbol{\theta})$, for $j = 1,\ldots,b$; specifically, only one state sequence ($S = 1$) is sampled per trajectory, leading to b samples per E-step. Finally, in the M-step model parameters are optimised based on this set of samples. To further reduce the computational cost of the algorithm, we don't run the optimisation routine until convergence but rather take a single optimisation step based on the current set of samples. In the next Sections, the Metropolis-Hastings step and the M-step will be described in detail. A pseudocode of the full MCEM algorithm is given in Algorithm 1.

Metropolis-Hastings sampler

The Metropolis-Hastings algorithm is a Markov chain Monte Carlo algorithm used to obtain samples for high dimensional target distributions which would otherwise be difficult to sample from (Hastings, 1970). A new sample x' is generated via a proposal distribution $Q(x'|x)$

and the new sample is accepted according to an acceptance probability. This probability is specified so that the Markov chain satisfies the detailed-balance condition (Hastings, 1970) and the stationary distribution of the Markov chain is the target posterior distribution.

In our application, the Metropolis-Hastings sampler is employed to sample from the posterior distribution $p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})$, needed to estimate Equation 5.16. Let $\mathbf{Z}^-$ be the current state sequence sample and ℓ^-_{cond} its conditional log-likelihood. At every step of the sampler a new state sequence sample $\mathbf{Z}^+$ is proposed via a generating proposal mechanism $Q(\cdot,\cdot)$ (a description of the proposal mechanism is found in Section 5.3.3) and it is either accepted or rejected via the following acceptance ratio

$$a = min\left\{1, \frac{exp(\ell^+_{cond})p(\mathbf{Z}^+|\boldsymbol{\theta}^-)}{exp(\ell^-_{cond})p(\mathbf{Z}^-|\boldsymbol{\theta}^-)}\frac{Q(\mathbf{Z}^-|\mathbf{Z}^+)}{Q(\mathbf{Z}^+|\mathbf{Z}^-)}\right\}, \tag{5.18}$$

where $Q(x'|x)$ is the probability of generating x' from x through the proposal mechanism. $p(\mathbf{Z}^+|\boldsymbol{\theta}^-)$ and $p(\mathbf{Z}^-|\boldsymbol{\theta}^-)$ the probabilities associated with the state sequences $\mathbf{Z}^+$ and $\mathbf{Z}^-$. These probabilities are calculated by extracting the residence times (namely the segment length of a partitioned trajectory which is a sufficient statistic) and employing the gamma distribution in Equation 5.2 as the prior, in accordance with our assumptions on the distribution of the residence times of the hidden semi-Markov process.

Proposal generating algorithm

A key component of the E-step is drawing samples from the posterior distribution $p(\mathbf{Z}|\mathbf{Y},\boldsymbol{\theta})$ which are then accepted or rejected by the MH sampler. Therefore, a proposal mechanism to generate hidden state sequences samples is needed and a degree of similarity between successive samples must be introduced to ensure the efficacy of the sampler. In our algorithm the first sample is randomly generated by sampling the sojourn times of each state from a gamma distribution Gamma$\left(\alpha, \frac{\alpha}{m}\right)$ and assuming that for $n > 2$ each state has equal probability $\frac{1}{n-1}$ of being transitioned to. To generate a new proposal based on an existing state sequence, we construct a proposal generating algorithm based on the following three moves that allow for a reversible mutation of state sequences:

1) ***Shift.*** A transition (switching) point is selected at random and shifted to a new location.

2) ***Add.*** One behavioural segment is added.

3) ***Remove.*** One behavioural segment is removed.

While the underlying behavioural transitions occur in continuous-time, data observations are made at discrete intervals and we assume that the sampling frequency is sufficiently high such that multiple behavioural transitions are very unlikely to occur in the time between observations. We also assume that residence times can be approximated by considering the scenario where transitions occur at the time of the first observation of a behavioural segment.

For the *shift* move, let f be the total number of transition points. Then after selecting a

transition point s at random with probability $\frac{1}{f}$, the transition points immediately before and after are identified. Let a and b be the two such switchpoints; the new transition point s' is selected with discrete uniform probability

$$\frac{1}{f} \cdot \frac{1}{n_{obs}^{a,b}}, \tag{5.19}$$

where $n_{obs}^{a,b}$ is the number of observations between a and b. Hence, for any interval $[a,b]$, for transition points $a,\ b$,

$$p(s'|s) = \frac{1}{f} \cdot \frac{1}{n_{obs}^{a,b}} = p(s|s'), \tag{5.20}$$

that is, the *shift* move is symmetric. We use a discrete uniform probability to account for irregularly spaced data, so that every time of observation has equal probability of being selected, whereas a continuous uniform distribution would favour selection of denser observations.

For the *add* move, a time point r is selected uniformly $r \sim U(0,T)$, where T is the last time point, and the startpoint and endpoint of the behavioural segment (the segment between two consecutive transition points) containing r are identified; as before, let those points be, respectively, a and b. The state to add is selected from the $n-1$ available states and we sample two switchpoints that define the startpoint and endpoint of the new behavioural segment. To avoid the situation where an *add* move becomes equivalent to a *shift* move, we explore the two following possible scenarios (see SI for further details). In the first scenario, with probability $\frac{n-2}{n-1}$, the new state is different from the state after b, hence we allow b to define the endpoint of the new behavioural segment. In this case, two switchpoints are selected from a segment of length $|a-b| = L$. In the second scenario, with probability $\frac{1}{n-1}$ the selected state is equal to the state after the switchpoint b. Then the new segment endpoint is not allowed to be b and the two switchpoints are selected from a segment of length $L-1$. Thence, the probability of adding a behavioural segment is given by

$$\frac{L}{|\mathbf{Z}|} \cdot 2\left(\frac{1}{n-1}\frac{1}{L-1}\frac{1}{L-2} + \frac{n-2}{n-1}\frac{1}{L}\frac{1}{L-1}\right), \tag{5.21}$$

where $|\mathbf{Z}|$ is the sample length. While our proposal scheme may produce very short segments, these will effectively be filtered out by our gamma prior on the sojourn times (Equation 5.2).

For the *remove* move, if k is the total number of behavioural segments then the segment to be removed is selected with probability $\frac{1}{k}$. Then the behaviour associated with the removed segment is set with equal probability either to the behaviour in the next segment or to the behaviour in the previous segment.

We conclude with a discussion on the validity of our sampler. From Section 5.3.3, we examine the Hastings ratio

$$\frac{Q(\mathbf{Z}^-|\mathbf{Z}^+)}{Q(\mathbf{Z}^+|\mathbf{Z}^-)}. \tag{5.22}$$

In case of a symmetric proposal mechanism, this ratio is 1 and therefore the acceptance ratio in Equation 5.18 simplifies to the ratio of the products of the likelihoods with the priors. Although the shift move is symmetric, the reversible mutation proposal is overall asymmetric due to asymmetry arising from the add and remove moves. However, these moves are constructed in tandem and therefore are reversible. This means that if the forward probability of generating a new sequence $\mathbf{Z}^+$ from an existing sequence $\mathbf{Z}^-$ through an add move is given by $Q_A(\mathbf{Z}^+|\mathbf{Z}^-)$, then the reverse probability of generating $\mathbf{Z}^-$ from $\mathbf{Z}^+$ is given by $Q_R(\mathbf{Z}^-|\mathbf{Z}^+)$, where $Q_A(\cdot|\cdot)$ and $Q_R(\cdot|\cdot)$ are the probabilities of proposing samples through, respectively, an add move and a remove move. Thus, by incorporating these probabilities in the acceptance ratio, the detailed-balance condition is satisfied.

Optimiser

In the E-step, the expectation of the complete data log-likelihood in Equation 5.15 is approximated with a Monte Carlo estimate for b different trajectories

$$\zeta = \sum_{j=1}^{b} ln\ (p(\mathbf{Y}_j|\mathbf{Z}_i, \boldsymbol{\theta})p(\mathbf{Z}_i|\boldsymbol{\theta})), \tag{5.23}$$

where $\mathbf{Z}_i \sim p(\mathbf{Z}|\mathbf{Y}_j, \boldsymbol{\theta}^-)$. In the M-step, maximisation of ζ with respect to $\boldsymbol{\theta}^-$ is carried out.

The covariance structure in Equation 5.9, however, presents identifiability issues arising from weak identifiability of σ and strong identifiability of the ratio $\frac{\sigma}{\beta}$ for larger time values. To avoid mixture of strongly and weakly identifiable parameters, we reparameterise Equation 5.9 as follows

$$\Sigma_{st} = \tilde{\sigma}_i^2 \frac{\tau_i}{2}\left(\frac{2}{\tau_i} min(s,t) - 1 - e^{-\frac{|t-s|}{\tau_i}} + e^{-\frac{s}{\tau_i}} + e^{-\frac{t}{\tau_i}}\right), \tag{5.24}$$

where the new parameters are

$$\tilde{\sigma} = \frac{\sigma}{\beta}, \quad \tau = \frac{1}{\beta}. \tag{5.25}$$

The parameter τ represents the autocorrelation time of the velocity process (Gurarie et al., 2017), whereas $\tilde{\sigma}^2$ is the long-term ($t >> \tau$) slope of the increase in the expected squared displacement over time.

We equip the optimiser with a convergence criterion to avoid expenditure of computational time with no return in improved model likelihood. We specify a minimum improvement threshold ε and a tolerance parameter that controls the number of log-likelihood values below the threshold that can be accepted. Once the optimisation is stopped, we continue sampling from the latent state sequence posterior distribution with fixed model parameters, thus treating the optimisation routine as burn-in phase for the sampler. We note that given that we store the optimisation results for the model parameters at every step, we could also estimate their uncertainty by calculating the variance of the optimisation outputs.

Algorithm 1 Pseudo-code for MCEM algorithm

$w = 0$
$opt = True$
Set tolerance and minimum threshold ε
Initialise model parameter $\boldsymbol{\theta}^0$
Sample initial state sequence $\mathbf{Z}^0$ from Eq. 5.2
Calculate initial prior and conditional log-likelihood $p(\mathbf{Z}^0|\boldsymbol{\theta}^0)$, ℓ^0_{cond}
while $k < steps$ **do**

▷ E-step

Propose one sample per trajectory, $\mathbf{Z}^k$
Accept $\mathbf{Z}^k$ with probability a
Update every accepted batch
if $opt = True$ **then**
if $\zeta^k < \varepsilon * \zeta^{k-1}$ **then**
$w+ = 1$
else
$w = 0$
end if

▷ M-step

if $w < tolerance$ **then**
Optimise ζ^k with respect to $\boldsymbol{\theta}^{k-1}$
Update parameters
else
$opt = False$
end if
end if
if $opt = False$ **then**
$k+ = 1$
end if
end while

5.3.4 Synthetic data generation

In order to evaluate our framework, we generate synthetic data that matches the movement model used for inference. We simulate multiple individuals following an integrated OU process random walk with behavioural switching occurring at random intervals and the sojourn times in each behaviour following a gamma distribution. State transitions are equally likely for all states. For simulations of dynamics we do not numerically integrate Equation 5.3 but use the exact solutions described in Michelot and Blackwell (2019), Johnson et al. (2008):

$$\begin{aligned} v_c(t+\delta) &= e^{-\beta\delta} v_c(t) + \zeta_c(\delta) \\ \zeta_c(\delta) &\sim N\big(0, \sigma^2(1-e^{-2\beta\delta})/2\beta\big) \end{aligned} \tag{5.26}$$

and

$$\begin{aligned} \mu_c(t+\delta) &= \mu_c(t) + v_c(t)\left(\frac{1-e^{-\beta\delta}}{\beta}\right) + \xi_c(\delta) \\ \xi_c(\delta) &\sim N\left(0, \frac{\sigma^2}{\beta^2}\left(\delta - \frac{2}{\beta}(1-e^{-\beta\delta}) + \frac{1}{2\beta}(1-e^{-2\beta\delta})\right)\right), \end{aligned} \tag{5.27}$$

where the subscript $c = x, y$ represents Easting and Northing and δ is the time interval. Observations were fixed at every five minutes and we used additive, independent and identically distributed zero-mean isotropic Gaussian error for the measurement error.

5.3.5 Empirical data collection

As an example case study for our method, we analyse data collected from a long-term study of 58 Merino sheep allowed to roam freely in a paddock of 700 hectares. Fieldwork was conducted at the "Campo Anexo Pilcaniyeu" from INTA (National Institute of Agricultural Technology) Bariloche, Patagonia, Argentina. The sheep were equipped with collars containing a GPS (CatLog-B, Perthold Engineering **www.perthold.de**), that was programmed to record location every five minutes from February 2019 to December 2019, resulting in approximately 3 million measurements.

5.4 Results

5.4.1 Synthetic data study

In the simulation study, we simulated 8 independent trajectories with 3 latent behavioural states consisting of 512 observations each with a fixed 5-minute time interval to simulate a high frequency GPS collar sampling rate. The model parameter vector therefore consisted of

9 parameters:

$$\boldsymbol{\theta} = [\tau_1, \tau_2, \tau_3, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \alpha, m, \omega] \tag{5.28}$$

and the true values are displayed in Table 5.1.

Symbol	Description	True value
$\boldsymbol{\tau}$	Autocorrelation time	$(60, 10, 10)$
$\tilde{\boldsymbol{\sigma}}$	Long-term RMS displacement	$(10, 299.4, 50)$
α	Gamma shape	10
m	Mean residence time	500
ω	Measurement error std	0.1

Table 5.1: Table of true model parameter values.

For the early stopping criterion, the minimum improvement threshold ε was set to 0.01% and the tolerance was set to accept 2000 log-likelihood values below ε, while the sampler was run for 5000 steps after the optimiser had been stopped. We saved every $10th$ sample and used these to calculate the probability of being in each state for each individual and this is reported in Figure 5.1. In Figure 5.2 we plotted the optimised model parameters.

The sampler was able to reconstruct the underlying state sequences for all trajectories. Only in the bottom left panel a discrepancy between the true state sequence and the reconstructed state sequence is shown. As for the optimiser, all parameters were optimised to the true values except for the gamma shape parameter. This is probably due to the small effective data size in terms of number of switches (70 switches), hence with a sufficiently large data set, we expect the maximum likelihood estimations to be asymptotically unbiased (Cramer, 1946).

5.4.2 Merino sheep case study

For the analysis of the telemetry data we selected three sheep and considered their trajectories from May 1st to May 8th 2019. The trajectories were split into two; the first half contains data from midnight of May 1st to around 4pm of May 4th, and the second half contains the remaining locations until approximately 7am of May 8th. This yielded 6 independent trajectories storing 1024 data points each, equalling a total of 6144 observations.

We considered a model with 2 behavioural states for a total of 7 parameters to optimise,

$$\boldsymbol{\theta} = [\tau_1, \tau_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \alpha, m, \omega]; \tag{5.29}$$

we placed a peaked gamma prior on the measurement error to reflect our knowledge on the GPS factory error of approximately 50 metres. Initial values are shown in Table 5.2.

The first state sequence was constructed by sampling the residence times from $Gamma(\alpha^0, m^0)$, then the MCEM algorithm was run by specifying a minimum improvement threshold $\varepsilon =$

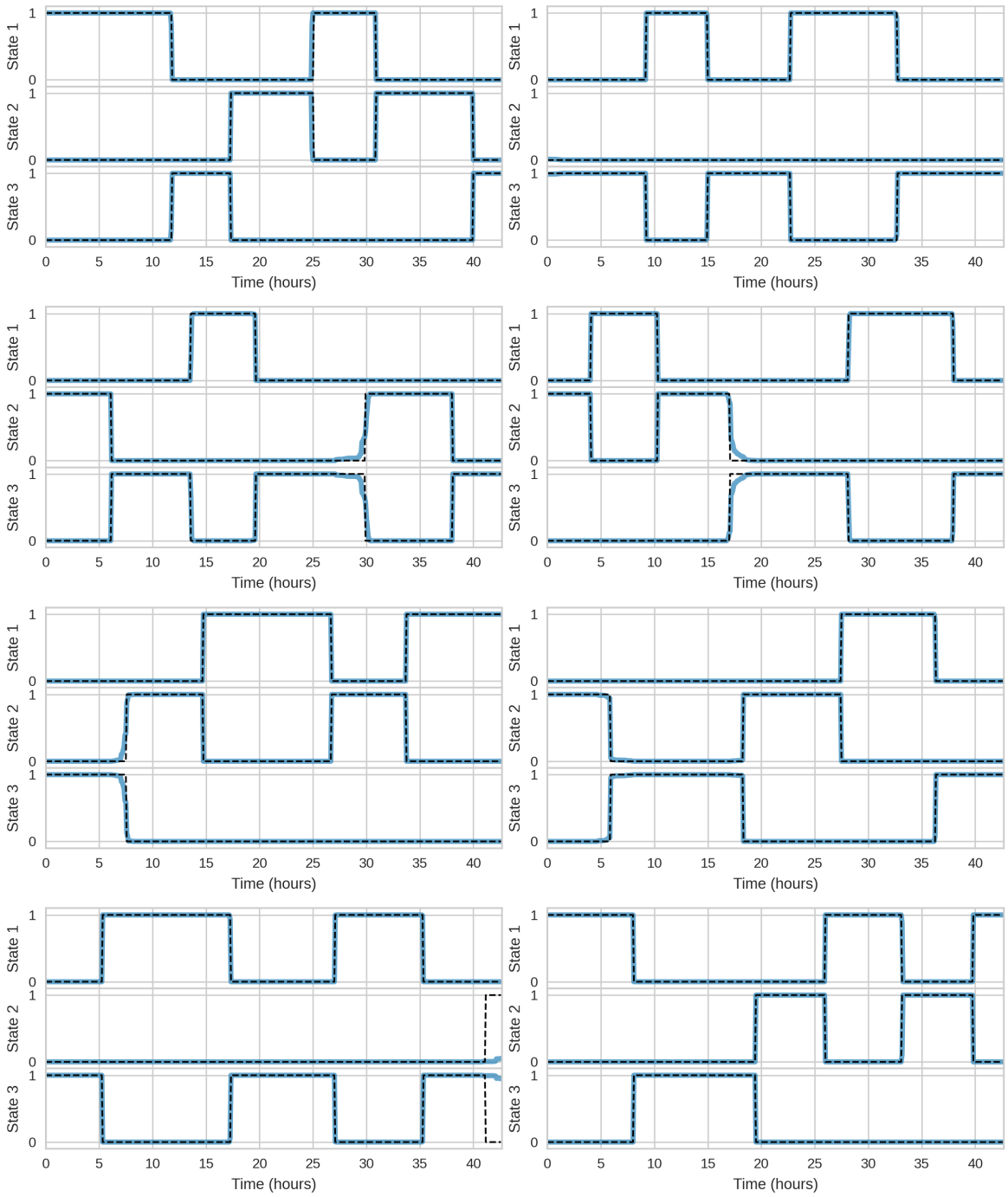


Figure 5.1: *Blue solid line:* Probability of being in a state. *Black dashed line:* Ground truth probability of being in a state.

0.001% and a tolerance of 2000 log-likelihood values below ε for the early stopping criterion. After early stopping the optimiser, the sampler was run for 50000 steps saving every $100th$ state sequence sample. We used the Adam optimiser (Kingma and Ba, 2017) with initial learning rate 0.1. We analysed the state sequence samples over a 24-hour window and reported it in Figure 5.3, whereas in Figure 5.4 the optimised model parameters are plotted.

In our 2-state model, the mean residence time was found to be about $12h$, indicating that

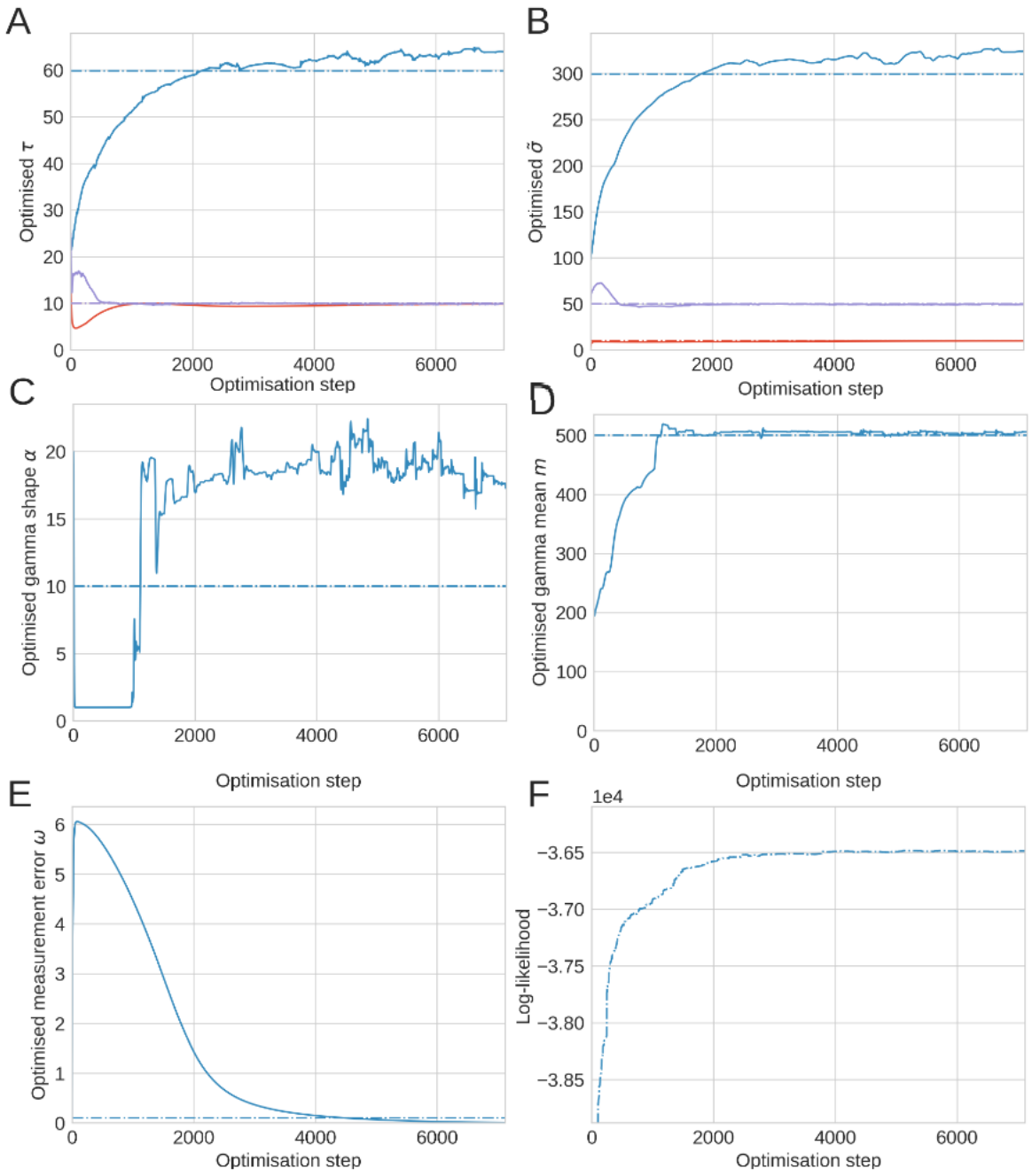


Figure 5.2: Optimised model parameters and associated log-likelihood summed across individuals. A-B) Optimised $\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\sigma}}$ (solid lines) with corresponding true values (dashed dotted lines). C-D) Solid lines: optimised gamma parameters α and m; dashed dotted line: true values. E) Solid line: optimised measurement error standard deviation; dashed dotted line: true value. F) Total expected complete log-likelihood per optimisation step.

what we see are the daily activity patterns of the sheep; state 1 (red) may be associated with a foraging or moving behaviour, with an autocorrelation function that decays faster and a higher speed, whereas state 2 (blue) represents a resting state, with lower speeds and longer correlated movement. Our results show agreement with a previous analysis that revealed

Symbol	Description	Initial value
$\boldsymbol{\tau}$	Autocorrelation time	$(0.1, 2.0)$
$\tilde{\boldsymbol{\sigma}}$	Long-term RMS displacement	$(0.2, 2.0)$
α	Gamma shape	100
m	Mean residence time	20
ω	Measurement error std	0.045

Table 5.2: Table of initial model parameter values.

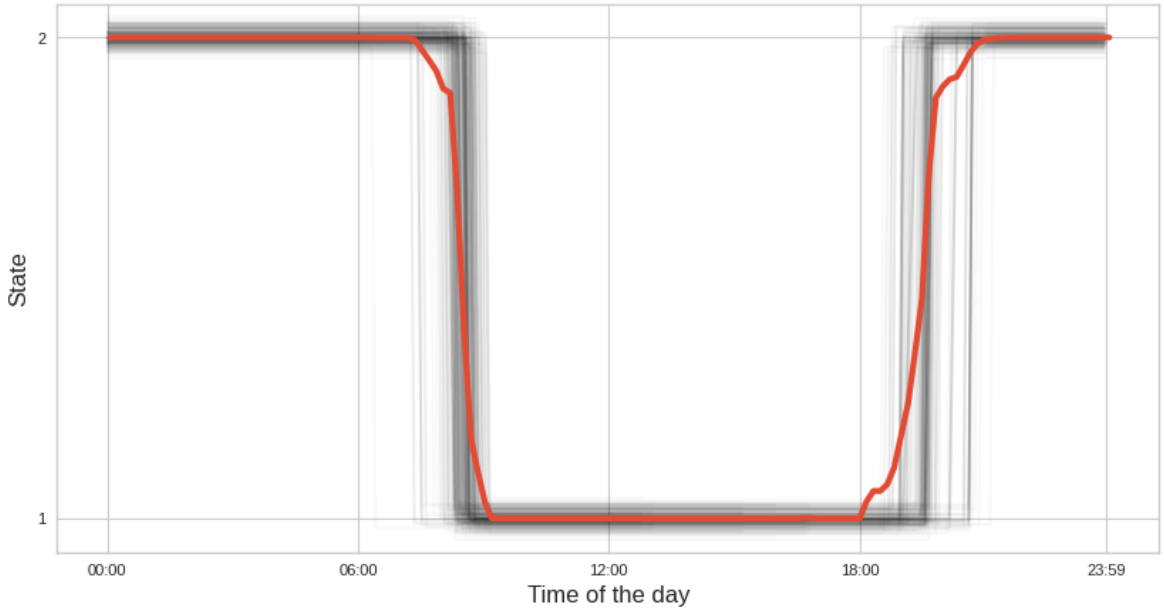


Figure 5.3: The 24-hour cycle of activity pattern of the sheep. The black lines represent the thinned samples for each trajectory. The red line represents the average cyclic pattern across all individuals over the total time period from May 1st to May 8th. The y-axis shows the state in which the sheep are throughout a day.

regular daily activity patterns in the sheep behaviour (Torney et al., 2021). These results also show the daily activity patterns of the sheep, suggesting two clear activity peaks with higher speeds between 09:00 and 21:00 as shown in Figure 5.3.

To check the convergence of the state sequence samples we ran multiple Geweke convergence tests (Geweke, 1991). Firstly, we considered the Geweke statistic for the full state sequences, thereby treating the behavioural state at each observation time as a parameter and comparing the first half of the chain with the second half. We excluded all observation times for which the chain contained only a single state value throughout, leaving 456 observation times and associated Geweke statistics. As we found all scores to be well within two standard deviations of zero (the maximum absolute z-score was 0.89) the diagnostic did not indicate a lack of convergence (see Fig. S1 for a plot of the values). Secondly, we ran a test on a summary statistic for the full state sequence, specifically we calculated the average time that the individual spent in each state for each MCMC sample. This gave 6 Geweke diagnostic val-

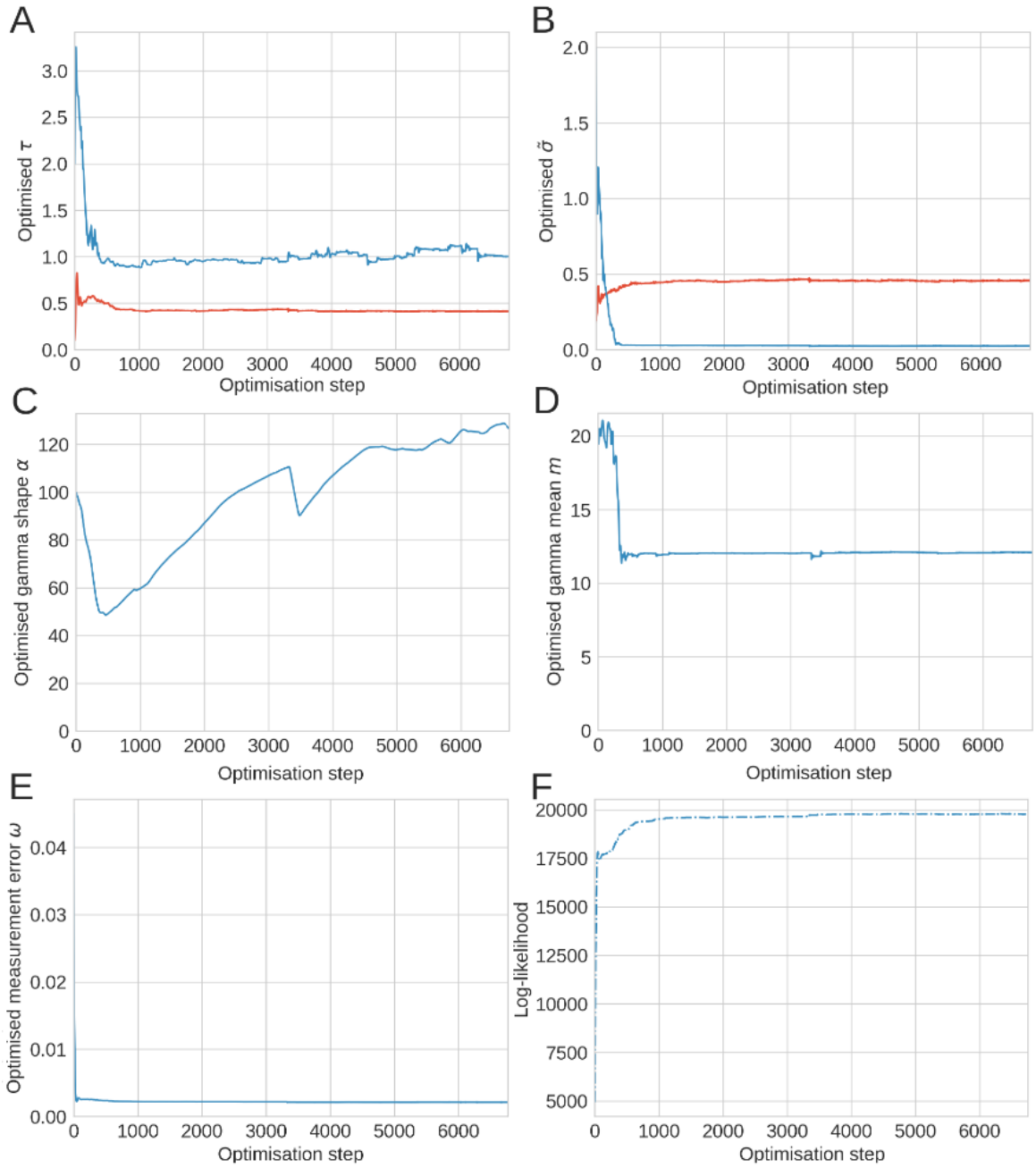


Figure 5.4: Optimised model parameters and associated log-likelihood summed across individuals. A-B) Optimised $\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\sigma}}$. C-D) optimised gamma parameters α and m. E) Optimised measurement error standard deviation. F) Expected complete log-likelihood summed across trajectories per optimisation step.

ues based on the summary statistic for each trajectory considered and the maximum absolute z-score was 0.4. Thus there was again no evidence that the chain had failed to converge.

5.5 Discussion

We have presented a new framework that can infer latent behavioural states as well as recover the model parameters from a continuous-time, multi-state model of animal movement. The model is described by a latent continuous-time behavioural process and an observation process consisting of a location process augmented with measurement error. A novel degree of flexibility is introduced in the behavioural process by making use of a continuous-time hidden semi-Markov model so that the dwell times are not constrained to follow an exponential distribution, thus avoiding making unrealistic assumptions on the animals' state-switching behaviour. The specification of the residence time distribution is arbitrary and we have used a gamma distribution in our work as its properties make up for more realistic assumptions on the animals' behaviour - thus our method may be regarded as a generalisation of standard hidden Markov models. However this specification can be changed on a case-by-case basis to accommodate the ecologist's prior assumptions. For example, in this work we have assumed that the residence time in each behavioural state follows the same distribution, however in order to introduce heterogeneity in the sojourn times each state could be assigned to a different distribution, or a bimodal distribution could be employed instead.

Our inference scheme was developed to overcome the computational limitations of existing methods. This was achieved by employing an MCEM algorithm whereby the non-analytical E-step was approximated by a Monte Carlo sum and the non-analytical M-step by a stochastic gradient descent scheme. Existing state-of-the-art methods that use continuous-time models perform inference of model parameters via maximum likelihood estimation using the Kalman filter (Michelot and Blackwell, 2019, Johnson et al., 2008) in combination with a hybrid MCMC scheme aimed at reconstructing the latent state sequence (Michelot and Blackwell, 2019, Blackwell, 2003). The scheme is a Metropolis-within-Gibbs sampling scheme and each iteration of the algorithm consists of an update for three groups, respectively an update for the underlying state sequence, an update for the movement parameters and an update for the transition rates of the IGM. Although such methods benefit from a proper uncertainty quantification in the parameter estimations, MCMC sampling schemes notoriously perform slowly with large datasets and calculation of the likelihood requires utilisation of the whole dataset. Given the increasing availability of high-frequency data and the improvements in the machine learning field, we believe that putting effort in the creation of an animal movement model that can leverage highly optimised machine learning libraries to fully make use of large datasets should be prioritised.

These innovations expand the array of methodologies available to the ecologists and we believe this method will positively impact new telemetry analyses as leveraging the information of big data is key to an increased understanding of animal movement.

Chapter 6

Applications to collective movement

In this chapter, we will revisit a concept introduced back in Chapter 3 and we will address the question of identifying the behavioural switches occurring at the group level applying the method developed in Chapter 5.

6.1 Introduction

In Chapter 3 we introduced a one-dimensional, self-propelled particle (SPP) model that was used to model the dynamics of desert locusts placed in a ring-shaped arena during an experiment (Buhl et al., 2006). By simulating the micro-level dynamics, we were then able to extract macro-level information about the group, specifically the group average velocity, and we showed how the group dynamics manifested two metastable states representing respectively ordered movement towards the left and ordered movement towards the right (Campioni et al., 2020). The group average velocity was therefore an indicator of the degree of order in the locusts' group: values approaching $|1|$ were an indicator of ordered, cohesive motion, namely the locusts moved in the same direction, whereas values around 0 were indicative of chaotic, disordered movement.

In Chapter 5 we developed a model aimed at reconstructing the history of changes in the behaviour of the animal, where in this context the word behaviour is synonym with different movement patterns. Starting from relocation data, we were able to individually reconstruct the daily activity patterns of three sheep during the first 8 days in May, and we showed how the three sheep were consistently active between $9am$ and $9pm$ and resting overnight. Following the idea introduced in Chapter 3, it would then be interesting to extract an order parameter (that is, the group-level metric) from the individual trajectories of the sheep flock and use it with the non-Markovian model described in Chapter 5 to identify the different group-level behaviours to see whether the behaviours that we identified with the three sheep were irregular patterns or whether the whole group adheres to those specific movement patterns.

In the subsequent sections we describe the dataset that we have used as well as the method employed for the analysis.

6.2 Materials and methods

In this section we are going to describe the dataset that we have used, we show how we extract the order parameter from the group and we describe the methodological framework that we employed for the analysis.

6.2.1 Data and data pre-processing

The data was collected from a long-term study of 58 female Merino sheep allowed to roam freely in a paddock of 700 hectares. Fieldwork was conducted at the "Campo Anexo Pilcaniyeu" from INTA (National Institute of Agricultural Technology) Bariloche, Patagonia, Argentina. All sheep were equipped with collars containing a GPS (CatLog-B, Perthold Engineering **www.perthold.de**), that was programmed to record location every five minutes

from February 2019 to December 2019, resulting in approximately 4 million measurements. Age and physical health were also recorded.

Some tags were faulty and did not work for the whole time span. In the effort to establish whether the movement patterns identified in Chapter 5 are characteristic of the whole sheep flock, we scanned the dataset and looked for a time window containing the most individuals. This was found to be in the first weeks of March, where we have records of 53 individuals out of the 58 total. We disregarded observations with associated HDOP>2 as well as consecutive observations that were associated with step lengths greater than 1 kilometer over a 5-minute interval. We selected the first 4096 fixes for each sheep, equalling 217088 observations ranging from March 1st to March 17th.

6.2.2 Order parameter

The scope of employing an order parameter is to extract group-level information that we can use to detect the group's different movement patterns. Given that the dataset contains the individual locations of 53 individuals belonging to the same sheep flock, to extract group-level information we take a bottom-up approach (Patterson et al., 2017a), that is, we extract an order parameter from the individual-level measurements. This group-level metric will be used for characterising and quantifying the degree of order or coordination within the group. We proceed as follows.

We start by calculating the velocity for each individual using numerical differentiation. Then, we establish a 5-minute time window, effectively creating a discrete moving average filter. Within this window, we group together all the individual velocities and their corresponding time points, a process applied to all the time series.

Once we have collected these grouped observations, we compute the average velocity over each 5-minute interval and consider its modulus, thus our order parameter is defined by the following equation:

$$\phi_t = \left| \frac{1}{\bar{N}_t} \sum_{i=1}^{\bar{N}_t} \mathbf{v}_{i,t} \right|, \tag{6.1}$$

where the time index t is taken over evey five minutes and $\bar{N}_t$ represents the total number of individuals measured at time t, which may vary due to potential missing data.

By taking the absolute value of the group's average velocity, we gain insight into the degree of cohesiveness in the group. Indeed, higher values of ϕ_t correspond to more ordered movement, meaning the group moves together in the same direction (Vicsek et al., 1995). Conversely lower values of ϕ_t are related to more chaotic, disordered movement, where each individual moves independently of one another. We plot the order parameter in Fig. 6.1, where we can see that the group tends to move quite slowly during the first weeks of March, never exceeding a speed of $2km/h$. Note that lower values of ϕ_t may indicate either chaotic movement or absence of movement.

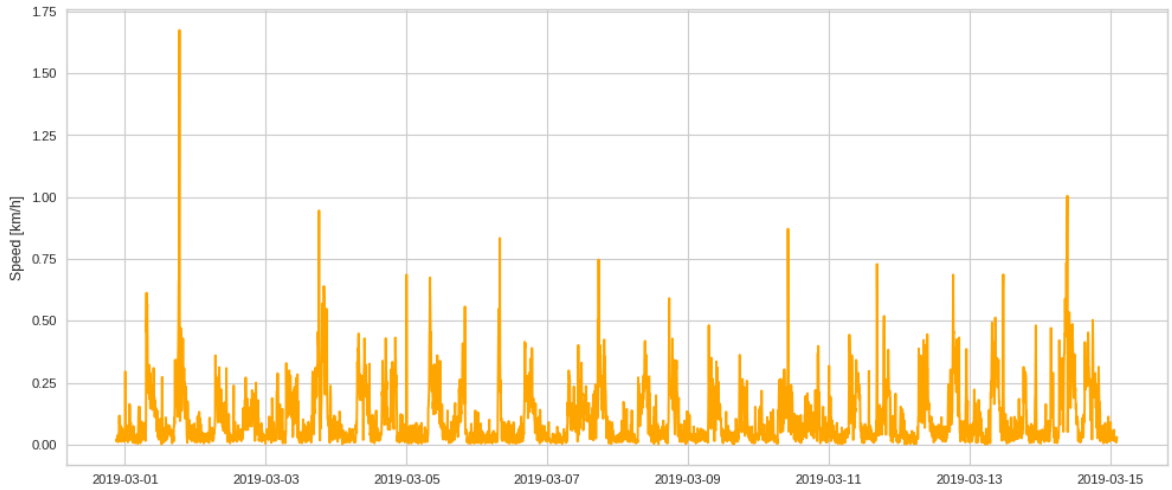


Figure 6.1: The extracted order parameter. This is the modulus of the group average velocity calculated over fixed five-minute intervals, Eqn. 6.1.

6.2.3 Flexible MCEM

In Chapter 5 we introduced a novel model of animal movement based on a movement process described, by continuous-time correlated random walk (CTCRW) model, and a latent non-Markovian process on which the movement parameters are dependent. A novel flexibility of the model was introduced by modelling the distribution of the sojourn times of each state with an arbitrary distribution chosen by the practitioner on case-to-case basis. However, the underlying assumptions of that model are firstly that each state is identically distributed, and secondly that the residence time in each state is the same. This might not be ideal in certain scenarios, for example in the study of air-breathing marine mammals in which case we know that the residence time of the dive-in behaviour is different to the residence time of the breathing behaviour.

For this chapter, we introduce a further layer of flexibility in the model introduced in Chapter 5 by modelling the sojourn time in each state by its own probability distribution. That means that for an n-state model we will define n prior distributions on the residence times of each state and optimise their parameters independently. Although this implies that we are now optimising more parameters and therefore we could potentially increase the computational cost of the algorithm, by doing do we efficiently augment the model with a greater degree of flexibility which we believe will be also beneficial to the general practitioner.

6.3 Results

Given that our aim is to identify the switches between ordered and disordered movement identified by higher and lower values of ϕ_t, respectively, we employ a two-state model and we specify a gamma distribution for each state to model the residence times in each state

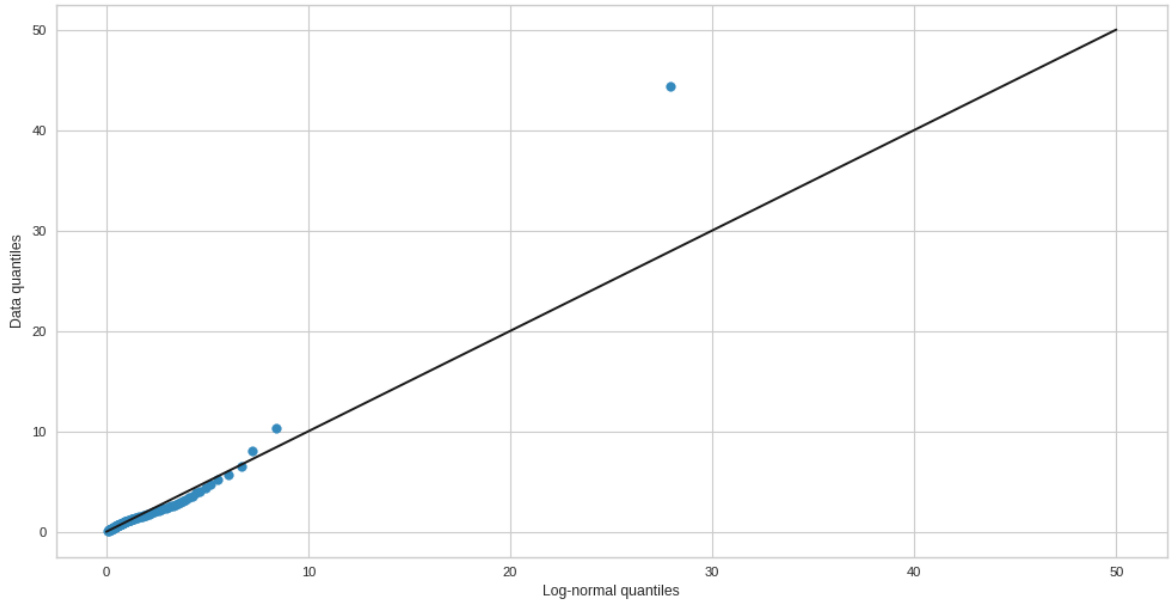


Figure 6.2: QQ plot to check whether our assumption that a log-normal distribution for the order parameter is consistent with the data is true. The data distribution and the log-normal distribution agree except for one outlier. This may be due to the noisy nature of the data.

independently. Furthermore, given that the values of the order parameter are positive, we employ a log-normal distribution $Lognormal(\mu, \sigma)$ for the conditional likelihood, where if a random variable $X \sim Lognormal(\mu, \sigma)$ then the parameters μ and σ represent the mean and standard deviation of $ln(X)$. As we can see in the QQ plot in Fig. 6.2, our assumption that a log-normal distribution is consistent with the data is verified except for one outlier.

In Fig. 6.3 we plot the order parameter together with the probability of being in state 2 across the first weeks of March. We divided the trajectory into 4 subplots that contain, respectively, the values of the order parameter from March 1st to March 5th, from March 5th to March 9th, from March 9th to March 12th and from March 12th to March 15th. In in Fig. 6.4 we report the optimised model parameters. Similarly to what we did in Chapter 5, we ran the sampler by specifying an early stopping criterion with a 0.001% improvement threshold.

As we can see, the sheep flock exhibits similar dynamics to the individual dynamics found in Fig. 5.3: the group is more active during daytime (state 1) and less active during nighttime (state 2), represented by different parameter combinations for the log-normal distribution. Specifically, lower standard deviation values and higher mean values are associated with the active state and higher standard deviation values and lower mean values are associated with the resting state. However, given the additional flexibility that this model has, through which each state is modelled by a different distribution, we can see how the sheep are not always active between $9am$ and $9pm$ but also rest in between. This is reflected by the optimised sojourn time prior distribution parameters. Indeed, the gamma parameters for state 1 were $\alpha = 15$, mean $= 4h$, whereas the values for state 2 were $\alpha = 2$ and mean $= 9h$, which can be

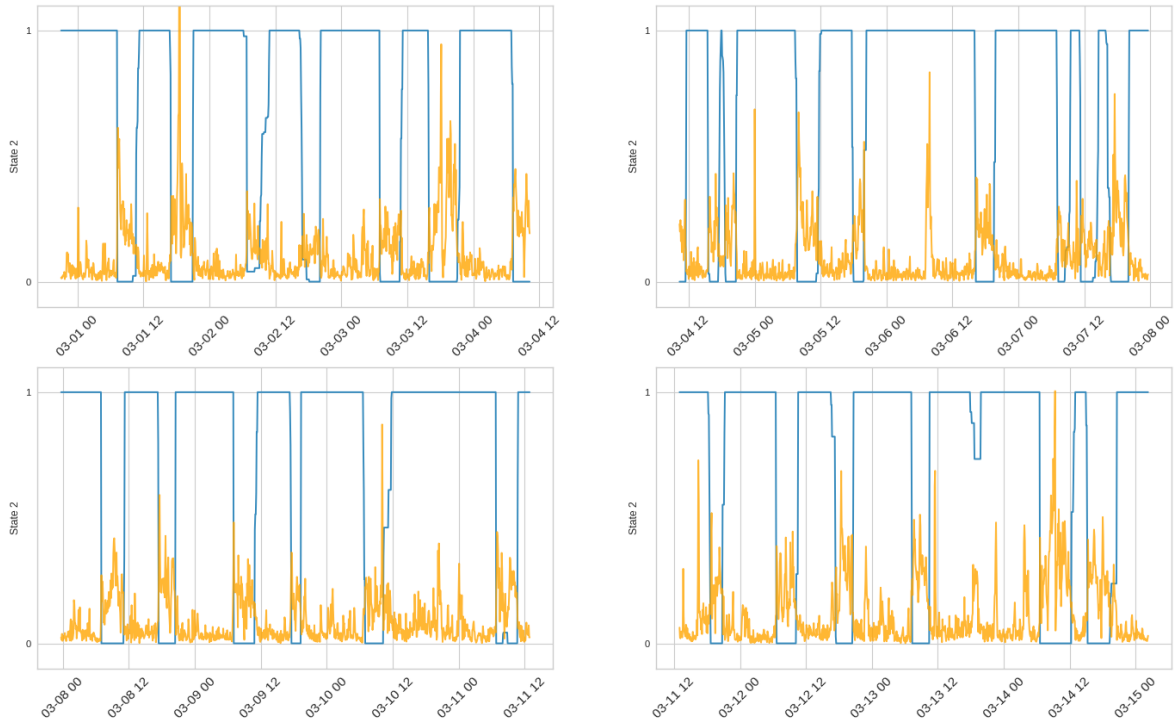


Figure 6.3: Order parameter (orange) together with the probability of being in state 2 (blue).

seen in Fig. 6.3, where in some cases the sojourn in state 2 is longer than the sojourn in state 1 and we can also see how the group alternates between the resting state and the active state between $9am$ and $9pm$, but it consistently rests overnight. We plot a kernel density estimation for the distribution of the residence times in each state in Figs. 6.5 and 6.6.

6.4 Discussion

In this chapter, we extracted an order parameter from the dataset containing the positions of sheep belonging to the same flock. We then modified the non-Markovian method from Chapter 5 by defining a prior distribution for each state and applied it to the order parameter. The results were in agreement with the results from Chapter 5, however the latest results may be indicative that the previous method may have lacked enough flexibility to capture the state switches occurring between the sheep active hours.

This immediately suggests a future route for this work: we could apply the more flexible method to the individual trajectories and compare the new results with the old results from Chapter 5. Furthermore, given the availability of individual trajectories, we could leverage both the individual analysis of the sheep and the group-level analysis to investigate the behavioural ecology of the group. Specifically, we could answer questions concerning the driving intra-group relationship of the group, meaning we could investigate whether the sheep flock manifests leader-follower dynamics or whether the sheep move randomly.

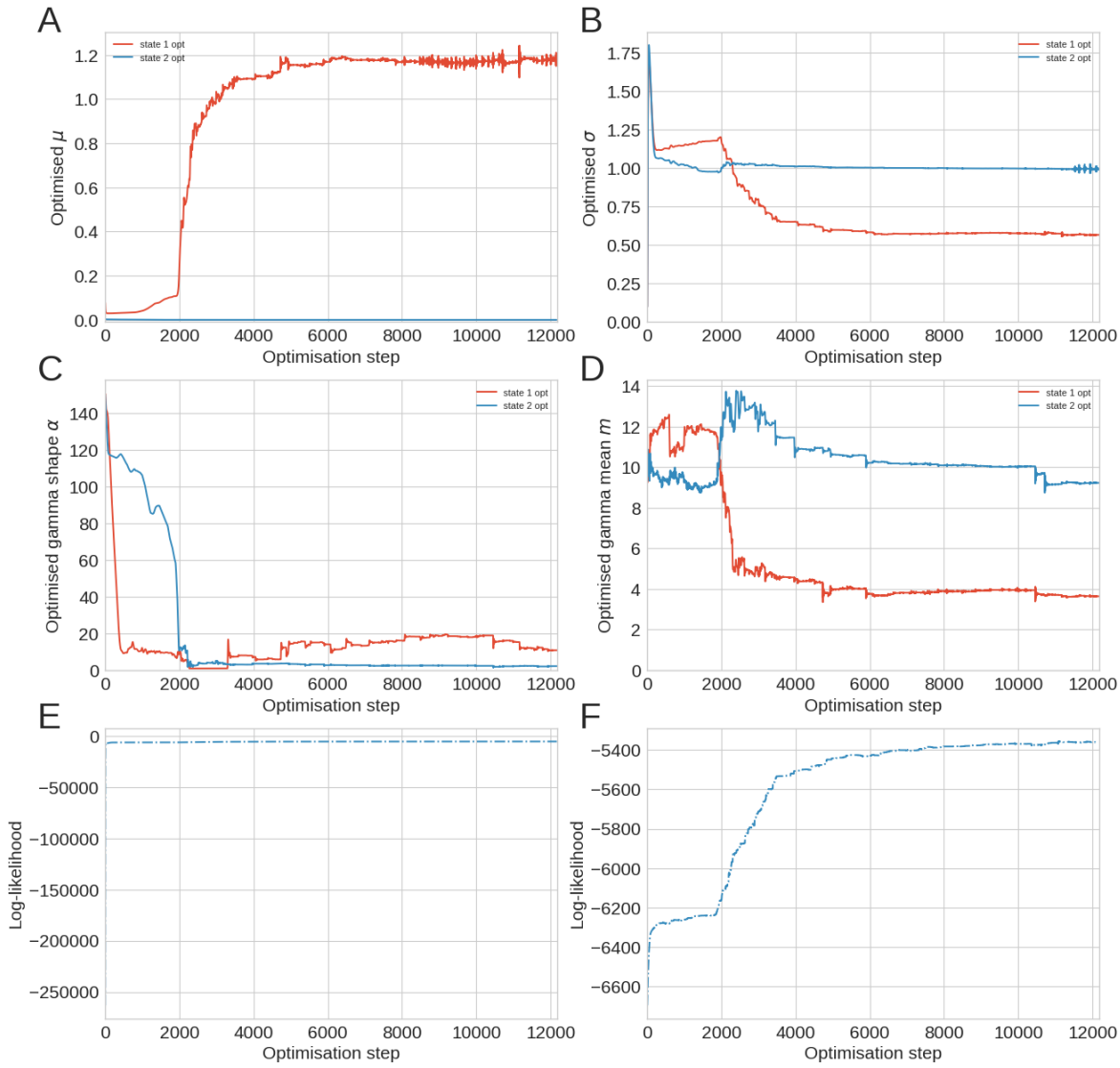


Figure 6.4: Optimised model parameters for the order parameter analysis. A-B) optimised log-normal distribution parameters μ and σ; C-D) optimised gamma parameters for each state; E) total log-likelihood; F) total log-likelihood without the first 100 values for a better visualisation.

Generally speaking, delving into the intricacies of leadership in animal groups is far from a straightforward endeavour. The path is fraught with challenges - technological, analytical and conceptual (Strandburg-Peshkin et al., 2018) - that have always posed a challenge to our quest to understand this central aspect of social living. However, we believe that by leveraging the availability of our high-frequency dataset, which stands as a doorway to the minute-to-minute life activities of the sheep, and the scalable, flexible non-Markovian model that we have developed, we could be in the position to perform more thorough analyses and address more profound questions regarding the behavioural ecology of the sheep flock.

In the literature we find different methods that are aimed at assessing whether a group of animals present a leader (Strandburg-Peshkin et al., 2018). One way to identify the presence

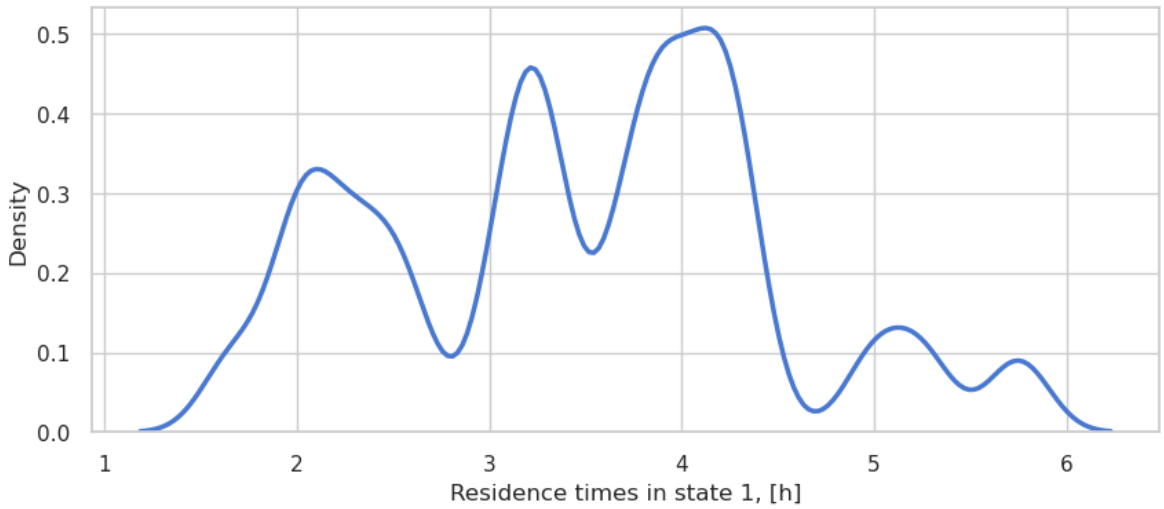


Figure 6.5: Kernel density estimation for the distribution of the residence times in state 1.

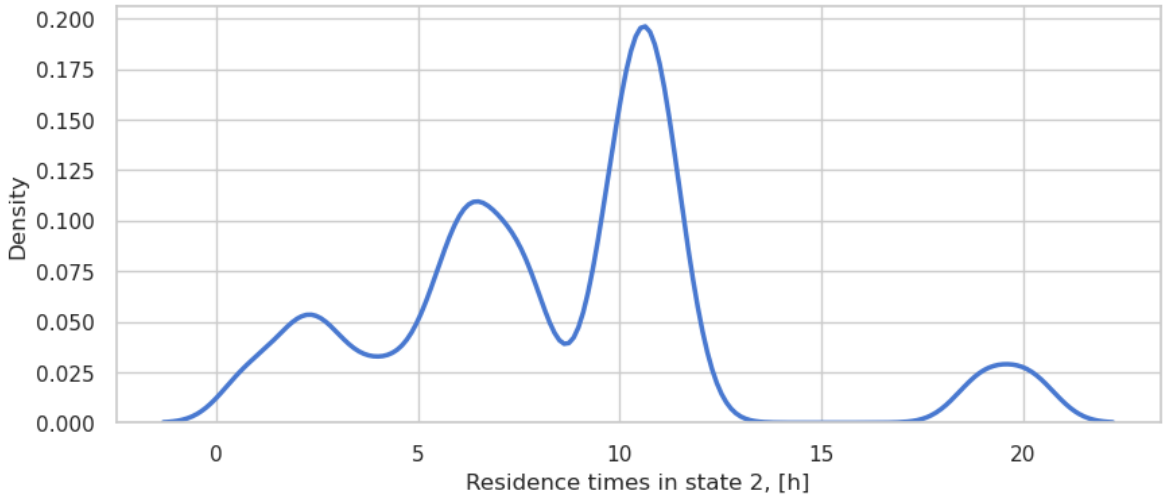


Figure 6.6: Kernel density estimation for the distribution of the residence times in state 2.

of a leader is to consider the intra-group spatial position of individuals. In particular, according to this criterion, a leader is associated with the individual that is placed in front of the group during transitions. This is an easy way to establish the presence of a leader, and it has been employed in many studies, such as in a study of spotted hyenas (Smith et al., 2015). However, the disadvantage of this method is that being at the front-most position does not necessarily imply influence or leadership. A second method takes into account the changes in direction; specifically, time-lagged correlation between any two individuals' headings (or some other metric of direction) is calculated and used to recreate a leader-follower network (Nagy et al., 2010). However, this approach requires that the animals are continuously on the move so that their directional changes have significance. If the animals aren't well-coordinated in their movements, it becomes challenging to reliably measure correlations in their travel directions because they might be mistaken for random noise (Strandburg-Peshkin

et al., 2018).

Another method assesses leadership in a complementary way (Strandburg-Peshkin et al., 2018) - from the outcome of decision. That means, given that the group has reached the destination, what individuals have benefited from this decision the most? This method, however, presents many drawbacks as knowing the preferred locations of each individual is not always possible, and even if it was, in the scenario where the group has reached a location preferred by two or more individuals, it is not possible to infer which animal has exerted the most influence.

Given the nature of the data and the applicability of our method, we believe that the best approach to identifying a leader in the sheep flock would be to identify a potential movement-initiator. In the studies that used this criterion, they labelled as leader who initiated movement at the departure from either sleeping locations (Stueckle and Zinner, 2008) or foraging sites (Tokuyama and Furuichi, 2017). Thus, as a step forward for our research, we could apply the method to the individual trajectories and then, by comparing the reconstructed individual switches to the reconstructed group-level switches, we would be in the position to assess whether the movement is repeatedly initiated by the same individual, or individuals, or not.

Chapter 7

Conclusions

This thesis has been focused on statistical modelling of animal movement. The methodological contributions that we have given are found in Chapters 3-5, and in Chapter 6 we have analysed telemetry data representing the location of Merino sheep with the method introduced in Chapter 5.

In Chapter 3, we have introduced a novel statistical framework in the context of multiscale inference, that is, inference that links macroscale properties to microscale properties. Specifically, our method performs inference of microscale parameters from macroscale measurements of interacting systems. We have used concepts from equation-free modelling (Kevrekidis et al., 2003) and augmented those ideas with sparse Gaussian process regression (Gardner et al., 2018), which has enabled us to get approximations of the probability density of macroscale observations, while simultaneously calculating the associated uncertainty caused by the use of a finite number of microscale simulations coming from the *lift* equation-free move. This has allowed us to construct a fast, adaptive MCMC sampler that employs a second Gaussian process to emulate the log-likelihood surface. In particular, we have employed an adaptive Metropolis-Hastings algorithm that was based on Conrad et al. (2016) with several modifications. Notably, the local Gaussian process approximation was replaced with a sparse Gaussian process that allowed us to use multiple samples from the stationary probability posterior distribution for each parameter set, saving us from spending computational effort to run microscale simulations at each step of the sampler. By passing these multiple samples into the algorithm, the emulator GP was able to learn an effective observation noise in the simulator that arises due to the stochastic nature of the microscale model.

Although we have applied our framework to a simple one-dimensional simulation model, our approach can be applied to any multiscale system that can be modelled at the microscale but can only be easily observed at the macroscale. Hence, an interesting future development of the method could extend the framework to those higher-dimensional macroscale systems for which the presented ideas of using GPs to approximate the unavailable stationary proba-

bility density of interacting systems would hold.

The novel statistical framework presented in Chapter 5 tackles a different inferential task. The aim was indeed to create a statistical framework that could solve the so-called switching problem, that is, assigning each observation to a different state, which is a proxy of an animal behaviour, based on quantitatively different characteristic of the observations. In the literature, hidden Markov models are the standard tool employed in this context. This model is formulated in discrete-time and is intuitive to apply, however lacks of flexibility as the dwell times in the states are constrained to follow a geometric distribution. Our aim has therefore been to create a flexible model that allowed for different distributions to be used and that was scalable to large datasets.

This was achieved by employing a latent continuous-time semi-Markov model to model the behavioural process and an integrated OU process to model the location process, which was then augmented with measurement error. Employing a semi-Markov process allowed us to introduce a novel degree of flexibility; we were able to reconstruct the history of the state switches as well as optimise the model parameters by employing a Monte Carlo Expectation-Maximisation (MCEM) algorithm. To this end, a semi-Markov chains generator was needed as we proposed new state sequences at each step of the algorithm. Chapter 4 was dedicated to illustrating the different strategies we adopted as proposal mechanisms, and that were not successful. The algorithms therein presented were the virtual state method, which tended to propose short-segments state sequences, the reversible rescaling method, that was equipped with 3 moved that didn't allow the algorithm to explore all state sequence configurations, and the reversible mutation method, which was the method that was employed in the MCEM algorithm. Our MCEM based method was also able to overcome the computational limitations of existing methods. Indeed, state-of-the-art methods rely on MCMC sampling techniques that notoriously perform slowly with large datasets given the need to use the whole dataset to calculate the model likelihood.

The MCEM framework was further expanded in Chapter 6, where the residence time for each state was modelled by a different probability distribution. This method was used to analyse a group-level metric described by the modulus of the group average velocity taken every five minutes. This order parameter was indicative of the degree of cohesiveness in the movement of the sheep, and we therefore defined a 2-state model for identifying the switches between higher values of the order parameter (associated with more ordered movement) and lower values of the order parameter (associated with more disordered movement).

As discussed in Chapter 6, future work could investigate the behavioural ecology of the sheep flock by comparing the individual switches to the order parameter switches in the effort to identify a leader. Another nice future investment on this project could lead to the creation of a user-friendly package to incentivise the movement ecologists to use the powerful machine learning libraries from TensorFlow, which make handling large dataset relatively

straightforward by constructing data input pipelines. This would enrich future research as we believe leveraging the information of big data is key to an increased understanding of animal movement.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

Anderson, P. W. More is different. *Science*, 177(4047):393–396, 1972.

Andrieu, C. and Roberts, G. O. Particle Markov chain Monte Carlo methods. *The Annals of Statistics*, 37(2):697 – 725, 2009. doi: 10.1214/07-AOS574. URL `https://doi.org/10.1214/07-AOS574`.

Andrieu, C., Doucet, A., and Holenstein, R. The pseudo-marginal approach for efficient Monte Carlo computations. *Journal of the Royal Statistical Society, Series B*, 37(72):269–342, 2010.

Batz, P., Ruttor, A., and Opper, M. Approximate Bayes learning of stochastic differential equations. *Physical Review E*, 98(2):022109, 2018.

Bellomo, N., Bellouquid, A., Tao, Y., and Winkler, M. Toward a mathematical theory of Keller–Segel models of pattern formation in biological tissues. *Mathematical Models and Methods in Applied Sciences*, 25(09):1663–1763, 2015.

Benhamou, S. Detecting an Orientation Component in Animal Paths when the Preferred Direction is Individual-dependent. *Ecology*, 87(2):518–528, 2006. doi: https://doi.org/10.1890/05-0495.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blackwell, P. G. Bayesian Inference for Markov Processes with Diffusion and Discrete Components. *Biometrika*, 90(3):613–627, 2003.

Blackwell, P. Random diffusion models for animal movement. *Ecological Modelling*, 100 (1):87–102, 1997. ISSN 0304-3800.

Bogdan Doytchinov and Rachel Irby. Time Discretization of Markov Chains. *Pi Mu Epsilon Journal*, 13:69–82, 2010.

Buhl, J., Sumpter, D. J. T., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., and Simpson, S. J. From Disorder to Order in marching locusts. *Science*, 312(5778):1402–1406, 2006. doi: 10.1126/science.1125142.

Cagnacci, F., Boitani, L., Powell, R. A., and Boyce, M. S. Animal ecology meets GPS-based radiotelemetry: a perfect storm of opportunities and challenges. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365:2157–2162, 2010.

Campioni, N., Husmeier, D., Morales, J. M., Gaskell, J., and Torney, C. J. Modelling multiscale collective behavior with Gaussian processes. *Proceedings of the Second International Conference on Statistics: Theory and Applications (ICSTA'20)*, 2020. doi: 10.11159/icsta20.124.

Campioni, N., Husmeier, D., Morales, J., Gaskell, J., and Torney, C. J. Inferring microscale properties of interacting systems from macroscale observations. *Phys. Rev. Res.*, 3:043074, Oct 2021. doi: 10.1103/PhysRevResearch.3.043074.

Coffey, W. T. and Kalmykov, Y. P. *The Langevin Equation*. WORLD SCIENTIFIC, 3rd edition, 2012. doi: 10.1142/8195.

Colebank, M. J., Paun, L. M., Qureshi, M. U., Chesler, N., Husmeier, D., Olufsen, M. S., and Fix, L. E. Influence of image segmentation on one-dimensional fluid dynamics predictions in the mouse pulmonary arteries. *Journal of The Royal Society Interface*, 16(159), 2019.

Conrad, P. R., Marzouk, Y. M., Pillai, N. S., and Smith, A. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *J Am Stat Assoc*, 111(516):1591–1607, 2016.

Cramer, H. *Mathematical methods of statistics*. Princeton University Press Princeton, 1946.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020a. ISSN 0027-8424. doi: 10.1073/pnas.1912789117.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 2020b.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

Czirók, A., Barabási, A.-L., and Vicsek, T. Collective Motion of Self-Propelled Particles: Kinetic Phase Transition in One Dimension. *Phys. Rev. Lett.*, 82:209–212, Jan 1999.

Demirel, G., Vazquez, F., Böhme, G., and Gross, T. Moment-closure approximations for discrete adaptive networks. *Physica D: Nonlinear Phenomena*, 267:68–80, 2014.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal Of The Royal Statistical Society, Series B*, 39:1–38, 1977.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Durbin, J. and Koopman, S. J. *Time Series Analysis by State Space Methods*. Oxford University Press, 2012.

Gardiner, C. *Stochastic methods*, volume 4. Springer Berlin, 2009.

Gardner, P., Rogers, T. J., Lord, C., and Barthorpe, R. J. Sparse Gaussian Process Emulators for Surrogate Design Modelling. In *Applied Mechanics and Materials*, volume 885, pages 18–31. Trans Tech Publ, 2018.

Gaskell, J., Campioni, N., Morales, J. M., Husmeier, D., and Torney, C. J. Inferring the interaction rules of complex systems with graph neural networks and approximate Bayesian computation. *Journal of The Royal Society Interface*, 20(198):20220676, 2023. doi: 10.1098/rsif.2022.0676.

Geweke, J. F. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Technical Report 148, Federal Reserve Bank of Minneapolis, 11 1991.

Glennie, R., Adam, T., Leos-Barajas, V., Michelot, T., Photopoulou, T., and McClintock, B. T. Hidden Markov models: Pitfalls and opportunities in ecology. *Methods in Ecology and Evolution*, 14(1):43–56, 2023. doi: https://doi.org/10.1111/2041-210X.13801.

Goldberg, P. W., Williams, C. K., and Bishop, C. M. Regression with input-dependent noise: A Gaussian process treatment. In *Advances in neural information processing systems*, pages 493–499, 1998.

Gramacy, R. B. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press, 2020.

Green, P. J. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, 82(4):711–732, 1995.

Gurarie, E., Fleming, C. H., Fagan, W. F., Laidre, K. L., Hernandez-Pliego, J., and Ovaskainen, O. Correlated velocity models as a fundamental unit of animal movement: synthesis and applications. *Movement Ecology*, 5, 2017. doi: 10.1186/s40462-017-0103-3.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer New York, NY, 2 edition, 2 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7.

Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian Processes for Big Data. *ArXiv*, abs/1309.6835, 2013.

Hobolth, A. and Stone, E. A. Simulation from Endpoint-Conditioned, Continuous-Time Markov Chains on a Finite State Space, with Applications to Molecular Evolution. *The Annals of Applied Statistics*, 3(3):1204–1231, 2009. ISSN 19326157, 19417330.

Hooten, M., Johnson, D., McClintock, B., and Morales, J. *Animal Movement: Statistical Models for Telemetry Data*. CRC Press, 2017a. ISBN 9781466582156.

Hooten, M., King, R., and Langrock, R. Guest Editor's Introduction to the Special Issue on "Animal Movement Modeling". *Journal of Agricultural, Biological, and Environmental Statistics*, 22:224–231, 2017b.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

Ito, K. and McKean, H. P. Diffusion Processes and their Sample Paths. *Journal of the London Mathematical Society*, s1-42(1):186–187, 1967. doi: https://doi.org/10.1112/jlms/s1-42.1.186b.

Johnson, D. S., London, J. M., Lea, M.-A., and Durban, J. W. Continuous-time Correlated Random Walk Model for Animal Telemetry Data. *Ecology*, 89:1208–1215, 2008.

Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Kelly, F. and Yudovina, E. Stochastic networks. *Stochastic Networks*, 01 2012.

Kerman, S., Brown, D., and Goodrich, M. A. Supporting human interaction with robust robot swarms. In *2012 5th International Symposium on Resilient Control Systems*, pages 197–202, 2012.

Kevrekidis, I. G., Gear, C. W., Hyman, J. M., Kevrekidid, P. G., Runborg, O., Theodoropoulos, C., et al. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

King, A. A., Nguyen, D., and Ionides, E. L. Statistical Inference for Partially Observed Markov Processes via the R Package pomp. *Journal of Statistical Software, Articles*, 69 (12):1–43, 2016. ISSN 1548-7660. doi: 10.18637/jss.v069.i12. URL `https://www.jstatsoft.org/v069/i12`.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv*, 2017.

Kullback, S. and Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694.

Langrock, R., Marques, T., Baird, R., and Thomas, L. Modeling the Diving Behavior of Whales: A Latent-Variable Approach with Feedback and Semi-Markovian Components. *Journal of Agricultural, Biological, and Environmental Statistics*, 19:82–100, 03 2013. doi: 10.1007/s13253-013-0158-6.

Lázaro-Gredilla, M. and Titsias, M. K. Variational heteroscedastic Gaussian process regression. In *ICML*, 2011.

Levine, R. A. and Casella, G. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10:422–439, 2001.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017.

McClintock, B., Johnson, D., Hooten, M., Ver Hoef, J., and Morales, J. When to be discrete: the importance of time formulation in understanding animal movement. *Movement Ecology*, 2:21, 10 2014. doi: 10.1186/s40462-014-0021-6.

McClintock, B. T. and Michelot, T. momentuHMM: R package for generalized hidden Markov models of animal movement. *Methods in Ecology and Evolution*, 9(6):1518–1530, 2018. doi: 10.1111/2041-210X.12995.

Meinhardt, H. *Models of Biological Pattern Formation*. Academic Press, 1982.

Meinhold, R. J. and Singpurwalla, N. D. Understanding the Kalman Filter. *The American Statistician*, 37(2):123–127, 1983. ISSN 00031305.

Michelot, T., Langrock, R., and Patterson, T. A. moveHMM: an R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, 7(11):1308–1315, 2016.

Michelot, T. and Blackwell, P. G. State-switching continuous-time correlated random walks. *Methods in Ecology and Evolution*, 10:637–649, 2019.

Morales, J. M., Haydon, D. T., Frair, J., Holsinger, K. E., and Fryxell, J. M. Extracting more out of relocation data: Building movement models as mixtures of random walks. *Ecology*, 85:2436–2445, 2004.

Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Nagy, M., Akos, Z., Biro, D., and Vicsek, T. Hierarchical group dynamics in pigeon flocks. *Nature*, 464:890–3, 04 2010. doi: 10.1038/nature08891.

Nathan, R., Monk, C. T., Arlinghaus, R., Adam, T., Alós, J., Assaf, M., Baktoft, H., Beardsworth, C. E., Bertram, M. G., Bijleveld, A. I., Brodin, T., Brooks, J. L., Campos-Candela, A., Cooke, S. J., Gjelland, K. , Gupte, P. R., Harel, R., Hellström, G., Jeltsch, F., Killen, S. S., Klefoth, T., Langrock, R., Lennox, R. J., Lourie, E., Madden, J. R., Orchan, Y., Pauwels, I. S., Říha, M., Roeleke, M., Schlägel, U. E., Shohami, D., Signer, J., Toledo, S., Vilk, O., Westrelin, S., Whiteside, M. A., and Jarić, I. Big-data approaches lead to an increased understanding of the ecology of animal movement. *Science*, 375(6582), 2022. doi: 10.1126/science.abg1780.

Neal, R. M. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer Verlag, 1996.

Nguyen, H. C., Zecchina, R., and Berg, J. Inverse statistical problems: from the inverse Ising problem to data science. *Advances in Physics*, 66(3):197–261, 2017.

Nielsen, S. F. The Stochastic EM Algorithm: Estimation and Asymptotic Results. *Bernoulli*, 6(3):457–489, 2000.

Opper, M. and Archambeau, C. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

Owen, J., Wilkinson, D., and Gillespie, C. Likelihood free inference for Markov processes: a comparison. *Statistical Applications in Genetics and Molecular Biology*, 14:189 – 209, 2015.

Patterson, T., Parton, A., Langrock, R., Blackwell, P., Thomas, L., and King, R. Statistical modelling of individual animal movement: an overview of key methods and a discussion of practical challenges. *AStA Advances in Statistical Analysis*, 101, 07 2017a.

Patterson, T. A., Parton, A., Langrock, R., Blackwell, P. G., Thomas, L., and King, R. Statistical modelling of individual animal movement: an overview of key methods and a discussion of practical challenges. *AStA Advances in Statistical Analysis*, 101(4):399–438, 2017b. doi: 10.1007/s10182-017-0302-7.

Paun, I., Husmeier, D., Hopcraft, J. G. C., Masolele, M. M., and Torney, C. J. Inferring spatially varying animal movement characteristics using a hierarchical continuous-time velocity model. *Ecology Letters*, 25(12):2726–2738, 2022. doi: https://doi.org/10.1111/ele.14117.

Paun, L. M., Colebank, M. J., Olufsen, M. S., Hill, N. A., and Husmeier, D. Assessing model mismatch and model selection in a Bayesian uncertainty quantification analysis of a fluid-dynamics model of pulmonary blood circulation. *Journal of The Royal Society Interface*, 17(173), 2020.

Pohle, J., Langrock, R., van Beest, F., and Schmidt, N. Selecting the Number of States in Hidden Markov Models - Pitfalls, Practical Challenges and Pragmatic Solutions. *Journal of Agricultural Biological and Environmental Statistics*, 01 2017. doi: 10.1007/s13253-017-0283-8.

Pokharel, G. and Deardon, R. Gaussian process emulators for spatial individual-level models of infectious disease. *Canadian Journal of Statistics*, 44(4):480–501, 2016.

Prokopenko, M., Boschetti, F., and Ryan, A. J. An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28, 2009.

Ragwitz, M. and Kantz, H. Indispensable finite time corrections for Fokker-Planck equations from time series data. *Physical Review Letters*, 87(25):254501, 2001.

Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Renshaw, E. and Henderson, R. The Correlated Random Walk. *Journal of Applied Probability*, 18(2):403–414, 1981.

Richardson, S. and Green, P. J. On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

Risken, H. *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer Series in Synergetics. Springer Berlin Heidelberg, 2012. ISBN 9783642968075. URL `https://books.google.co.uk/books?id=dXvpCAAAQBAJ`.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387212396.

Robert, C. P., Ryden, T., and Titterington, D. M. Bayesian Inference in Hidden Markov Models through the Reversible Jump Markov Chain Monte Carlo Method. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(1):57–75, 2000.

Ruiz-Suarez, S., Leos-Barajas, V., and Morales, J. M. Hidden Markov and Semi-Markov Models When and Why are These Models Useful for Classifying States in Time Series Data? *Journal of Agricultural, Biological and Environmental Statistics*, pages 1–25, 2022. doi: 10.1007/s13253-021-00483-.

Santner, J. T., Williams, J. B., and Notz, I. W. *The Design and Analysis of Computer Experiments*. New York, NY: Springer Series in Statistics. Springer New York, 2003.

Saul, A. D., Hensman, J., Vehtari, A., and Lawrence, N. D. Chained Gaussian Processes. In Gretton, A. and Robert, C., editors, *Proceedings of the Nineteenth International Workshop on Artificial Intelligence and Statistics*, volume 51, pages 1431–1440. PMLR, 2016.

Sellier, M. Inverse problems in free surface flows: a review. *Acta Mechanica*, 227, 03 2016. doi: 10.1007/s00707-015-1477-1.

Sethna, J. *Statistical mechanics: entropy, order parameters, and complexity*, volume 14. Oxford University Press, 2006.

Shalizi, C. R. Methods and techniques of complex systems science: An overview. In *Complex systems science in biomedicine*, pages 33–114. Springer, 2006.

Smith, J. E., Estrada, J. R., Richards, H. R., Dawes, S. E., Mitsos, K., and Holekamp, K. E. Collective movements, leadership and consensus costs at reunions in spotted hyaenas. *Animal Behaviour*, 105:187–200, 2015. ISSN 0003-3472. doi: https://doi.org/10.1016/j.anbehav.2015.04.023.

Snelson, E. and Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18:1257–1264, 2005.

Sood, V., Antal, T., and Redner, S. Voter models on heterogeneous networks. *Physical Review E*, 77(4):041121, 2008.

Strandburg-Peshkin, A., Papageorgiou, D., Crofoot, M. C., and Farine, D. R. Inferring influence and leadership in moving animal groups. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1746):20170006, 2018. doi: 10.1098/rstb.2017.0006.

Stueckle, S. and Zinner, D. To follow or not to follow: decision making and leadership during the morning departure in chacma baboons. *Animal Behaviour*, 75(6):1995–2004, 2008. ISSN 0003-3472. doi: https://doi.org/10.1016/j.anbehav.2007.12.012.

Titsias, M. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 5:567–574, 16–18 Apr 2009. URL `http://proceedings.mlr.press/v5/titsias09a.html`.

Tokuyama, N. and Furuichi, T. Leadership of old females in collective departures in wild bonobos (Pan paniscus) at Wamba. *Behavioral Ecology and Sociobiology*, 71, 02 2017. doi: 10.1007/s00265-017-2277-5.

Toner, J. and Tu, Y. Flocks, herds, and schools: A quantitative theory of flocking. *Physical review E*, 58(4):4828, 1998.

Torney, C., Morales, J., and Husmeier, D. A hierarchical machine learning framework for the analysis of large scale animal movement data. *Movement Ecology*, 9, 02 2021. doi: 10.1186/s40462-021-00242-0.

Torney, C. J., Levin, S. A., and Couzin, I. D. Decision accuracy and the role of spatial interaction in opinion dynamics. *Journal of Statistical Physics*, 151(1):203–217, 2013.

Turchin, P. *Quantitative analysis of movement : measuring and modeling population redistribution in animals and plants*. Sinauer Associates, 1998. URL `https://cir.nii.ac.jp/crid/1130282271101830912`.

Uhlenbeck, G. E. and Ornstein, L. S. On the Theory of the Brownian Motion. *Phys. Rev.*, 36: 823–841, Sep 1930. doi: 10.1103/PhysRev.36.823.

Vicsek, T. and Zafeiris, A. Collective motion. *Physics reports*, 517(3-4):71–140, 2012.

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. Novel Type of Phase Transition in a System of Self-Driven Particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

Wei, G. C. G. and Tanner, M. A. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

Weiss, G. H. *Aspects and applications of the random walk*. 1994.

Wilkinson, R. Accelerating ABC methods using Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1015–1023, 2014.

Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.

Yates, C. A., Erban, R., Escudero, C., Couzin, I. D., Buhl, J., Kevrekidis, I. G., Maini, P. K., and Sumpter, D. J. T. Inherent noise can facilitate coherence in collective swarm motion. *PNAS*, 106(14):5464–5469, 2009. doi: 10.1073/pnas.0811195106.

Young, G. A. *Essentials of Statistical Inference*. Cambridge University Press, 2005. doi: 10.1017/CBO9780511755392.

Zucchini, W., MacDonald, I., and Langrock, R. *Hidden Markov Models for Time Series: An Introduction Using R, First Edition*. CRC Press, 2009.