Bayesian nonparametric methods for dynamics identification and segmentation for powered prosthesis control

Neil Dhir

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

The nation which insist on drawing a broad line of demarcation between the fighting man and the thinking man is liable to find its fighting done by fools and its thinking done by cowards.

—Sir William Francis Butler

Abstract

Robots need to be able to adapt to their surroundings. Robots whose core function relates to the rehabilitation and assistance of humans, need to be able to adapt to humans. But not all humans, one human in particular: their user. This is an adaptive control problem which speaks of the need for powered prostheses to have anthropomorphic adaptive capabilities. However, it is inconceivable that *all* possible movements, dynamics and tasks can be preprogrammed into such a system. Prostheses, like robots, need to be able to learn and improve, either by themselves unsupervised, or with the help of human supervision.

The United States alone is home to more than two million amputees, where below-knee amputations is the most common form. The same country is also home to 4.7 million stroke survivors, many of whom could make use of rehabilitation robots. Vascular diseases, such as diabetes, cause 54% of all amputations and the number of diabetes and pre-diabetes cases in the United States currently exceeds 100 million and growing.

Whilst this thesis does not attack the root causes, it does present a three pronged approach for framing the adaptive control problem required for effective rehabilitation. We develop methodology that structures the problem into one of *learning* a control framework for lower extremity active prosthetics. First, we consider *incidence detection* as a key element of any control strategy. We study the dynamics of falling under the aegis of classification and state-space modelling. We use the standard Kalman smoother to inpaint missing observations. Then dimensionally reduce the feature space to demonstrate increased classification accuracy, compared with our reference study.

Secondly, we move to the *temporal segmentation* of similar time-series. This is done using the Bayesian nonparametric paradigm within the framework of state-space modelling and probabilistic programming. Models are rarely correct for real world data. Rather than comparing models that vary in complexity, this approach fits a single model that can adapt its complexity to the observations. This unbounded analysis of the state-space is required given the need for the proposed control framework to grow with new observations.

Finally, we investigate multiple incidents of walking velocity, and propose a learned control strategy able to smoothly transition between the incidents. We combine Gaussian process regression with impedance control, and gait-cycle regression to form a *locomotion envelope*. The learned control capabilities allow the wearer to smoothly transition between self-selected velocities.

Acknowledgements

The thing about acknowledgements is that either you compose them to include every man and his dog, so that anyone who has had even the faintest of impact on your work, will be immortalised in these pages. Conversely of course, there is the 'fuzzy-cluster' approach, where one can thank select, but broad groups of people, so that anyone who might have been left out, can claim membership of a nearby neighbour (not unlike the nearest neighbour algorithm in fact), and so also the glory that comes with being immortalised in these pages. The reader will be sad (or happy) to hear that yours truly will be opting for the former approach. Some will disagree with this course of action; indeed, to echo the words of Prof. Posner, this approach is "way(!) too elaborate. Less Neil".

⋆⋆⋆ [*Dinkus*[1]]

The first and most heartfelt thanks must go to my supervisors: Prof. Frank Wood, Prof. Ingmar Posner and Prof. Michael Osborne. Despite being brimful of sometimes rather silly ideas and projects, not once was I discouraged, rather they imposed a timetable and made me follow it and simply said 'do this and *then* do that'. This may come as shock to most, but I am used to doing 58+ things at once, being told to do tasks in a sequential manner was nothing short of a revolution for yours truly. For that and everything else, I owe them a debt of gratitude.

Granted, I would not be where I am today either, were it not for my parents. One would be hard pressed to find two people so utterly dedicated to the welfare of their children, not once skipping a beat in trying to make my life, and that of my siblings' as happy as can be – despite all the missed calls and ignored emails, they never lost faith. Though the inspiration to pursue a Ph.D. came from my father, the subject matter originates with my mother and her Churchillian approach to life. Her inherent kindness and selflessness will serve a guiding paragon for me long after this thesis has been deposited in the dusty vaults of the Bodleian. Thank you. To my sisters Emma and Sonja I can but express my most heartfelt appreciation; the sarcasm you taught me and the Christmas trees that you consistently ruin with your hideous taste (of course, Christmas would not be the same without it).

[1] Three stars used in this manner is a typographical symbol used to indicate a section break in writing, and is known as a 'dinkus'. It is used extensively in this thesis.

Then of course there is Claudia. A peculiar little woman if there ever was one. Fiercely independent, astonishingly clever and very loving. These past few years would not have been same without her, in thick and in thin, she was there, always coaxing you on to go further and try new things, and leave that comfort zone.

Further, my sincerest thanks must go to my friends in all the labs that I have been fortunate to be part of including the Wood group and the A2I group. In the former, Andy, with his bushy beard, kindness and generosity, never failed to offer a coffee nor cheer me up. Brooks, Jan-Willem and Tom – thick as thieves they were, and as clever as they come, any question on *anything* in machine learning (and other topics too it transpired) never stumped them. Yura, the kindest and perhaps one of the most intelligent people I have the privilege of calling my friend. You will go far. We were a small but valiant bunch, and Tuan Ahn did his utmost to control our divergence from sanity, in the late nights and mornings, prior to a paper deadline. It did not always work, but mostly it did. If I am certain of anything it is that one day Stefan will (a) become a billionaire or (b) get a billionaire to pay for his quite frankly outrageous, but brilliant, ideas.

In the A2I there is no shortage of people to thank (mainly because there are so many people in the Oxford Robotics Institute), but some include: Julie, Corina, Marcus, Geoff, Jeff, Matt and Paul. I would also like to extend my greatest gratitude to Adam for being so gracious in the face of the utter nonsense that I would sometimes profess to be the key to artificial intelligence, and proceed to derive things that quite frankly made no sense at all. Naturally an acknowledgement would be amiss if not Matthijs was in it too; thank you for joining in the lion work. I enjoyed our late nights at Wolfson and Hertford, seemingly working against the clock a lot of the time. Not to mention our 'gym' workouts which were anything but. Matthijs, Yura and Adam are all rather alike after all; hideously intelligent and the most excellent of friends.

Naturally I, like most graduate students, have taken time away from Oxford, and done a spot of collaboration elsewhere. Hence, first I would like to thank Dr. Matthew Howard at the Centre for Robotics Research at King's College London. I spent a summer doing very peculiar work on integrating sensors into jogging leggings, and then operating upon them our time-series models. It was the first time in ten years I used thread and needle, but nonetheless a healthy break from `vim`. In the same city I was also hosted for a few weeks by Dr. Aldo Faisal of Imperial College London. Thanks must go to him for allowing me to join Imperial's Cybathlon team as an external member. Further, I would like to extend my thanks to Dr. Houman Dallali and Dr. Mo Rastgaar for our fruitful collaboration over the past year. I will remember with fondness my frantic ways of trying to ascertain, from across the Atlantic, whether or not Houman was dead or alive, 10 minutes prior to the ICRA deadline (he was, after all, just getting a sandwich). I would also like to thank my CDT and especially Jo for doing so much administrative work on my behalf, and never once complaining about it, simply asking for the next form I wanted signed. An absolute trooper.

Then there are all my friends outside of academia in Oxford which have provided me with a truly spectacular social life, where any thesis trouble I may have had disappeared in an

instance with its interaction. First I must thank Wolfson College for this, and also for providing me with funding to various conferences around the world, and also for lodging me for two years. An incredible place in short. Then naturally the boat club follows, wherein I met some of my closest confidantes, as captain, secretary, rower and coach. In every trail of that life, new friends and relationships were formed. And now regrettably I too have reached the 'fuzzy-cluster' acknowledgement, though I shall at least try. To all my boats throughout the years, but a special thanks to Tom, Jack, Peter, Big Josh, Big Boy, Jasper, Gido, Stef, Oscar, PK (for teaching me how to row), Simpson, Nanda, Jimbo and Lucian. Mad men and women the lot of them, but completely irreplaceable. Least but not last there is the Amazonian warrior Lea, descended from the East, with her equally towering kinsman Atamlu – a true class act if there ever was one.

Finally, my home from home; 26 Western Road. Even in the darkest hour, when things simply did not work, and focus had to be met to see the light, *they* were always there, the house. There is considerable warmth in coming home, every day, to a hot meal and a table full of friendly faces, discussing everything under the sun. It seems as if the house never had less than six people living there, and sometimes it had 12, some of whom I had not yet had the pleasure of making their acquaintance. Irrespective, to all to whom I had the honour of calling 'housemate' – thank you. It was an absolute blast, even when someone ripped of our door.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In the last few decades there have been immense strides in both robotics and computer science, which have led to the development of numerous wearable robotic devices such as exoskeletons, powered prostheses as well as powered orthoses. These devices have appeared both for industrial and medical applications – for the latter case, mainly for the rehabilitation and restoration of locomotion (Tucker et al., 2015). Though there are many successes, challenges remain, particularly with respect to control, where strategies are required to deal with the formidable task of sharing control load between the human user (intent) and the machine (effect). Most of today's control architecture rely on analytical approaches and pre-calculated joint-trajectories. Though having claimed many achievements, they have strong model dependency and are often vulnerably to disturbances and noise. Contrast this with human locomotion, which is elegant and robust, and compared to other forms of locomotion, highly energy efficient. It is a form of dynamics that has considerable richness to its manifestation, a richness that grows and adapts with the morphology and anatomy of its progenitor – not to mention her operating environment. Therein lies the problem: the modern world was designed for people, not robots.

Humans are skilled at quickly understanding and adapting to the world around them. From scene understanding to the learning of new skills, humans excel. This unique transfer learning capability is most likely conditional on extensive prior knowledge (Battaglia et al., 2013) and innate knowledge, which we can draw upon where and when we need it. Given that this is the case for humans, why not so too for machines? Our goal is to create the building blocks required for an adaptive control framework, that ultimately leads to anthropomimetic devices for human rehabilitation. We posit that by equipping a system

with that prior knowledge, will lend them the necessary tools to learn from interactions and measurements of unknown environments.

In this thesis we introduce application driven methodology, manifested through the study of human-centred complex time-series, for ultimate application to adaptive powered prostheses. Times-series observations are measurements taken over time, following the temporal evolution of an experiment, a person, a storm or any other natural or unnatural behaviour under investigation. Data, especially nowadays, is ubiquitous and society is producing enormous amounts of it. We are fast reaching a point where our analysis of these observations needs to become ever more automatic. This is not only to lessen the human workload, but also to allow us to uncover the latent dynamics within, to better understand the world around us. And, in some cases, enable us to control, predict and learn from those those measurements.

1.1 Motivation

The case for this study is thus pervasive. Patterns in time-series are typically manifested through the correlations induced by the temporal structure of the observations. Consequently we seek to build models that can manipulate and extract that structure. But in the absence of truth we can but build models that are an idealised view about the world, and from where observations originated. Despite this absence, and idealisations, models are useful because they allow us to make informed decisions about the future as well as analyse the past. Indeed, as this thesis was being composed, and at the locations it was composed, data was being recorded, from the GPS sensors and the Pitot tubes on the aircraft, to the WiFi signals quietly working away in the background noise of the crowded café. Not to mention the more obvious cases such as stock market returns, weather station recordings, Facebook 'likes' or click-rates for websites – all are examples of time-series. However, the sort of observations that will be primarily under investigation in this thesis are those generated by us humans.

The human body is, for all intents and purposes a grey box (rather than a black box, since we have *some* understanding). Neuroscience, biology, psychology and other related fields, have not yet reached a stage where they treat our function as a deterministic system; where at every point in time there is mapping between state intent and resulting action. Rather,

how our brain functions is slowly being uncovered, bit by bit, and in so doing time-series observations are recorded as a measure of the causality that we assume is inherent in our control. Those observations, like those recorded from the stock market, are complex, often high-dimensional and usually operate in a highly complex state-space. That is without considering the duration properties of the phenomena under investigation, let alone the marked difference in time-series observation, generated by different individuals – despite originating from the same 'type' of black-box (the human body).

Consequently, we are faced with a difficult analysis regime. Not only are the observations inherently elaborate, they are also dynamically inconsistent depending on the sampling source. Though broadly speaking they correspond to some distribution, on a smaller scale these observation sets are harder to deal with. Thus, from an analysis point-of-view it is clear that ever more complex models are required. Models that can grow both in size as well as complexity, as the observations they admit become ever richer and contain patterns that may be interpreted as noise by simpler models. There is thus a strong argument for explicitly modelling the uncertainty that we have about the observations. One way of doing this is by adopting the Bayesian nonparametric paradigm.

But we seek to do more than segmentation and temporal classification of time-series. We are also interested in their very nature, the latent patterns and how synthesising them can lead to effective control mechanisms. Taken together, the themes that we investigate have, in common, their operation on highly nonlinear time-series observations. Typically these are extracted from difficult and highly dynamic natural phenomena such as human movement. Whilst the final application of this work concerns the useful application of Bayesian nonparametric techniques for human use, the scope of this thesis is not restricted to analysis of such domain-specific observations. Instead, as we shall show, a number of nonlinear natural phenomena, will be used to derive, evaluate and demonstrate, new techniques for usage not just within our chosen domain. This all being said, this study can be construed as one of *understanding* (features), *segmenting* (classifying) and *synthesising* (control) – human locomotion.

1.2 Problem statement

Walking and generally using our legs for locomotion is something most of us take for granted. It requires the activation of the leg muscles to appropriately control the ground reaction forces, whilst also modulating the mechanical impedance of the leg – particularly at the ankle (Ficanha et al., 2016). This is not a trivial set of dynamics to understand let alone to replicate. In some instances though, replication is necessary because of an amputation. In the United States alone, there are nearly two million amputees (Ficanha et al., 2016). Put differently one in nearly 200 Americans has experienced some sort of amputation (Quintero et al., 2011). The reasons for amputations are multiple but, generally speaking, they fit into one of two categories (Ficanha et al., 2016). The great majority suffer from vascular diseases (54%) (including, e.g diabetes and arterial diseases) where the rest, broadly, have suffered from some major trauma such as severe accidents or war.

To deal with this patients are usually fitted with prostheses, most of which are still energetically passive, the function of which is to replace the biomechanical functionality of the absent limb (and as closely as possible mimic its operation) (Lawson et al., 2014). They should thus ideally have anthropomorphic characteristics. But whilst modern mobility devices play an ever increasing role in creating movement and improving mobility for people who have lost it due to disease, accident or war. In a not too distant future, rehabilitation robots (powered; exoskeletons, prostheses and orthoses) will be a natural part of daily life in healthcare and for those with disabilities, providing assistance in areas ranging from clinical applications to caregiving. However, it is not conceivable that all possible movements and tasks can be preprogrammed into such systems. Prostheses, like robots, need to be able to learn and improve, either by themselves or with the help of human supervision. Since their benefit is for human rehabilitation, they need to function like humans. It is likely that humans rely upon a control strategy which is adaptable, can become more robust and accurate with more data and provides a nonparametric approach which allows the strategy to grow with the number of observations. This is the type of strategy we need to endow upon modern rehabilitation devices.

To enable this study, we will make use of three major themes, alluded to at the bottom of §1.1. First in chapter 3 we will investigate representative features of human locomotion

by virtue of incidence detection. In particular, we consider the dynamics of falling as an example case of motion that a prosthesis should be able to recognise from time-series observations. Following on from that in chapter 4, we investigate how we can automatically extract such dynamical events from time-series measurements by employing Bayesian nonparametric state-space models. Finally, in chapter 5, multiple incidents of a similar type of dynamical event (forward motion including walking and running) are used to create a novel control scheme that uses Gaussian processes and impedance control to robustly let a user transition between observed incidents, as well as new ones, by virtue of Gaussian process regression.

1.3 Detailed contributions

A detailed breakdown of this thesis' main contributions, are presented in the following sections.

1.3.1 Representative latent structure of human locomotion

In chapter 3, we take a deeper look at a popular study by Luštrek & Kaluža (2009). Therein the authors deal with the classification of human motion-capture data. But they do so through the utility of hand-crafted feature-sets, which are high-dimensional. We demonstrate, by using simple and off-the-shelf methods, increased classification accuracy across the board for all classifiers in their study.

Further, in the initial study, a small part of the observations were missing due to data-processing issues (Luštrek & Kaluža, 2009). By employing the common Kalman smoother (Kalman, 1960), we show that it is possible, with high accuracy, to in-paint the missing observations, having first trained a linear-Gaussian state-space model on the available (high-dimensional) observations. Certainly, this has the effect of supplying the observation space with additional members, but the increased classification performance was rather due to the dimensionality reduction that we imposed on the observations.

The popularity of the reference study is down to its high accuracy classification of a highly dynamic human motion sequence: falling. But by using dimensionality reduction we show that this classification can be higher still, by compressing the dimensions used in feature construction, as they do not contain any information.

Finally, this work (Dhir & Wood, 2014) should function as a motivation for why nonparametric methods are to be preferred. Particularly in domains where the classification task is dynamic, i.e. the number of classes grows or shrinks, with the size of the observations. Having to impose on the model a fixed size is unrealistic for many real-life applications, particularly in those where we are, a priori, uncertain of the size of the classification space. One such example is novelty detection, which we consider in chapter 4.

1.3.2 Bayesian nonparametric state-space models

Bayesian modelling is a natural fit for many time-series modelling problems. It is not uncommon that we have certain ideas regarding the origin of the measurement, ideas which we can impose on the analysis framework in a Bayesian fashion. Typically though our prior knowledge concerns what we expect to see from the measurement. That is to say that we have certain preconceptions regarding the *state* of the time-series – or what segment corresponds to which dynamical behaviour using human activity recognition as a guiding reference.

The state corresponds to the statistical properties of the observations during that segment, and the generative modelling approach seeks to specify a model which mimics our ideas regarding the origin of those observations. In this sense, the hierarchical modelling paradigm becomes pervasive as we are then faced with the idea that there is some higher generative oracle responsible for generating the measured phenomena. Consequently this provides us with a very rich and flexible model class, particularly when combined with the Bayesian paradigm.

Discrete state-space models that are based on these ideas are represented by extensions to the hidden Markov model (Rabiner, 1989). One of the most powerful was the introduction of the Bayesian nonparametric hierarchical Dirichlet process (HDP) HMM (Teh et al., 2006), which allowed for an unbounded state-cardinality on part of the model. This has the effect of allowing the complexity of the model to grow with the size of observations. This model serves as the foundation for our contributions, discussed in chapter 4. Where we are interested in inference over categorical switching variables, which tell us when dynamics of our system switch mode.

The HDP-HMM has a number of drawbacks, the primary of which is that it is not state-persistent and favours fast switching dynamics, caused by having a geometric duration distribution. This was addressed by Fox et al. (2008) who introduced a more parametrised model that yielded some control over the switching behaviour. We extend this model in (Dhir et al., 2016b,a) by making the model parameters *stateful* i.e. we index them by the state itself. In (Dhir et al., 2017c) we introduced the infinite duration (ID) HMM and the stateful IDHMM, which gives a nonparametric treatment to the duration distribution. This is a more flexible model than Johnson & Willsky (2010)'s HDP hidden semi-Markov model (HDP-HSMM), because it allows for a nonparametric duration distribution whilst the HDP-HSMM has a parametric form. For all models we employed automatic inference methods via the probabilistic programming paradigm manifested through *Anglican* (Wood et al., 2014).

1.3.3 Gaussian process regression for rehabilitation robot control

We continue in the Bayesian nonparametric vein in chapter 5 where we move away, to some degree, from modelling abstractions and focus on transition dynamics in powered prostheses. This is a compelling area of research due to the sheer number of individuals who are partially disabled. The United States alone is home to almost two million amputees, and amongst these there are six hundred thousand people with below-the-knee amputation, who are in need of suitable ankle-foot prostheses, to regain an agile and stable gait. There are a further 4.7 million stroke survivors, with another 700,000 added each year many of whom could make use of a powered ankle-foot prosthesis (Holgate et al., 2008).

In (Dhir et al., 2017a, 2018) we show that by analysing and regressing over time-series observations, specifically motion-capture data, we can effect robust and adaptive transition control for powered ankle-prostheses. Current powered prostheses have problems adapting across velocities as most control schemes only specify one or a discrete number of locomotion velocities. Conversely it is likely that humans rely upon a control strategy, which is adaptable, can become more robust and accurate with more data and provides a nonparametric approach, which allows the strategy to grow with the number of observations. We demonstrate such a method in this study and successfully simulate locomotion

well beyond our training data. The method we propose is based on common physical features observed in numerous human subjects walking at different speeds. Based on the derived *locomotion envelopes* we demonstrate that ankle power increases monotonically with speed among all subjects. Using this idea we demonstrate our methods in simulation and human experiments, using a powered ankle-foot prosthesis to show the effectiveness of the method.

By using Gaussian process regression we are able to find the locomotion variate manifolds, which jointly express the evolution of the parameters as velocity increases or decreases, over one gait cycle. By bounding the range over which we perform velocity regression and using the average of multiple gait-cycles (Moore et al., 2015), we can combine any required number of gait-cycles to effect robust locomotion, which is able to transition between velocities using impedance control. Gaussian processes are a good fit for the methods we propose. This is due to (1) given observations and a kernel, the posterior predictive distribution can be found exactly in closed form (Rasmussen & Williams, 2006) (*with* a Gaussian likelihood that is, we do not consider non-Gaussian likelihoods), (2) by nature of its construction, expressivity is considerable, allowing us to incorporate a host of modelling assumptions and domain knowledge. Finally, as noted by Rasmussen & Williams (2006); given a fixed kernel, the Gaussian process posterior allows us to integrate exactly over hypotheses. This means that by using probabilistic inference, we can take a group of hypotheses (or models), and weights those hypotheses based on how well their predictions match our observations (Duvenaud, 2014), and by doing so, overfitting becomes less of an issue than in comparable model classes. This makes them ideal for dealing with complex human-centred time-series observations.

1.4 Contributions

This thesis rests upon a foundation of joint-authored peer-reviewed contributions as well as some unpublished material. The former is listed below with the contributions noted of all the authors involved, for each project. The work was funded by the Centre for Doctoral Training in Healthcare Innovation, sponsored by the Engineering and Physical Research Council (EPSRC).

- Dhir, N. and Wood, F. Improved activity recognition via Kalman smoothing and multiclass linear discriminant analysis. In *Proceedings of the Engineering in Medicine and Biology Society (EMBC), IEEE*, pp. 582–585. IEEE, 2014

  N.D. was responsible for all the modelling and implementations, experiments as well as formulating the problem. F.W. provided state-space model advice and general theory advice.

- Dhir, N., Perov, Y., Wijers, M., Wood, F., Markham, A., Trethowan, P., du Preez, B., Loveridge, A., and Macdonald, D. Tracking african lions with nonparametric hierarchical models using probabilistic programming. In *Proceedings of the International Society of Bayesian Analysis (ISBA)*, 2016a

  N.D. was responsible for conceptualising the model and theory, and provided the experimental results. Y.P. provided the majority of the Anglican implementations, jointly N.D. and Y.P. extended the initial model proposed by N.D. Moreover M.W. and A.M. provided advice and data from real-world experiments. F.W. provided advice on general theory.

- Dhir, N., Perov, Y., and Wood, F. Nonparametric Bayesian models for unsupervised activity recognition and tracking. In *Intelligent Robots and Systems (IROS), IEEE/RSJ*, pp. 4040–4045. IEEE, 2016b

  N.D. and Y.P. jointly extended the initial models, N.D. provided the modelling, implementations and experimental results, as well as conceptualising the problem.

- Dhir, N., Vakar, M., Markham, A. C., Wijers, M., Wood, F., Trethowan, P., Du Preez, B., Loveridge, A., and Macdonald, D. Interpreting lion behaviour with nonparametric probabilistic programs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017c

  N.D. supplied the novel models and jointly developed the theory with M.W. Further, M.W. (primarily) and N.D. provided model implementations whereas N.D. designed the experimental pipeline and conceptualised the problem further. N.D. was further responsible for implementing the variational autoencoder framework used for dimensionality reduction and feature learning. A.M and M.J. provided experimental data and general problem domain advice. F.W. provided general theory and implementation advice.

➢ Dhir, N., Dallali, H., and Rastgaar, M. Coregionalised Locomotion Envelopes. In *Proceedings of the Adaptive Control Methods in Assistive Technologies Workshop, IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, 2017a

N.D. conceptualised the idea of using coregionalised Gaussian processes (GP) for prosthesis control (a special thanks to Michael Osborne for advising on theory and pointing to resources). H.D. supplied motivation and problem formulation. M.R. provided general theory advice.

➢ Dhir, N., Dallali, H., Ficanha, E. M., Ribeiro, G. A., and Rastgaar, M. Locomotion Envelopes for Adaptive Control of Powered Ankle Prostheses. In *Robotics and Automation (ICRA)*. IEEE, 2018

N.D. provided the concept and theory for using Gaussian process regression for adaptive prosthesis control, and was responsible for GP modelling, implementation and experiments. H.D. was responsible for simulation and hardware experiments, and proposed several extension to the initial concept. E.F. and G.R. contributed the control implementation and design. M.R. provided general theory advice.

1.5 Thesis outline

➢ In chapter 2 we relate some necessary preliminary material required for informed reading of this thesis. Each chapter refers back to chapter 2 and the passages therein, should not be read as a contiguous piece of text but referred to when called upon in the main chapters. Certain topics have been left out such as probability theory, measure theory and graphical models, to mention a few. It is assumed that the reader has knowledge of these topics. If not however, literary references have been provided throughout to aid understanding. Standard material which has not been covered, can generally be found in one off (Bishop, 2006; Rasmussen & Williams, 2006; Murphy, 2012; MacKay, 2002) or (Barber, 2012).

➢ In chapter 3 we cover the first part of our main contributions, by considering incidence detection in high-dimensional time-series observations. We employ standard classification methodology such as random forests, where background is provided in chapter 2, as well as common dimensionality reduction, also covered in the

preliminary sections. We use the Kalman smoother as our generative model, which we use for in-painting of missing observations, covered in §2.4. We demonstrate a basic method for classifying incidents, by studying in the more detail the highly dynamic event of falling.

➢ In chapter 4 we consider the problem of automatically (unsupervised) finding incidents in time-series observations. We derive new Bayesian nonparametric state-space models and employ probabilistic programming systems for inference. We also demonstrate the usage of Bayesian optimisation in empirical validation, for finding appropriate model parameters. We demonstrate our methods on synthetic observations as well as two real human activity recognition tasks. Where automatically segment activities. Further, we also demonstrate our methods on a novelty detection task, where we seek to model lion behaviour. Several dependencies for this chapter are found in chapter 2.

➢ In chapter 5 we demonstrate a control schema combining Gaussian process regression and impedance control. We demonstrate that by regressing across multiple locomotion parameters (such as the ankle plantarflexion angle) it is possible to illicit a control schema, which allows smooth transitioning between observed and inferred walking velocities. We test this concept in simulation and in hardware experiments. Here an incidence corresponds to a different walking velocities. Conditional material is provided in §2.7.1.

➢ In chapter 6 we conclude. We discuss the building blocks developed in this thesis, and consider future work. In particular we note further steps, how these methods could be combined to form a framework which allows for adaptive control of powered prostheses.

CHAPTER 2

Preliminaries

Contents

This thesis will combine a host of topics covering not just machine learning, but also signal processing, as well as some zoology. To this end, we will require the reader to have at least some familiarity with the sub-topics into which we shall venture. The basis for

those topics will be found in this chapter, and we invite the reader to refer back here (as we shall be doing throughout the text) when clarity is required, or further explanation warranted. That being said, this is *primarily* a thesis which investigates novel forms of machine learning applications and methods.

Machine learning can be employed to detect patterns in observations (we shall be using 'observations' and 'data' interchangeably throughout), and the uncovered patterns used to perform decision making under uncertainty, as well as a host of other applications. But the patterns in themselves may not have easily interpretable governing dynamics, especially not when the observations are manifested as noisy, high-dimensional, time-series – the primary structure under investigation. Complex non-linear dynamics arise in many fields of science and engineering, but uncovering the underlying governing dynamics directly from observation poses a challenging task. The ability to symbolically model complex networked systems is key to understanding them, and an open problem in many disciplines. It is in this domain that this proposed work is set.

2.1 Dimensionality reduction

It is often the case that, when analysing high dimensional observations, it is useful to reduce the dimensionality by projecting the data to a lower dimensional subspace, which captures the principal information contained in the observations. Indeed, as noted by Murphy (2012) what we are really interested in can be quantified as "latent factors", and they represent the key variabilities in our observations (in the case of principal component analysis for example). The principal information can be ascertained in a multitude of ways, linear and nonlinear, and herein we will compose a brief synopsis of the dimensionality reduction methods employed and investigated in this thesis.

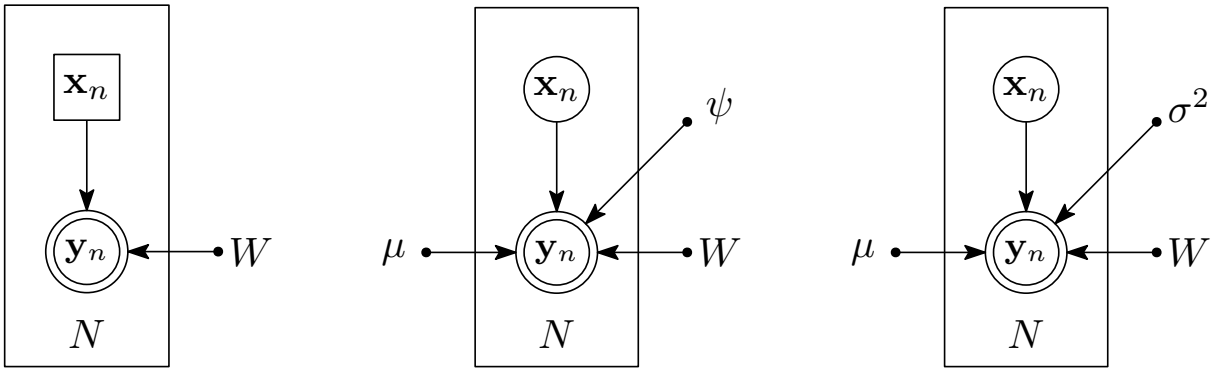
As usual we operate with latent and observed variables. Observed variables are typically multivariate i.e. $\mathbf{y} \triangleq [y_1, \ldots, y_D]$ where $\{y_i \in \mathbb{R} \mid i = 1, \ldots, D\}$ and stored in a design matrix $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\mathsf{T}$ such that $\mathbf{Y} \in \mathbb{R}^{N \times D}$. The same same specification holds for latent variables i.e. $\mathbf{x} \triangleq [x_1, \ldots, x_d]$ where $\{x_i \in \mathbb{R} \mid i = 1, \ldots, d\}$ and stored in a design matrix $\mathbf{X} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\mathsf{T}$. Unless otherwise specified, we operate on continuous spaces.

2.1.1 Linear methods

In this section we provide a brief overview of some commonplace linear dimensionality reduction schemes, used and investigated in this thesis. We provide a synopsis of each method in turn. Though we use principal component analysis (PCA) we will not review it, as we assume the reader is familiar with the details of this method.

2.1.1.1 Factor analysis

Bishop (2006) explains that factor analysis (FA) is a linear-Gaussian latent variable model that is closely related to probabilistic PCA (PPCA) – see fig. 2.1. We assume that the observations, the feature vectors, are caused by a linear transformation of lower dimensional latent factors and added Gaussian noise, without loss of generality the factors are distributed according to a Gaussian with zero mean and unit covariance. The noise is also zero mean and has an arbitrary diagonal covariance matrix (Pedregosa et al., 2011). Moreover, if we restrict the model further, by assuming that the Gaussian noise is evenly isotropic (all diagonal entries are the same) we obtain PPCA.



(a) Principal component analysis.

(b) Factor analysis.

(c) Probabilistic principal component analysis.

Figure 2.1: Graphical models in plate notation, of some common latent variable model (also known as latent factor models). Note the difference between fig. 2.1b and fig. 2.1c stems only from the difference in the prior

.

Our interest is in finding a lower dimensional probabilistic description of $\mathbf{y}$. From Barber (2012) and Bishop (2006), we give a brief description of the FA model. The conditional distribution of $\mathbf{y}$ given the latent variable $\mathbf{x}$ is taken to have a diagonal isotropic covariance

$$\mathbb{P}(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid \mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \boldsymbol{\psi}) \tag{2.1}$$

where the columns of $\mathbf{W}$ are the factor loadings and $\boldsymbol{\mu}$ is the constant bias which sets the origin of the coordinate system. The complete model is given by:

$$\mathbb{P}(\mathbf{y}) = \int \mathbb{P}(\mathbf{y} \mid \mathbf{x})\, \mathbb{P}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathsf{T}} + \boldsymbol{\psi}), \tag{2.2}$$

where we can determine the parameters $\{\mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\psi}\}$ of the model via maximum likelihood estimation. For comparison, the analogue PCA model is given by:

$$\mathbb{P}(\mathbf{y}) = \int \mathbb{P}(\mathbf{y} \mid \mathbf{x})\, \mathbb{P}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathsf{T}} + \sigma^2\mathbf{I}), \tag{2.3}$$

where the noise has covariance $\sigma^2\mathbf{I}$. Consequently in the FA model the covariance is given instead by a $D \times D$ dimensional matrix $\boldsymbol{\psi}$.

2.1.1.2 Truncated singular value decomposition

TSVD performs linear dimensionality reduction by means of truncated singular value decomposition. It is very similar to PCA, but operates on sample vectors directly, instead of on a covariance matrix. This means it can work with sparse matrices efficiently (Pedregosa et al., 2011). Assuming that all our feature vectors are contained in the matrix $\mathbf{Y}$, then there exists a full-rank decomposition

$$\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}} \tag{2.4}$$

where the columns of $\mathbf{U}$ and $\mathbf{V}$ are orthonormal and the matrix $\mathbf{S}$ is diagonal with positive real entries (the singular values). We can find a reduced-rank approximation (or truncated SVD) to $\mathbf{Y}$ by setting all but the first i largest singular values equal to zero and using only the first i columns of $\mathbf{U}$ and $\mathbf{V}$, whereupon we use the TSVD approximation to $\mathbf{Y}$ to build our new feature vectors.

2.1.1.3 Gaussian random projection

Random projections involve taking a high-dimensional data set and then linearly mapping it into a lower-dimensional space, while providing some guarantees on the approximate preservation of distance (Menon, 2007). Hence, using random projections, the original D-dimensional data is projected onto a d-dimensional subspace ($d \ll D$) using a random matrix $\mathbf{R} \in \mathbb{R}^{d \times D}$, such that

$$\mathbf{B} = \mathbf{R}\mathbf{X} \tag{2.5}$$

where $\mathbf{B} \in \mathbb{R}^{d \times N}$ and $\mathbf{X} \in \mathbb{R}^{D \times N}$ (Bingham & Mannila, 2001). The power of random mappings arise from the Johnson-Lindenstrauß lemma, which states that if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved (Bingham & Mannila, 2001; Menon, 2007; Pedregosa et al., 2011). Hence, the lemma deals with Euclidean-distance preserving embeddings. Now, $\mathbf{R}$ is generally not orthogonal, but is in fact a linear mapping, which if not orthogonal causes distortions in $\mathbf{B}$. Orthogonalising $\mathbf{R}$ is computationally expensive. But, as it turns out, we do not have to perform this operation but can instead rely on the Hecht-Nielsen theorem (Hecht-Nielsen, 1994), which says that as we go into higher-dimensional space, the number of nearly-orthogonal vectors increases. Thus, vectors having random directions might be sufficiently close to orthogonal, and thus $\mathbf{R}\mathbf{R}^{\mathsf{T}}$ will approximate an identity matrix (Bingham & Mannila, 2001).

The choice of projection matrix $\mathbf{R}$ is naturally important, such that the random projections do indeed have distance-preserving properties. Generating an orthogonal subspace is not easy, but as the Hecht-Nielsen theorem shows, we do not need to consider strictly orthogonal subspaces. Hence, we choose to draw each projection matrix entry, from a zero mean, independent and identically distributed Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{1})$ (Menon, 2007; Pedregosa et al., 2011). The advantage of this is that we get spherical symmetry among the random vectors. Even though our subspace is not strictly orthogonal, the spherical symmetry is enough to ensure that it comes close enough to this property anyway (Menon, 2007).

2.1.1.4 Partial least squares regression

PLSR is a technique used with data, which contains correlated predictor variables; it can be thought of as a form of supervised PCA (Murphy, 2012) (or a cross between multiple linear regression (Neter et al., 1996) and PCA). PLSR constructs new predictor variables as linear combinations of the original predictor variables. PLSR combines information about the variances of both the predictor variables and the responses, while also considering the correlations among them, whilst PCA decomposes the data (our feature vectors) $\mathbf{Y}$ in order to obtain components, which explain $\mathbf{Y}$ best (Abdi, 2010). By contrast PLSR finds components from $\mathbf{X}$ that predict our response values $\mathbf{Y}$ (the activity labels). Specifically PLRS searches for a set of latent vectors that perform a simultaneous decomposition of $\mathbf{X}$ and $\mathbf{Y}$ with the constraint that these components explain as much as possible of the

covariance between $\mathbf{X}$ and $\mathbf{Y}$ (Abdi, 2010). This is followed by a regression step where the latent vectors obtained from $\mathbf{X}$ are used to predict $\mathbf{Y}$.

Hence, PLSR decomposes both $\mathbf{X}$ and $\mathbf{Y}$ as a product of a common set of orthogonal factors and a set of specific loadings. Thus, the independent variables are decomposed as

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathsf{T}} \text{ where } \mathbf{T}^{\mathsf{T}}\mathbf{T} = \mathbf{I}, \tag{2.6}$$

where $\mathbf{I}$ is the identity matrix. Where we call $\mathbf{T}$ the score matrix and $\mathbf{P}$ the loading matrix. As such we estimate the response values as

$$\widehat{\mathbf{Y}} = \mathbf{T}\mathbf{B}\mathbf{C}^{\mathsf{T}}, \tag{2.7}$$

where $\mathbf{B}$ is a diagonal matrix with the regression weights on the diagonal, and $\mathbf{C}$ is the weight matrix of the dependent variables, and the columns of $\mathbf{T}$ are the latent vectors. When several response variables are to be estimated simultaneously we use the PLS2 algorithm (for details see (Manne, 1987)). We can use dimensionality reduction via PLSR by setting the number of components in the PLSR model where we choose components so that the majority of the variance in the response variables, is explained through our choice of latent vectors.

⋆ ⋆ ⋆

We now turn our attention to the more exotic and not quite-so-commonplace, nonlinear methods.

2.1.2 Variational autoencoders

Throughout this work multiple nonlinear methods have been scrutinised, analysed and tested. Some of that work has not been included in this thesis due to space constraints, brevity and wanting to maintain some coherent line of reasoning (and focus) throughout. Nonetheless the ones left out do at least bear mentioning: the Gaussian process latent variable model (and its many variations) (Lawrence, 2003), the Gaussian process dynamics model (Wang, 2005) and the conditional restricted Boltzmann machine (and its extensions) (Taylor et al., 2006).

One model, which is included and which we have spent considerable time studying, applying and deriving new versions of (not included), is the variational autoencoder by Kingma &

Welling (2013) and Rezende et al. (2014), which is reviewed herein. To understand the end-state variational autoencoder, it is worthwhile considering its precedents.

2.1.2.1 Learning representations using autoencoders

Traditional autoencoders (Bengio et al., 2009) are models designed to output a reconstruction of their input. Typically they consist of a deep neural network which transforms the (typically noisy) observed states $\mathbf{y} \in \mathbb{R}^d$ to a latent representation $\mathbf{x} \in \mathbb{R}^{d'}$ s.t. $d' \ll d$ (van Hoof et al., 2016). For graphical representations see fig. 2.2a and fig. 2.2b.

The *encoder* is usually of the form

$$\mathbf{x} = f_\theta(\mathbf{y}) = s(\mathbf{w}^\mathsf{T}\mathbf{y} + \mathbf{b}) \tag{2.8}$$

where $\theta \triangleq \{\mathbf{w}, \mathbf{b}\}$ and $s(\cdot)$ is a nonlinear transition function, such as $\tanh(\cdot)$. The *decoder* predictably takes a similar form

$$\tilde{\mathbf{y}} = g_{\theta'}(\mathbf{x}) = s(\mathbf{w}'^\mathsf{T}\mathbf{x} + \mathbf{b}') \tag{2.9}$$

where $\tilde{\mathbf{y}}$ is the reconstructed input and $\theta' \triangleq \{\mathbf{w}', \mathbf{b}'\}$ (van Hoof et al., 2016). Additionally, as before $\mathbf{y} \triangleq [y_1, \ldots, y_d]$ and $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\mathsf{T}$, where the same analogue holds for the latent and reconstructed space. The parameters of the autoencoder are trained using gradient descent on the reconstruction error, which is received if we use as our loss function $\mathcal{L}(\cdot)$, the mean squared error:

$$\theta^*, \theta'^* = \underset{\theta^*,\ \theta'^*}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left[\mathbf{y}^{(i)}, g_{\theta'}\left(f_\theta(\mathbf{y}^{(i)})\right)\right]. \tag{2.10}$$

The above relations constitute the most basic autoencoder. For all full list of its many extensions see (Goodfellow et al., 2016, §14), we shall mentioned briefly one other flavour: the denoising autoencoder (DAE) (Vincent et al., 2008).

The DAE is an autoencoder that receives a corrupted data point $\mathbf{y}_\bullet$ as input and is trained to predict the original, uncorrupted data point $\mathbf{y}$ as its output, meaning that gradient descent operates on the following reconstruction error:

$$\theta^*, \theta'^* = \underset{\theta^*,\ \theta'^*}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left[\mathbf{y}^{(i)}, g_{\theta'}\left(f_\theta(\mathbf{y}_\bullet^{(i)})\right)\right]. \tag{2.11}$$

Typically autoencoders were used for dimensionality reduction or feature learning (Goodfellow et al., 2016), but have recently been connected to latent variable models

(such as those mentioned at the start of this section). This in turn have put them at the heart of generative modelling, where variational autoencoders (VAE) efficiently infer the latent variables of probabilistic generative models (Kingma & Welling, 2013; van Hoof et al., 2016). To gain a complete understanding of this model, we shall also draw upon an review, results from other fields, which we will use to derive the model from scratch.

2.1.2.2 Variational inference

Without wanting to labour too much on the finer details of variational inference (VI), it is worth beginning with the core idea. In essence: *variational (Bayesian) methods allow us to re-write statistical inference problems as optimisation problems.* This enables us to advanced optimisation tools to solve statistical inference problems, and vice versa. Incisive detail is offered in (Bishop, 2006, §10), but we shall derive the fundamentals that will lead us to the VAE.

We are interested in $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$. This is an inference problem. Finding this posterior *exactly* is an NP-hard problem (MacKay, 2002) – we would rather avoid that. Thus we turn to the approximate inference family of which VI is a member (Bishop, 2006, §10.1), and under whose guise we instead seek (as the name suggests) a (good) approximation to $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$ (Jang, 2016). One such approximation is the mean-field approximation (Bishop, 2006; Jang, 2016)

2.1.2.3 Variational lower bound objective for mean-field approximation

We first consider Bayes' law to formulate precisely where the problem lies:

$$\mathbb{P}(\mathbf{x} \mid \mathbf{y}) = \frac{\mathbb{P}(\mathbf{y} \mid \mathbf{x})\,\mathbb{P}(\mathbf{x})}{\mathbb{P}(\mathbf{y})} \tag{2.12}$$

wherein the evidence is given by

$$\mathbb{P}(\mathbf{y}) = \int_{\mathbf{x}} \mathbb{P}(\mathbf{y}, \mathbf{x})\,\mathrm{d}\,\mathbf{x} \tag{2.13}$$

which is intractable due to the (most likely) high dimensionality of $\mathbf{x}$ and we also run into problems when we consider complex probability distributions over our space, specifically ones for which we do not know how to compute the posterior density in closed form. What we can do instead then is to inference on some nice and simple parametric distribution

$\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})$ where we *do* know how to do posterior inference. They key idea is to adjust the parameters ϕ of $\mathbb{Q}(\cdot \mid \cdot)$ so that it is a close as possible to $\mathbb{P}(\cdot \mid \cdot)$ in eq. (2.12). This is done by minimising the Kullback-Leibler (KL) divergence $\mathcal{D}_{\text{KL}}$ (Murphy, 2012, §21).

Lets consider the reverse KL divergence, whilst noting that this is not a symmetric property i.e.

$$\mathcal{D}_{\text{KL}}(\mathbb{Q} \,||\, \mathbb{P}) \neq \mathcal{D}_{\text{KL}}(\mathbb{P} \,||\, \mathbb{Q}) \tag{2.14}$$

is given by

$$\mathcal{D}_{\text{KL}}(\mathbb{Q} \,||\, \mathbb{P}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x} \mid \mathbf{y})} \tag{2.15}$$

and essentially measures the amount of information required to "distort" (Jang, 2016) $\mathbb{P}(\mathbf{X})$ into $\mathbb{Q}_\phi(\mathbf{X})$. Consequently we seek to minimise the amount of distortion required by manipulating ϕ. But, this quantity is still not tractable as is, since it requires the point wise evaluation of the intractable normalisation constant (Murphy, 2012). Instead, consider the substitution of the conditional distribution

$$\mathbb{P}(\mathbf{x} \mid \mathbf{y}) = \frac{\mathbb{P}(\mathbf{x}, \mathbf{y})}{\mathbb{P}(\mathbf{y})} \tag{2.16}$$

into eq. (2.15):

$$\begin{aligned}
\mathcal{D}_{\text{KL}}(\mathbb{Q} \,||\, \mathbb{P}) &= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x} \mid \mathbf{y})} \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})\, \mathbb{P}(\mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \left(\log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} + \log \mathbb{P}(\mathbf{y}) \right) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} + \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \mathbb{P}(\mathbf{y}) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} + \log \mathbb{P}(\mathbf{y}) \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \qquad (2.17) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} + \log \mathbb{P}(\mathbf{y}) \qquad (2.18)
\end{aligned}$$

and note that the difference between eq. (2.17) and eq. (2.18) came about since

$$\sum_{\mathbf{x} \in \mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) = 1.$$

Consequently to minimise $\mathcal{D}_{\text{KL}}(\mathbb{Q} \,||\, \mathbb{P})$ we only need to minimise the first part of eq. (2.18), since $\log \mathbb{P}(\mathbf{y})$ is independent of ϕ. Now, recall that the $\mathcal{D}_{\text{KL}}$ is the expectation of the

logarithmic difference between the probabilities $\mathbb{Q}$ and $\mathbb{P}$, where the expectation is taken using the probabilities $\mathbb{Q}$ – we use this to re-write eq. (2.18) w.r.t. expectations

$$\begin{aligned}\sum_{\mathbf{x}\in\mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})}\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) - \log \mathbb{P}(\mathbf{x}, \mathbf{y})\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) - \log \mathbb{P}(\mathbf{y} \mid \mathbf{x}) - \log \mathbb{P}(\mathbf{x})\right]. \quad (2.19)\end{aligned}$$

We are now well on our way to establishing a variational lower bound. By minimising eq. (2.19) we are maximising the negative of the same function

$$\begin{aligned}\mathcal{L}(\mathbb{Q}) &= -\ eq.\ (2.19) \\ &= -\sum_{\mathbf{x}\in\mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}(\mathbf{x}, \mathbf{y})} \\ &= -\mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) - \log \mathbb{P}(\mathbf{y} \mid \mathbf{x}) - \log \mathbb{P}(\mathbf{x})\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[-\log \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) + \log \mathbb{P}(\mathbf{y} \mid \mathbf{x}) + \log \mathbb{P}(\mathbf{x})\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}(\mathbf{y} \mid \mathbf{x}) + \log \frac{\mathbb{P}(\mathbf{x})}{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}\right] \quad (2.20)\end{aligned}$$

and thus we have established a variational lower bound, provided we can evaluate the densities in the above expression. But we can go one step further to yield a more intuitive (Jang, 2016) expression

$$\begin{aligned}\mathcal{L}(\mathbb{Q}) &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}(\mathbf{y} \mid \mathbf{x}) + \log \frac{\mathbb{P}(\mathbf{x})}{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}(\mathbf{y} \mid \mathbf{x})\right] + \sum_{\mathbf{x}\in\mathbf{X}} \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \log \frac{\mathbb{P}(\mathbf{x})}{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})} \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}(\mathbf{y} \mid \mathbf{x})\right] - \mathcal{D}_{\mathrm{KL}}(\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \mid \mathbb{P}(\mathbf{x})). \quad (2.21)\end{aligned}$$

There are now two operations that we can leverage over:

1. If we sample a latent variable $\mathbf{x} \sim \mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})$ then we have an *encoding* which processes our observation and converts it to latent code $\mathbf{x}$.
2. Then by sampling $\mathbf{y} \sim \mathbb{Q}_\phi(\mathbf{y} \mid \mathbf{x})$ we are *decoding* our latent code $\mathbf{x}$ back into observed space, thereby reconstructing it to its original form (with some loss, unless it is a perfect reconstruction).

This measure $\mathcal{L}$ tells us how effectively the decoder has learned to reconstruct the input $\mathbf{y}$, given its latent representation $\mathbf{x}$. By framing the loss as the sum over the expected decoding likelihood and the KL divergence between the variational approximation and the prior on $\mathbf{x}$, we force the variational distribution to decode a sample of $\mathbf{x}$ back to $\mathbf{y}$ as well as possible.

The most popular form of variational inference is the one known as the mean field approximation (Murphy, 2012, §21.3), and is a simple assumption on part of the posterior, where we assume that it factors as

$$\mathbb{Q}_\phi(\mathbf{X}) = \prod_{i=1}^{N} \mathbb{Q}_\phi(\mathbf{x}_i) \tag{2.22}$$

and we optimise the parameters of each marginal distribution. From a statistical physics point of view, "mean field" refers to the relaxation of a difficult optimisation problem to a simpler one which ignores second-order effects (Mezard & Montanari, 2009, §4.4.2).

2.1.2.4 Evidence lower bound

Why is it called a *variational lower bound*? Recall that we can decompose (Bishop, 2006, §10.1) the log marginal probability into

$$\log \mathbb{P}(\mathbf{x}) = \mathcal{L}(\mathbb{Q}) + \mathcal{D}_{\mathrm{KL}}(\mathbb{Q} \,||\, \mathbb{P}). \tag{2.23}$$

This means that under the true distribution, the log-likelihood of a data point $\mathbf{y}$, is the sum of $\mathcal{L}(\mathbb{Q})$ and an error term $\mathcal{D}_{\mathrm{KL}}(\mathbb{Q} \,||\, \mathbb{P})$ (Jang, 2016). The error term captures the distance between our proposal distribution $\mathbb{Q}(\mathbf{x} \mid \mathbf{y})$ and the true distribution $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$ at that particular observation $\mathbf{y}$. But the Kullback-Leibler divergence is always strictly greater or equal to zero. This means that since $\mathcal{D}_{\mathrm{KL}}(\mathbb{Q} \,||\, \mathbb{P}) \geq 0$ then $\log \mathbb{P}(\mathbf{x})$ must be greater than $\mathcal{L}$, consequently $\mathcal{L}$ is a *lower bound* on $\log \mathbb{P}(\mathbf{x})$. Sometimes $\mathcal{L}$ is referred to as the evidence lower bound (ELBO), but then takes on a different manifestation:

$$\begin{aligned} \mathcal{L} &= \log \mathbb{P}(\mathbf{x}) - \mathcal{D}_{\mathrm{KL}}(\mathbb{Q}(\mathbf{x} \mid \mathbf{y}) \,||\, \mathbb{P}(\mathbf{x} \mid \mathbf{y})) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})} \left[\log \mathbb{P}(\mathbf{y} \mid \mathbf{x})\right] - \mathcal{D}_{\mathrm{KL}}(\mathbb{Q}(\mathbf{x} \mid \mathbf{y}) \,||\, \mathbb{P}(\mathbf{x})) \end{aligned} \tag{2.24}$$

where $\log \mathbb{P}(\mathbf{x})$ is *always* greater or equal to this evidence lower bound.

2.1.2.5 Deriving the variational autoencoder

In the VAE, the encoder $\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})$ consists of a neural network – see fig. 2.2b. It takes as input an observation $\mathbf{y}$ and outputs a latent representation $\mathbf{x}$, and has weights and biases labelled ϕ. The decoder $\mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})$ is similarly a neural net. Its input is the latent representation $\mathbf{x}$, it outputs the parameters to the probability distribution of the observations, with weights and biases θ. The encoder must learn an efficient compression of the data into a lower-dimensional representation $\mathbf{x}$. Because $\mathbb{Q}_\phi$ is a typically a Gaussian density, we can sample noisy representations of $\mathbf{x}$. Conversely the decoder $\mathbb{P}_\theta$ takes the latent representation $\mathbf{x}$ and tries to decode the original input $\mathbf{y}$. Throughout this computational flow, information will be lost, the magnitude of which we measure with the reconstruction log-likelihood $\log \mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})$, given by

$$\mathbb{E}_{\mathbf{X}\sim\mathbb{Q}_\phi(\mathbf{X}|\mathbf{Y})}\left[\log \mathbb{P}_\theta(\mathbf{Y} \mid \mathbf{X})\right] - \mathcal{D}_{\mathrm{KL}}(\mathbb{Q}_\phi(\mathbf{X} \mid \mathbf{Y}) \mid\mid \mathbb{P}_\theta(\mathbf{X})) \tag{2.25}$$

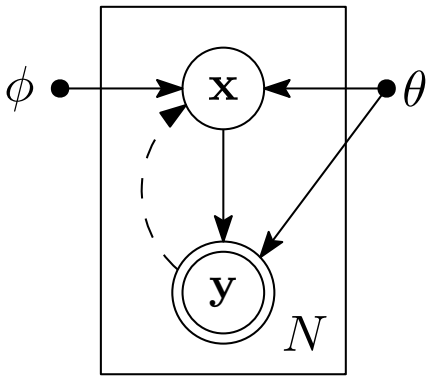
where capital bold variables contain all data points. We can train the VAE using gradient descent to optimise the loss with respect to the parameters of the encoder and decoder. Apart from giving us a low-dimensional representation of our sequential observation, the VAE construction lends itself further to our methodological design. We assume that our observation model is Gaussian in our generative modelling framework. The VAE, by construction, yields latent variables which are normally distributed and it is these that we use for classification and segmentation. For further information see the original paper by Kingma & Welling (2013).

Derivation of the evidence lower bound (ELBO) of the VAE specifies that we have a dataset $\mathcal{D} = \left\{\mathbf{y}^{(n)}\right\}_{n=1}^{N}$. Then we seek to maximise the marginal likelihood of our observations under our model. Because all observations are taken to be I.I.D. we can express this as

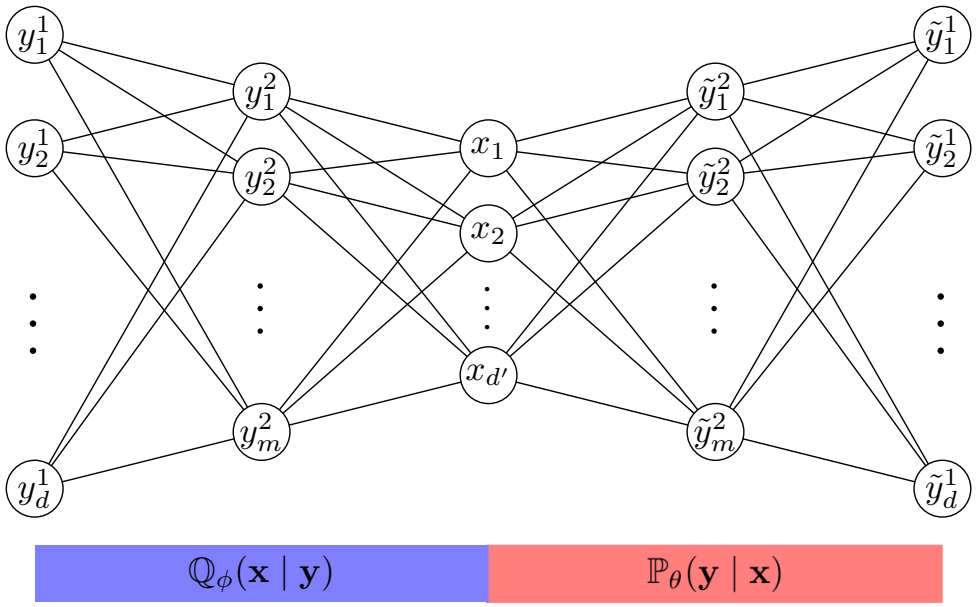
$$\log \mathbb{P}_\theta\left(\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\right) = \log \sum_{n=1}^{N} \mathbb{P}_\theta\left(\mathbf{y}^{(n)}\right) \tag{2.26}$$

from whence maximisation is given by

$$\begin{aligned}
\theta^* &= \arg\max_\theta \sum_{n=1}^{N} \mathbb{P}_\theta\left(\mathbf{y}^{(n)}\right) \\
&\texttt{/ Use the re-parametrisation trick /} \\
&= \arg\max_\theta \sum_{n=1}^{N} \log \mathbb{P}_\theta\left(\mathbf{y}^{(n)}\right).
\end{aligned}$$



(a) The variational autoencoder as a graphical model using plate-notation. Solid lines denote the generative model and dashed lines the variational approximation (Kingma & Welling, 2013). The variational parameters ϕ are jointly learned with the generative θ.



(b) Rolled out version of the graphical model shown in fig. 2.2a. The encoder network is underscored by the blue bar, and the decoder network by the red bar.

Figure 2.2: Variational autoencoder graphical models. A compact plate-notation version is shown in fig. 2.2a and its rolled-out analogue shown in fig. 2.2b.

Note that the problem with many models such as the VAE, is that they use neural networks as encoders and decoders. To implement encoder and decoder as a neural network, you need to backpropogate through random sampling and that is the problem because backpropogation cannot flow through random nodes; to overcome this obstacle, we use the reparameterization trick.

The reparametrization trick (Kingma, 2013; Dumoulin et al., 2016) refers to the operation of instead of directly sampling from the distribution of interest, the random variable is computed as a deterministic transformation of some chosen noise model, such that its

distribution is the desired distribution (Dumoulin et al., 2016).

Having dealt with that, our problem boils down to one of finding an expression for $\log \mathbb{P}_\theta\left(\mathbf{y}^{(n)}\right)$ under our model. For notational brevity and to reduce clutter, we will remove the observation index n in the derivation

$$\begin{aligned}
\log \mathbb{P}_\theta(\mathbf{y}) &= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}_\theta(\mathbf{y})\right] \quad \text{Since } \mathbb{P}_\theta(\mathbf{y}) \perp\!\!\!\perp \mathbf{x} \\
&= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \frac{\mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})\,\mathbb{P}_\theta(\mathbf{x})}{\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{y})}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \frac{\mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})\,\mathbb{P}_\theta(\mathbf{x})}{\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{y})}\frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})\right] - \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}_\theta(\mathbf{x})}\right] \\
&+ \mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \frac{\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})}{\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{y})}\right] \\
&= \underbrace{\mathbb{E}_{\mathbf{x}\sim\mathbb{Q}_\phi(\mathbf{x}|\mathbf{y})}\left[\log \mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})\right] - \mathcal{D}_{\mathrm{KL}}\left(\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \,||\, \mathbb{P}_\theta(\mathbf{x})\right)}_{\mathcal{L}(\mathbf{y}^{(n)},\theta,\phi)} \\
&+ \underbrace{\mathcal{D}_{\mathrm{KL}}\left(\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y}) \,||\, \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{y})\right)}_{\geq 0}.
\end{aligned} \tag{2.27}$$

Then the ELBO of the VAE is given by $\log \mathbb{P}_\theta\left(\mathbf{y}^{(n)}\right) \geq \mathcal{L}(\mathbf{y}^{(n)}, \theta, \phi)$. If $\mathbb{P}_\theta(\mathbf{y} \mid \mathbf{x})$ and $\mathbb{Q}_\phi(\mathbf{x} \mid \mathbf{y})$ can be computed point wise, and are differentiable with respect to their parameters, the ELBO can be maximised via gradient descent. Explicitly, training amounts to maximising

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{n=1}^{N} \mathcal{L}(\mathbf{y}^{(n)}, \theta, \phi). \tag{2.28}$$

That is the basic machination required for a standard VAE. In our model the ELBO takes a different form but follows the same approach as that in eq. (2.27).

2.2 Classification methods

Like §2.1 we do not propose to investigate, in incisive detail, the classification methods used in chapter 3. We provide a brief, and predominantly qualitative, synopsis and invite the reader to seek out the relevant literature cited below for each method. The reasoning for this is the same as previously; classification algorithms have been studied for decades, and the repetition of what is a standard feature in any machine learning toolbox, is best avoided.

2.2.1 Pruned C4.5 decision tree

Nominally a *decision tree* (see fig. 2.3) is a method by which the input space to the algorithm is recursively partitioned, and a local model is defined in each resulting region of the input space. Murphy (2012) explains that this can be represented by a tree, with one leaf per partitioned region. Witten et al. (2016) notes that improvements to basic decision tree induction "culminated in a practical and influential system for decision tree induction called C4.5". Under these improvements methods were developed for dealing with missing values, noisy data, as well as generating rules from other trees. Once a decision tree has been fully expanded, it typically contains unnecessary structure for the problem at hand. The act of simplifying said structure is known as *pruning* (Witten et al., 2016).

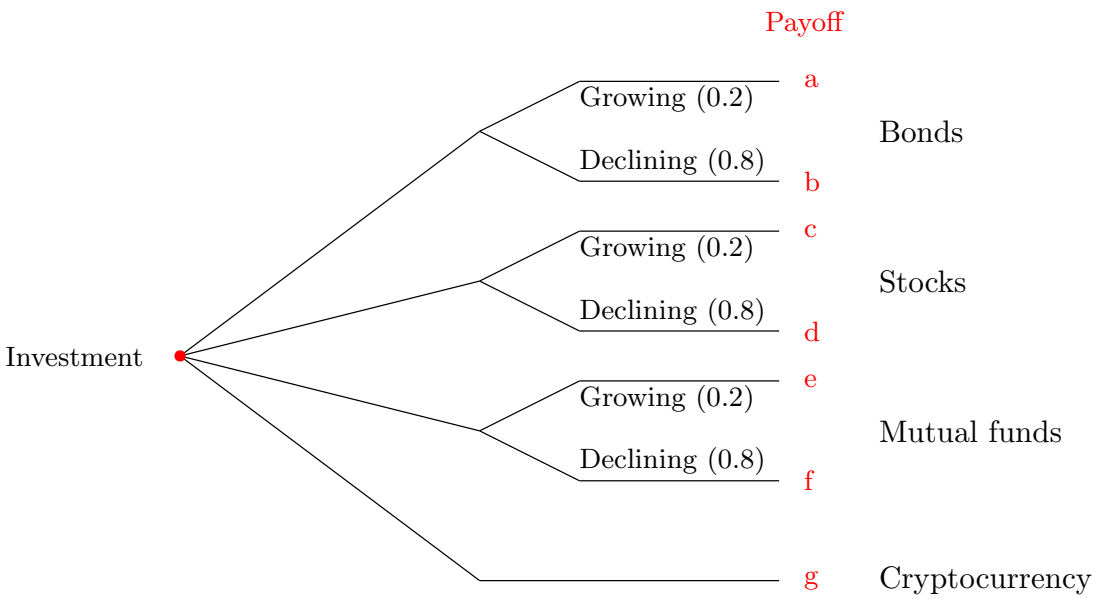


Figure 2.3: A simple example of a decision tree, illustrating various financial instruments, which each yield a binary reward (payoff) depending on the increase of decrease of that particular instrument.

2.2.2 Propositional rule learner

Decision tree algorithms employ a divide-and-conquer approach in order to solve the classification problem (Witten et al., 2016). As a decision tree is learned, it seeks attributes, which it can separate into classes. But another approach to the classification task is to learn a rule at each stage instead, where a rule-based classifier uses a set of '`if` [*condition*] `then`' rules for classification (Han et al., 2011a).

2.2.3 Naïve Bayes classifier

The naïve Bayesian classifier works by taking a dataset made up of tuples (observations $\mathbf{X}$ and classes $\mathcal{C}$), where the classifier will predict that $\mathbf{x}_n \in \mathbf{X}$ belongs to class $c \in \mathcal{C}$, with the highest probability (Han et al., 2011a), conditioned on $\mathbf{X}$. The *naïve* epitet is derived thus

$$\mathbf{x}_n \text{ belongs to } c_i \text{ iff } \mathbb{P}(c_i \mid \mathbf{X}) > \mathbb{P}(c_j \mid \mathbf{X}) \text{ for } 1 \leq j \leq |\mathcal{C}| \text{ and } i \neq j \tag{2.29}$$

where the conditionals are the posterior densities on the class instance. The classifier assumes very strong (i.e. naïve) independence assumptions between the features.

2.2.4 k-Nearest neighbours

The kNN classifier is a simple algorithm available in our arsenal. An incisive discussion is provided by Murphy (2012, §1.4.2) and Bishop (2006, §2.5.2). It is summarised as follows: for each point $\mathbf{x}_n \in \mathbf{X}$ the classifier considers all k neighbours in the vicinity of $\mathbf{x}_n$, counts how many members of each class are in this set, and the fraction return amounts to an estimate of the class membership.

2.2.5 Multiclass support vector machine

With the help of a nonlinear mapping, the SVM transforms observations $\mathbf{X}$ into a higher dimension. In a sufficiently high dimensional space, any two, multivariate observations can be linearly separated. The separating hyperplane is found using *support vectors*. Critical boundary instances, between classes, are called support vectors and are used to build a linear discriminant function that separates them as much as possible (Witten et al., 2016).

2.2.6 Random forest

Decision tress, as introduced above, are what Murphy (2012) calls "high variance estimators", meaning that small changes in the input observations can have a very large effect on the structure of the tree, making it an inherently unstable method. One way to deal with this variance problem is to average many estimates, i.e. by training multiple trees on random and different parts of the input space and then take the mean of the whole thing.

Unsurprisingly, this method is known as a *random forest*. This way of averaging trees is also known as *bagging*.

2.2.7 Boosting of the fast decision tree learner

Bagging, as has been explained, inherently takes advantage of learning instability. But it only really works when the learned models are inherently different from one another on their separate domains (Witten et al., 2016). This is intuitive enough. Better would be a scenario in which models complemented each other; if each was an expert in their own domain, then they would be able to complement each other. The *boosting* method does this precisely, by seeking advise from multiple models rather than one. Like bagging it is an ensemble method (Han et al., 2011a).

2.3 Bayesian optimisation

Bayesian optimisation is a type of optimisation concerned with global optimisation of black-box function. For the uninitiated, a 'black-box' in this parlance means a system, function or model, which we can only analyse by way of considering its inputs and outputs, and nothing else. Nothing, in this sense, is known about the internal workings. Those are the functions, which Bayesian optimisation seeks to tackle.

Definition 2.3.1. (Bayesian optimisation). Bayesian optimisation (BO) is a strategy for global optimisation of $f : \mathcal{X} \to \mathbb{R}$. Where f is a $L-$Lipschitz continuous function[1] (Eriksson et al., 2013), defined on a compact subset $\mathcal{X} \subseteq \mathbb{R}^D$ – i.e. it contains all its limit points and is bounded, by having all its points lie within some fixed distance of each other. The idea is to find the global maximum defined on the subset

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}). \tag{2.30}$$

The overbearing assumption is that only noisy function evaluations are available from f, of the type $\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i$ with noise distributed as $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \Sigma)$. We then define a

[1] A function $f : A \to \mathbb{R}^m, A \subset \mathbb{R}^n$, is said to be L–Lipschitz, $L \geq 0$, if $|f(a) - f(b)| \leq L|a - b|$ for every pair $(a, b) \in A$, where L is a constant, independent of a and b (O'Searcoid, 2006).

cumulative regret[2] which is a measure which we seek to minimise

$$r_N = Nf(\mathbf{x}^*) - \sum_{j=1}^{N} f(\mathbf{x}_j^*), \tag{2.31}$$

by using evaluations $\mathbf{X} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from f – an operation which requires an acquisition function which specifies the $\mathbf{x}$ to try next (see §2.3.1 for further details). Bull (2011) explains that the cumulative regret is minimised the closer we get to the optima $\mathbf{x}^*$.

As is well explained by the authors of GPyOpt (2016) there are two bits of the BO framework that are especially important to get right:

1. We need to define a prior belief about our function f. Fortunately, such a Bayesian function space can be imposed via the Gaussian process, which furthermore allows us to condition on the available observations and henceforth yield a posterior.
2. An acquisition function is required in order to know where to sample next, so that maximum information is received regarding the location of the global maximum of the function under investigation. We discuss it further in §2.3.1.

With these properties, the model can be updated and the acquisition function re-optimised, each time a new data point is received, until the process converges on a model f, or some upper bound of optimisation iterations is reached. A simple example of this is demonstrated in example 2.1.

EXAMPLE 2.1: NOISY MINIMISATION

In this example we demonstrate how Bayesian optimisation works in a one-dimensional setting but one made a bit more difficult due to the added noise. The true (noiseless) minimum of this function resides at $f(x^*) = -1.7596$.

$$f(x) = \sin(3x) \cdot (2 - \tanh(x^2)) + \mathcal{N}(0, 0.2) \tag{2.32}$$

For this task we use the expected improvement acquisition function and evaluate the function 20 times, where x is bounded by $[-2, 2]$.

2.3.1 Acquisition Function

The role of the acquisition function is to guide the search for the optimum (Brochu et al., 2010), by explicitly encoding a trade-off between *exploitation* (evaluating at points

[2]Cumulative regret is the expected regret of not having sampled the single best option in hindsight (Pepels et al., 2014).

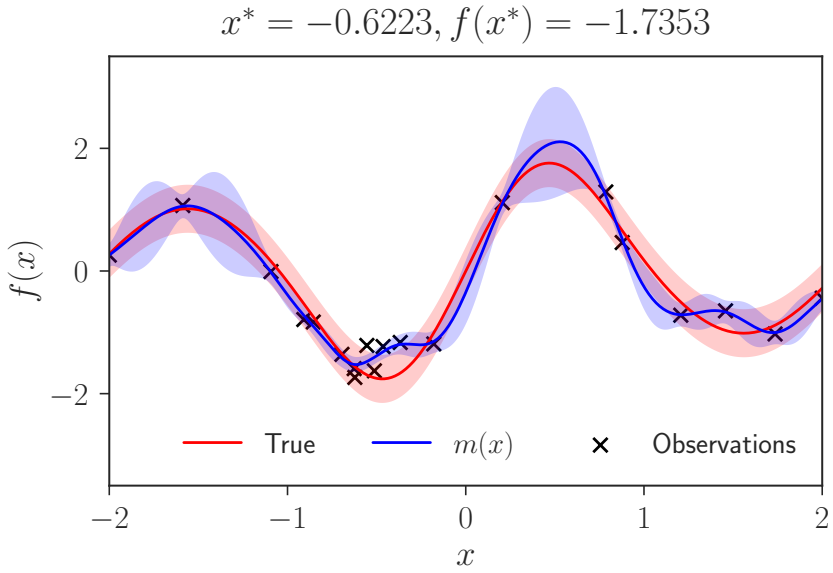


Figure 2.4: Results of using Bayesian optimisation on a one dimensional noisy optimisation task, with the goal of finding the minimum value of eq. (2.32). Shaded regions depict confidence intervals at twice the standard deviation around the predictive mean and true function value (in red). The plot title demonstrates the final results of optimisation, compare those with the true values in example 2.1.

with low mean) and *exploration* (evaluating at points with high uncertainty). Intuitively, the acquisition function evaluates the utility of a set of candidate points for the next evaluation of the objective function – effectively a means by which 'resources' (for example computational) are most effectively allocated. Typically, acquisition functions are defined such that high acquisition corresponds to potentially high values of the objective function $f(\mathbf{x})$, whether because the prediction is high, the uncertainty is great, or both. Maximising the acquisition function is used to select the next point at which to evaluate f. That is to say, we wish to sample f at $\arg\max_{\mathbf{x}} a(\mathbf{x} \mid \mathcal{D})$ (Snoek et al., 2012), where $\mathcal{D}$ is the available data from previous experiments (i.e. where f was previously evaluated at the 'advise' of the acquisition function, this also includes the noisy outputs).

Though there are many choices of $a(\cdot)$, a common choice, and the one used in this thesis, is one where we seek to maximise the expected improvement (EI) over the current best. A small exposition is provided thus; Mockus et al. (1978) define the improvement function as

$$\mathrm{I}(\mathbf{x}) = \max\{0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}_{\mathsf{best}})\} \tag{2.33}$$

which is to say that $\mathrm{I}(\mathbf{x})$ is positive when the prediction is higher than the best value known thus far, otherwise it is set to zero. The next evaluation point is found by maximising the

expected improvement

$$\mathbf{x}_{\mathsf{next}} = \arg\max_{\mathbf{x}} \mathbb{E}\left[\max\{0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}_{\mathsf{best}})\} \mid \mathcal{D}_t\right]. \tag{2.34}$$

This acquisition function $a_{\mathsf{EI}}(\mathbf{x})$, has closed form under the GP

$$\begin{cases} (\mu(\mathbf{x}) - f(\mathbf{x}_{\mathsf{best}}))\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) \geq 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \tag{2.35}$$

with

$$Z = \frac{\mu(\mathbf{x}) - f(\mathbf{x}_{\mathsf{best}})}{\sigma(\mathbf{x})}.$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the PDF and CDF of the standard normal distribution respectively, with $\mu(\cdot)$ and $\sigma(\cdot)$ the mean and variance of the same.

2.4 State-space models

In this section we build upon the theory unveiled in the previous one, to tackle the notion of *state-space models* (SSM).

Definition 2.4.1. (State-space model). A type of model which uses state variables to describe a system by a set of first-order differential or difference equations (in favour of using one or more n^{th}–order differential or difference equations). More precisely they are a type of probabilistic graphical model, which describe the stochastic dependence between latent state variable and the observed measurement (Koller & Friedman, 2009). The latent states can be reconstructed from the measured input-output observations, but are not themselves measured during a recorded event or phenomena under study (Durbin & Koopman, 2012; Friedland, 2012; Hinrichsen & Pritchard, 2005).

In this work we shall be dealing with the two most important (Bishop, 2006, §13) examples of SSMs; the *hidden Markov model*, where the latent variables are discrete, and *linear dynamical systems*, in which the latent variables are Gaussian. The nomenclature in the literature is not quite precise regarding taxonomy of the these two models, in the grander scheme of SSMs, where some authors prefer to attach the 'SSM' label only to the Kalman filter but not to the HMM (Murphy, 2012, §18.1). This is not of any major importance, save for knowing that the issue exists.

We begin by examining the HMM.

2.4.1 Hidden Markov model

The graphical model of the hidden Markov model is shown in fig. 2.5. Shown is a Markov chain with latent variables given by $x_{0:T}$ and observed of measured variables given by $y_{1:T}$ (we frame this exposition through univariate variables). The first-order Markovian nature of the model manifests itself through the conditional relationships found in the latent and observed structure of the model. An *emission model* arises from the conditional relationship $\mathbb{P}(y_t \mid x_t)$ and a *transition model* through $\mathbb{P}(x_t \mid x_{t-1})$ (Barber, 2012). A formal definition is provided in definition 2.4.2.

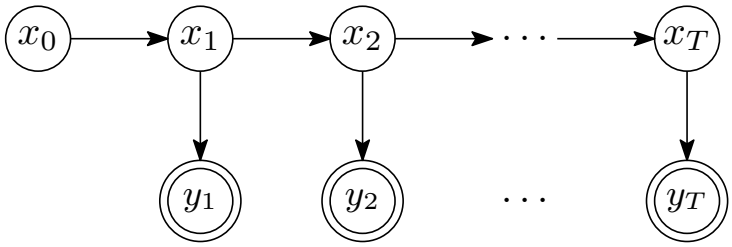


Figure 2.5: Directed graphical model of a state-space model (of the hidden Markov model is a subset). Double barreled nodes represent observed (measured) variables and single barreled represent latent (hidden) variables.

Definition 2.4.2. (Hidden Markov model). A hidden Markov model (HMM) is a doubly-stochastic[3] Markov chain in which the state-sequence $x_{1:T}$ is drawn according to a Markov chain on a discrete state-space defined on $\mathcal{X}$ (Teh & Jordan, 2010, §3). The size of $\mathcal{X}$ is given as $|\mathcal{X}| = m$. In parallel, there evolves an observation sequence $y_{1:T}$ which is step-wise conditionally dependent on $x_{1:T}$. Each observation y_t is conditionally independent of all the other observations given x_t. The state-specific transition distribution π_k for state k, allows latent states to evolve as $x_t \mid x_{t-1} \sim \pi_{x_{t-1}}$, yielding the generative process for the HMM:

$$x_1 \sim \pi_0, \;\; \pi_0 \in \mathbb{R}_+^m \qquad \text{Initial state distribution} \qquad (2.36)$$

$$x_t \mid x_{t-1} \sim \boldsymbol{\pi}_{x_{t-1}}, \;\; \boldsymbol{\pi} \in \mathbb{R}_+^{m \times m} \qquad \text{Transition distribution} \qquad (2.37)$$

$$y_t \mid x_t \sim \mathbb{P}(\cdot \mid \theta_{x_t}), \theta \in \boldsymbol{\theta} \qquad \text{Emission distribution} \qquad (2.38)$$

where π_k is the k^{th} row of transition matrix $\boldsymbol{\pi} \triangleq \{\pi_k\}_{k=1}^m$ and θ_k the state-specific emission parameters s.t. $\boldsymbol{\theta} \triangleq \{\theta_k\}_{k=0}^m$. Transition function are implicitly normalised by virtue of the

[3]The two sources of uncertainty in the HMM are (1) the state transition distribution and (2) the emission distribution – which we model as draws from a probability mass and density function respectively.

modelling density employed in the model construction, where we either perform element-wise division by the sum of the transition function or use e.g. a Dirichlet distribution as our density.

Given definition 2.4.2 and provided that we have specified a density for the emission distribution, the joint density for an HMM with T observations is given by

$$\mathbb{P}(x_{1:T}, y_{1:T}) = \mathbb{P}(x_0) \prod_{t=1}^{T} \mathbb{P}(x_t \mid x_{t-1}) \, \mathbb{P}(y_t \mid x_t) \tag{2.39}$$

where we have written $\mathbb{P}(x_0) = \pi_0(x_0)$, and implicitly conditioned on the model parameters. Comparing eq. (2.39) and fig. 2.5 we can clearly deduce the origins of the model structure. Thus we see that a HMM defines a joint distribution over latent variables and observed measurements. For a much deeper discussion on HMM origins see the seminal paper by Rabiner (1989) or (Barber, 2012, §23.2). Now, given that we will be fully Bayesian in this thesis, the extension to the Bayesian HMM (BHMM) is warranted.

Definition 2.4.3. (Bayesian hidden Markov model). The BHMM places a prior on the model parameters introduced in definition 2.4.2, like so

$$\mathbb{P}(x_{1:T}, y_{1:T}) = \mathbb{P}(x_0) \int_{\Delta_{\boldsymbol{\Psi}}} \prod_{t=1}^{T} \mathbb{P}(x_t \mid x_{t-1}, \boldsymbol{\Psi}) \, \mathbb{P}(y_t \mid x_t, \boldsymbol{\Psi}) \, \mathbb{P}(\boldsymbol{\Psi}) \, \mathrm{d}\boldsymbol{\Psi} \tag{2.40}$$

where the model elements are constructed as follows:

$$\pi_0 \sim \text{Dirichlet}(\alpha_0) \tag{2.41}$$

$$\pi_k \sim \text{Dirichlet}(\alpha_k) \qquad \text{for } k = 1, \ldots, m \tag{2.42}$$

$$\theta_k \sim \mathbb{P}(\cdot \mid \phi_k) \qquad \text{for } k = 1, \ldots, m \tag{2.43}$$

where the BHMM consequently has model hyperparameters: α_0, $\boldsymbol{\alpha} \triangleq \{\alpha_k\}_{k=1}^m$, $\boldsymbol{\phi} \triangleq \{\phi_k\}_{k=1}^m$ and for convenience $\boldsymbol{\Psi} \triangleq \{\alpha_0, \boldsymbol{\alpha}, \boldsymbol{\phi}\}$. Once parameters are sampled, the generative process evolves as outlined in definition 2.4.2. We use the notation $\mathbb{P}(\cdot \mid \phi_k)$ to denote an arbitrary prior on the emission density, with parameters ϕ_k indexed by state k.

In §2.4.1 we have defined a prior on the state transition functions. The reason for this is primarily one of conjugacy. Hidden Markov models have discrete states, and the distribution over state-variables is multinomial (Goldwater & Griffiths, 2007). A multinomial

distribution has a natural prior model: the $m-$dimensional Dirichlet distribution, which is conjugate to the multinomial.

HMMs have found widespread use (in e.g. automatic speech recognition, activity recognition as well as gene finding amongst others (Murphy, 2012; Fox, 2009)) for well over three decades now, and continue to find new ones. Part of its popularity stems from its exact inference properties. If the number of states are fixed *a priori* and do not grow throughout the inference task, we can address the most common inference tasks that HMMs are used for:

- Filtering (inferring the present) via $\mathbb{P}(x_t \mid y_{1:t})$
- Prediction (inferring the future) via $\mathbb{P}(x_t \mid y_{1:h})$ where $t > h$
- Smoothing (inferring the past) via $\mathbb{P}(x_t \mid y_{1:h})$ where $t < h$.

The above list contains but a short snippet of the much more incisive discussion on the subject by Barber (2012, §23.2.1) – for an excellent summary figure see (Murphy, 2012, fig. 17.11). The problems in the above list are *not* those which this thesis focus on primarily. Consequently, we will only briefly consider exact inference for above cases. Hence, the forward-backward algorithm (FBA) is a special case of a more general technique known as belief propagation (Yedidia et al., 2003). Also known as the sum-product algorithm, it computes the marginals needed to distribute the sum over "variable states over the product of factors" (Barber, 2012), where a Markov network has been represented as a factor graph.

Forward-backward is best understood by considering the intermediate functions, in the forward and backward recursions, as *messages*, which convey beliefs about the variables over which they operate. Furthermore, it is worth noting that FBA gives us the marginal probability for each individual state, whereas the Viterbi algorithm (Barber, 2012) gives us the probability of the most likely sequence of states. The difference is important and we can illustrate it with a simple example. If the HMM is used to predict whether or not it will be be pouring down or sunny over a number of days then the FBA will result in the probability of it pouring on each individual day, whereas the Viterbi calculation will result in the most likely sequence of sunny and pouring days, as well as the probability of this dire (or not) weather projection (since we are most concerned with the exact marginals

(Bishop, 2006, §13.2.2)). Hence for a thorough discussion on *exact* inference in the HMM see (Bishop, 2006, §13.2.2) and (Barber, 2012, §23.2).

2.4.2 Linear dynamical system

A linear dynamical system (LDS) is a probabilistic model that captures the time evolution and the measurement processes (Bishop, 2006) of some practical setting. We shall focus on the most important example (Bishop, 2006), the linear-Gaussian state space (LGSS) model, which views the latent variables x, and the observed variables y as multivariate Gaussian distributions whose means are linear functions of the states of their parents in the corresponding tree-structured directed graph (see fig. 2.6 for a graphical model). The LDS is important because unlike almost all other models, it supports exact inference (Murphy, 2012).

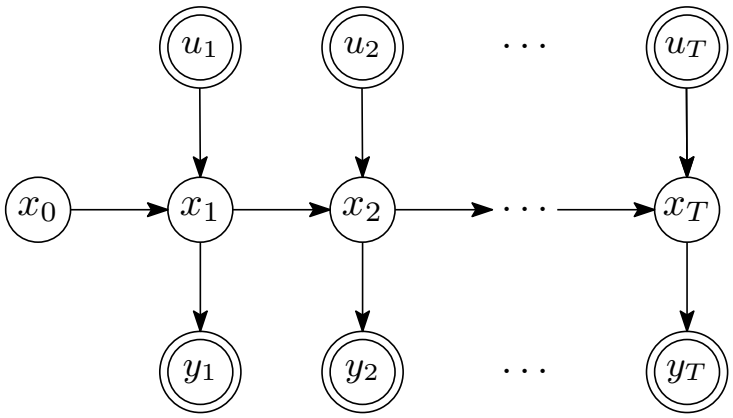


Figure 2.6: A linear dynamical system in which the optional control input u_t is also shown alongside the hidden observed state x_t and y_t respectively. For a more thorough description of the model see (Murphy, 2012, §18.1).

Inference in the graph is efficiently performed using the sum-product algorithm, the *forward recursions* of which are known as the Kalman filter (KF) (Bishop, 2006; Murphy, 2012; Fleet, 2011; MacKay, 2002). The KF is widely used in many online tracking applications, and is suitable for linear-Gaussian observations and motion models. The main strength of Kalman filters, and indeed Kalman smoothers, is that the linear transformation of a Gaussian is a still a Gaussian, as are the associated operations of marginalisation and conditioning. But these strengths come at a cost, which we shall consider briefly below.

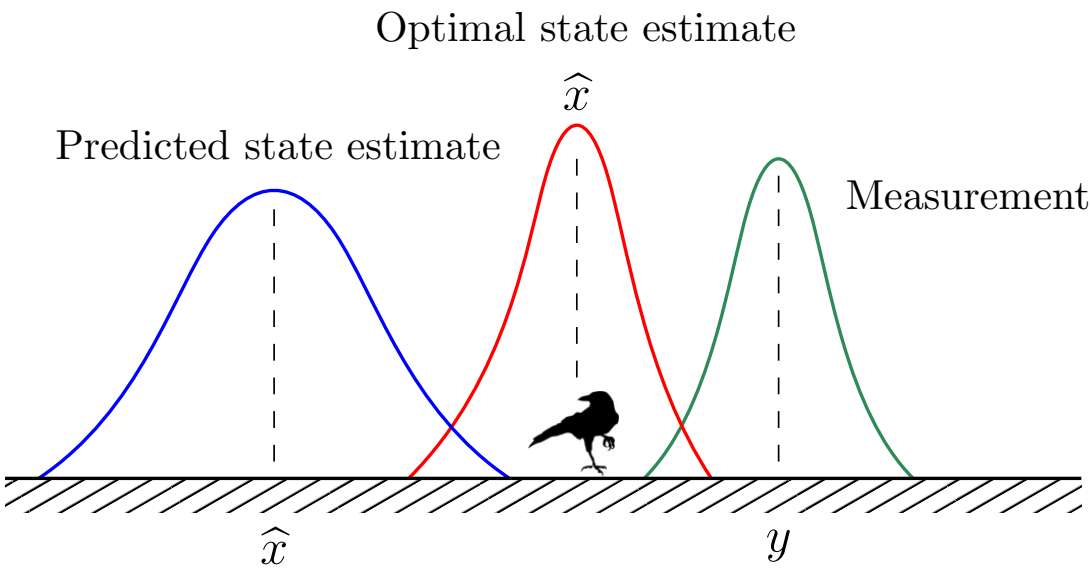


Figure 2.7: Where is the raven? The Kalman filter can tell you. A Kalman filter is an optimal estimation algorithm used to estimate states of a system from indirect and uncertain measurements.

State space model First, consider the foundational model used to study dynamical systems, namely the state-space model (SSM). The introduction to this section gave a brief taste of it, but we shall now dwell on it in more detail. A dynamical system evolves in time, and at each time t, we assume that the system inhabits some state x_t which we index by time (we will not consider the notion of state-less systems). The state, as it stands, contains all the necessary information required for us to garner full insight into its inner workings. If it were for example some completely noiseless observation of a star, provided the observation was rich enough, we should (under the model) be able to regard the inner workings of the thermonuclear fusion process that powers the celestial body. However, this is not possible most of time. Instead what we receive is a measurement of the state y_t also indexed by time. This notion is aptly demonstrated in fig. 2.7. Here we are provided with a model seeking to extract knowledge about a raven's behaviour as it lands on the ground. The model is one-dimensional, thus not that good, so can only make one measurement at time t. But if we combine a whole sequence of measurements (up to and including T) as so $\mathbf{Y}_T = \{y_1, \ldots, y_T\}$ we seek to draw inference about the corresponding latent state sequence $\mathbf{X}_T = \{x_1, \ldots, x_T\}$, by assuming that there is some sort of dependence between all measurements and consequently all latent states. This process is what we term *state inference*. Whereas the static parameters, under our model, that influence that process, fall under *parameter inference* as noted by Schön et al. (2015).

Hence, there is more to SSMs (an independent field of research in fact) than the humble KF. We will now demonstrate how to get from this general hypothesis of dynamics, to a tool which we can use. For more challenging probabilistic models, such as non-linear and non-Gaussian SSMs, as well as more general non-Markovian latent variable models, inference is intractable and one has to turn to advanced simulations methods, some of which we employ in this thesis (and covered extensively in these preliminaries). For now though, consider the general, and as of now (for sake of argument), non-linear or linear, Gaussian or non-Gaussian SSM:

$$x_t \mid x_{t-1}, \phi \sim f_\phi(x_t \mid x_{t-1}) \qquad \text{System process} \tag{2.44}$$

$$y_t \mid x_t, \lambda \sim g_\lambda(y_t \mid x_t) \qquad \text{Observation process} \tag{2.45}$$

in which the initial hidden state is distributed according to some distribution

$$x_1 \sim \mu_\omega(x_1),\ \omega \in \Omega \in \mathbb{R}^{n_\omega} \qquad \text{Initial state distribution} \tag{2.46}$$

where the states and the observed measurements are denoted by $x_t \in \mathsf{X} \subseteq \mathbb{R}^{n_x}$ and $y_t \in \mathsf{Y} \subseteq \mathbb{R}^{n_y}$ respectively, $\forall t \in \mathcal{T} \triangleq \{1, \ldots, T\}$. Now, the dynamics and the observations are modelled by probability density functions $f_\phi(\cdot)$ and $g_\lambda(\cdot)$ respectively, parametrised by $\phi \in \Phi \subseteq \mathbb{R}^{n_\phi}$ and $\lambda \in \Lambda \subseteq \mathbb{R}^{n_\lambda}$ respectively. To make full use of this notion, we can express the joint distribution of the SSM as

$$\mathbb{P}(\mathbf{X}_T, \mathbf{Y}_T \mid \boldsymbol{\theta}) = \mathbb{P}(\mathbf{X}_T \mid \boldsymbol{\theta})\, \mathbb{P}(\mathbf{Y}_T \mid \mathbf{X}_T, \boldsymbol{\theta}) \tag{2.47}$$

$$= \mu(x_1 \mid \omega) \prod_{t=2}^{T} f(x_t \mid x_{t-1}, \phi) \prod_{t=2}^{T} g(y_t \mid x_t, \lambda) \tag{2.48}$$

where $\boldsymbol{\theta} = \{\phi, \lambda, \omega\}$. Also note the inherent conditional independence properties of the SSM (apparent by also considering the graphical model in fig. 2.5). This is the most general form of the SSM.

State inference If we seek to perform inference over the states, we adopt the Bayesian paradigm to ascertain the posterior of $\mathbf{X}_T$, i.e.

$$\mathbb{P}(\mathbf{X}_T \mid \mathbf{Y}_T) = \frac{\mathbb{P}(\mathbf{X}_T, \mathbf{Y}_T)}{\mathbb{P}(\mathbf{Y}_T)} \equiv \frac{\mathbb{P}(\mathbf{X}_T)\, \mathbb{P}(\mathbf{Y}_T \mid \mathbf{X}_T)}{\int \mathbb{P}(\mathbf{X}_T, \mathbf{Y}_T)\, \mathrm{d}\mathbf{X}_T}. \tag{2.49}$$

In eq. (2.49), we have omitted the dependence on the model parameters for clarity. Though elegant, this is of limited use since we cannot typically evaluate (high dimensional

integration) the normalising factor in the denominator (Schön et al., 2015; Andrieu et al., 2004). This leads us to the first notion of filtering, which, in its turn, will lead us to smoothing, where we are interested in incorporating each measurement into the posterior of the latent state at time t. Recursive Bayesian estimation follows thus

$$\begin{aligned}
\mathbb{P}(x_t \mid \mathbf{Y}_t) &= \frac{\mathbb{P}(\mathbf{Y}_t \mid x_t)\,\mathbb{P}(x_t)}{\mathbb{P}(\mathbf{Y}_t)} \\
&= \frac{\mathbb{P}(y_t, \mathbf{Y}_{-t} \mid x_t)\,\mathbb{P}(x_t)}{\mathbb{P}(y_t, \mathbf{Y}_{-t})} \\
&= \frac{\mathbb{P}(y_t \mid \mathbf{Y}_{-t}, x_t)\,\mathbb{P}(\mathbf{Y}_{-1} \mid x_t)\,\mathbb{P}(x_t)}{\mathbb{P}(y_t \mid \mathbf{Y}_{-t})\,\mathbb{P}(\mathbf{Y}_{-t})} \\
&= \frac{\mathbb{P}(y_t \mid \mathbf{Y}_{-t}, x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_{-t})\,\mathbb{P}(\mathbf{Y}_{-t})\,\mathbb{P}(x_t)}{\mathbb{P}(y_t \mid \mathbf{Y}_{-t})\,\mathbb{P}(\mathbf{Y}_{-t})\,\mathbb{P}(x_t)} \\
&= \frac{\mathbb{P}(y_t \mid x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_{-t})}{\mathbb{P}(y_t \mid \mathbf{Y}_{-t})} \\
&= \frac{g(y_t \mid x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_{-t})}{\mathbb{P}(y_t \mid \mathbf{Y}_{-t})}
\end{aligned} \tag{2.50}$$

where we have introduced notation $\mathbf{Y}_{-t} \triangleq \{y_1, \ldots, y_{t-1}\}$ and $\mathbf{Y}_t \triangleq \{y_1, \ldots, y_t\}$. The prior-part of eq. (2.50) can be simply characterised as

$$\mathbb{P}(x_t \mid \mathbf{Y}_{-t}) = \int f(x_t \mid x_{t-1})\,\mathbb{P}(x_t \mid \mathbf{Y}_{-t})\,\mathrm{d}x_t, \tag{2.51}$$

where the evidence of eq. (2.50) provides us (for free) with the marginal likelihood of the model:

$$\mathbb{P}(\mathbf{Y}_t) = \mathbb{P}(y_1) \prod_{t=2}^{T} \mathbb{P}(y_t \mid \mathbf{Y}_{-t}). \tag{2.52}$$

Thus, our inference only considers information up until and including the present time t, conditioned on all the information provided by the observations up until now. From the established literature point-of-view, what we have arrived at now are two things; first, eq. (2.51) is known as the prediction step, whereas eq. (2.50) is known as the update or measurement step (Murphy, 2012; Doucet & Johansen, 2009). This becomes relevant in §2.4.2.1 where we plug in the necessary densities to reach the aforementioned filters and smoothers.

The smoothing problem on the other hand assumes that we have information up until and including time T (also known as the end), and seek the marginal distributions $\mathbb{P}(x_t \mid \mathbf{Y}_T)$ or $\mathbb{P}(\mathbf{X}_T \mid \mathbf{Y}_T)$. Having derived the filtering relations the Bayesian smoothing relations follow

effortlessly. The Markov property[4] tells us that x_t is independent of $y_{t+1:T}$ conditioned on x_{t+1} yielding $\mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_T) = \mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_t)$ (Särkkä, 2013). Consequently

$$\begin{aligned}\mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_T) &= \mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_t) \\ &= \frac{\mathbb{P}(x_t, x_{t+1} \mid \mathbf{Y}_t)}{\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t)} \\ &= \frac{\mathbb{P}(x_{t+1} \mid x_t, \mathbf{Y}_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_t)}{\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t)} \\ &= \frac{\mathbb{P}(x_{t+1} \mid x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_t)}{\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t)}.\end{aligned} \tag{2.53}$$

The joint conditional distribution on the present latent state and the next future latent state can be factorised as

$$\mathbb{P}(x_t, x_{t+1} \mid \mathbf{Y}_T) = \mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_T)\,\mathbb{P}(x_{t+1} \mid \mathbf{Y}_T) \tag{2.54}$$

where we can then plug in the result from eq. (2.53), and use the Markov property;

$$\begin{aligned}\mathbb{P}(x_t, x_{t+1} \mid \mathbf{Y}_T) &= \mathbb{P}(x_t \mid x_{t+1}, \mathbf{Y}_t)\,\mathbb{P}(x_{t+1} \mid \mathbf{Y}_T) \\ &= \frac{\mathbb{P}(x_{t+1} \mid x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_t)\,\mathbb{P}(x_{t+1} \mid \mathbf{Y}_T)}{\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t)}\end{aligned} \tag{2.55}$$

then by marginalising over the next latent state x_{t+1} we receive

$$\mathbb{P}(x_t \mid \mathbf{Y}_T) = \mathbb{P}(x_t \mid \mathbf{Y}_t) \int \frac{\mathbb{P}(x_{t+1} \mid x_t)\,\mathbb{P}(x_{t+1} \mid \mathbf{Y}_T)}{\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t)}\,\mathrm{d}x_{t+1} \tag{2.56}$$

Here eq. (2.56) is the update step which is preceded by the prediction step

$$\mathbb{P}(x_{t+1} \mid \mathbf{Y}_t) = \int \mathbb{P}(x_{t+1} \mid x_t)\,\mathbb{P}(x_t \mid \mathbf{Y}_t)\,\mathrm{d}x_t \tag{2.57}$$

where $\mathbb{P}(x_t \mid \mathbf{Y}_t)$ is simply the filtering distribution for time step t and $\mathbb{P}(x_{t+1} \mid x_t)$ the system process model (the state transition function). For more details on these derivations see (Särkkä, 2013, §8). To complete the picture we need to consider the system identification (SI) problem, or how to compute the posterior of $\boldsymbol{\theta}$.

[4]The Markov property: the conditional probability distribution of future states of the stochastic process (conditional on both past and present states) depend only on the present state, not on the sequence of events that preceded it.

Parameter inference If we are interested in the parameter problem, we typically assign a prior distribution to the parameters i.e. $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$, upon which the inference problem amounts to computing the posterior distribution $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{Y}_T)$. For now, we see that the SSM-SI problem has been reduced to one of density estimation of eq. (2.47). We shall labour this point a bit further, before plugging in the necessary densities to yield the KF and Kalman smoother (KS).

As explained, often we are interested in state inference, i.e. to infer the latent states given some observations. This could the true position of a missile, guided by GPS, or the true temperature of a dish in some fancy restaurant, whose reputation rests upon getting this one dish right (which hence relies upon setting its temperature just right). Though the Bayesian setting of the problem is natural, the complicating factor arises when the Bayesian mechanics are actually applied:

$$\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{Y}_T) = \frac{\mathbb{P}(\mathbf{Y}_T \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\mathbb{P}(\mathbf{Y}_T)}. \tag{2.58}$$

Herein Lindsten et al. (2013) explain that the likelihood $\mathbb{P}(\mathbf{Y}_T \mid \boldsymbol{\theta})$ in "general cannot be computed in closed form" except when we are dealing with a linear Gaussian state-space models (Murphy, 2012, §18.1) as noted in the previous section on state inference. If we model the relationship between the likelihood and the latent states, then by marginalising the joint density w.r.t. the latent space

$$\mathbb{P}\left(\mathbf{Y}_T \mid \boldsymbol{\theta}\right) = \int \mathbb{P}\left(\mathbf{X}_T, \mathbf{Y}_T \mid \boldsymbol{\theta}\right) \mathrm{d}\mathbf{X}_T \tag{2.59}$$

$$= \int \cdots \int \mathbb{P}\left(x_1, \ldots, x_T, y_1, \ldots, y_T \mid \boldsymbol{\theta}\right) \mathrm{d}x_1 \cdots \mathrm{d}x_T \tag{2.60}$$

we arrive at the marginal. Equivalently we can express the same thing using the one-step predictive likelihood (Schön et al., 2015), in which case

$$\mathbb{P}(\mathbf{Y}_T \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} \mathbb{P}(y_t \mid y_{t-1}, \boldsymbol{\theta}) \tag{2.61}$$

where

$$\mathbb{P}(y_t \mid y_{t-1}, \boldsymbol{\theta}) = \int g(y_t \mid x_t, \boldsymbol{\theta})\, \mathbb{P}(x_t \mid \mathbf{Y}_{-t}, \boldsymbol{\theta})\, \mathrm{d}x_t \tag{2.62}$$

Unfortunately these expressions are of limited utility since this is usually a high-dimensional intractable integral. The solution to this problem is to target the joint state (auxiliary parameters) and parameters posterior in a process known as *data augmentation* (Lindsten et al., 2013; King, 2012) (also known as an auxiliary or latent variable approach). Under

this machinery the joint posterior of the model parameters and the auxiliary variables are formed via Bayes' theorem

$$\mathbb{P}(\mathbf{X}_T, \boldsymbol{\theta} \mid \mathbf{Y}_T) = \frac{\mathbb{P}(\mathbf{Y}_T \mid \mathbf{X}_T, \boldsymbol{\theta})\,\mathbb{P}(\mathbf{X}_T, \boldsymbol{\theta})}{\mathbb{P}(\mathbf{Y}_T)} \tag{2.63}$$

$$= \frac{\mathbb{P}(\mathbf{Y}_T \mid \mathbf{X}_T, \boldsymbol{\theta})\,\mathbb{P}(\mathbf{X}_T \mid \boldsymbol{\theta})\,\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\mathbf{Y}_T)} \tag{2.64}$$

where the system process and observation process are available to us, rendering some of the above components computable. The correct Bayesian way is then to marginalise the latent states, leaving only $\boldsymbol{\theta}$. This, however, means that we still need to deal with a nasty integral, requiring approximate inference methods. In some models, though, this marginalisation can be done exactly (Schön et al., 2015). It should come as no surprise to anyone that this class is the linear Gaussian SSM. But there is also the frequentist approach wherein we maximise the likelihood of the model parameters

$$\widehat{\boldsymbol{\theta}}_{\mathsf{ML}} = \arg\max_{\boldsymbol{\theta}} \mathbb{P}(\mathbf{Y}_T \mid \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log \mathbb{P}(\mathbf{Y}_T \mid \boldsymbol{\theta}) \tag{2.65}$$

this is the maximum likelihood formulation where the logarithm has been employed since it is typically more numerically stable (Särkkä, 2013). The maximum a posteriori point estimate can be found in a similar fashion.

2.4.2.1 Linear Gaussian state-space model

Having done the heavy lifting in the preceding section, we are now at liberty to introduce the densities which, in sum, yield the linear Gaussian state-space model (LGSSM). Hence, when

$$\mathbb{P}(\mathbf{X}_T, \mathbf{Y}_T \mid \boldsymbol{\theta}) = \mu(x_1 \mid \omega) \prod_{t=2}^{T} f(x_t \mid x_{t-1}, \phi) \prod_{t=2}^{T} g(y_t \mid x_t, \lambda) \tag{2.66}$$

then, in the LGSSM, the Kalman filter (forward recursions) (Kalman, 1960) prescribes a model structure

$$\mu(x_1 \mid \omega) = \mathcal{N}(x_1 \mid \mu_0 + b, V_0) \tag{2.67}$$

$$f(x_t \mid x_{t-1}, \phi) = \mathcal{N}(x_t \mid Ax_{t-1} + a, \Gamma) \tag{2.68}$$

$$g(y_t \mid x_t, \lambda) = \mathcal{N}(y_t \mid Cx_t + c, \Sigma) \tag{2.69}$$

for the univariate case, where $\omega \triangleq \{\mu_0, b, V_0\}$, $\phi \triangleq \{A, a, \Gamma\}$ and $\lambda \triangleq \{C, c, \Sigma\}$. In the common state-space formulation this is written as

$$x_1 = \mu_0 + \gamma, \ \gamma \sim \mathcal{N}(b, V_0) \tag{2.70}$$

$$\widehat{x}_t = A\widehat{x}_{t-1} + \alpha, \ \alpha \sim \mathcal{N}(a, \Gamma) \tag{2.71}$$

$$y_t = C\widehat{x}_t + \beta, \ \beta \sim \mathcal{N}(c, \Sigma). \tag{2.72}$$

The parameters are usually found using the maximum likelihood estimate via the expectation-maximisation algorithm (Bishop, 2006) – exact inference is only available when the model parameters are fixed, if not they have to be learned. We shall not labour on the exact minutiae of this, as it has been covered in much greater detail in (Bishop, 2006; Barber, 2012; Murphy, 2002). But on a high level we are continuously propagating the state and measurement updates of our model, whilst taking uncertainty into account.

Finally, the importance that should be derived from these equations is their origin, starting with the optimal Bayesian filtering and smoothing problem, where the exact smoothing recursions under the LGSSM are well described by Barber (2012, §24.4.2). Though this is a widely used model, the origin is more important, because the moment we replace the transition and emission distributions with almost any other density the problem becomes ill-posed (with a slight abuse of terminology) and hence intractable. There is no shortage of time-series phenomena that do not evolve according to LGSSM dynamics.

⋆ ⋆ ⋆

Though the LDS model is attractive through its simplicity and ease of computation, it is limited to a small set of problems. By definition non-linear variations in the state-space will be treated as noise in the model, which will lead to overly smoothed simulations. Hence, it is not suitable for human pose modelling where the dynamics are typically non-linear, with likelihood functions that are multi-modal and non-Gaussian (Fleet, 2011; Bishop, 2006). Indeed, as said, except for a few special cases, including HMMs, it is impossible (Andrieu et al., 2004) to obtain the optimal filter and likelihood in closed-form. This is why we need numerical approximation methods also known as approximate inference.

2.5 Approximate inference

Problems in human-pose estimation arise from kinematic singularities (Pons-Moll & Rosenhahn, 2011), depth and orientation ambiguities, this is in addition to problems with occlusions. This is the reason why the posterior density, as well as the observation process, is highly peaked and multi-modal (Pons-Moll & Rosenhahn, 2011). Instead of propagating a single pose hypothesis, as inference in LDS models does, we can approximate the likelihood of the image given the pose parameters, by propagating a set of particles from one time step to the next one. Hence, where the KF fails, the posterior can be approximated by particle filtering.

Sequential importance sampling (SIR) is the original particle filtering algorithm (Gordon et al., 1993), and is the application of importance sampling to approximate averages with respect to some intractable temporal distributions $\mathbb{P}(\cdot)$ (Barber, 2012). The basic idea is to approximate the belief state of the posterior distribution using a weighted set of K particles $\{w_n^{(k)}, \theta_n^{(k)}\}_{k=1}^K$ drawn from the posterior distribution (Bishop, 2006; Barber, 2012), such that $\mathbb{P}(\theta_{1:n} \mid y_{1:n}) \approx \sum_{k=1}^K \hat{w}_n^k \delta_{\theta_{1:n}^k}(\theta_{1:n})$, where w_n^k is the normalized weight of particle k at time n and $\delta_{\theta_{1:n}^k}(\theta_{1:n})$ are the point masses of the particles that approximate the distribution. The weights satisfy $0 \leq w_n^{(k)} \leq 1$ and $\sum_K w_n^{(k)} = 1$, and $\theta_n^{(k)}$ corresponds to one pose parameter configuration.

From this representation we can easily compute the marginal distribution over the most recent state, by simply ignoring the previous parts of the trajectory (Murphy, 2012). Hence, at each time step a new set of pose parameters θ_n^* is estimated (Bishop, 2006) by the mean of the weighted particles $\theta_n^* = \mathbb{E}[\theta_n] \simeq \sum_{k=1}^K w_n^{(k)} \theta_n^{(k)}$. Assuming a first-order Markov process, the posterior can be updated (Doucet et al., 2001b) recursively

$$\begin{aligned} \mathbb{P}(\theta_n \mid y_{1:n-1}) &= \int \mathbb{P}(\theta_n \mid \theta_{n-1})\, \mathbb{P}(\theta_{n-1} \mid y_{1:n-1})\, \mathrm{d}\theta_{n-1} \\ \mathbb{P}(\theta_n \mid y_{1:n}) &= \frac{\mathbb{P}(y_n \mid \theta_n)\, \mathbb{P}(\theta_n \mid y_{1:n-1})}{\int \mathbb{P}(y_n \mid \theta_n)\, \mathbb{P}(\theta_n \mid y_{1:n-1})\, \mathrm{d}\theta_n} \approx \sum_{k=1}^K \hat{w}_n^k \delta_{\theta_n^k}(\theta_n). \end{aligned} \tag{2.73}$$

Equation (2.73) computes the pose prediction from the previous posterior $\mathbb{P}(\theta_{n-1} \mid y_{1:n-1})$ propagated with the dynamical model $\mathbb{P}(\theta_n \mid \theta_{n-1})$, importance sampling (Pons-Moll & Rosenhahn, 2011, ch. 9) approximates the integral.

Particle filtering has been used extensively (see (Fleet, 2011, p. 3) for a comprehensive overview) for monocular tracking of 3D human pose. A recent paper (Gonczarek &

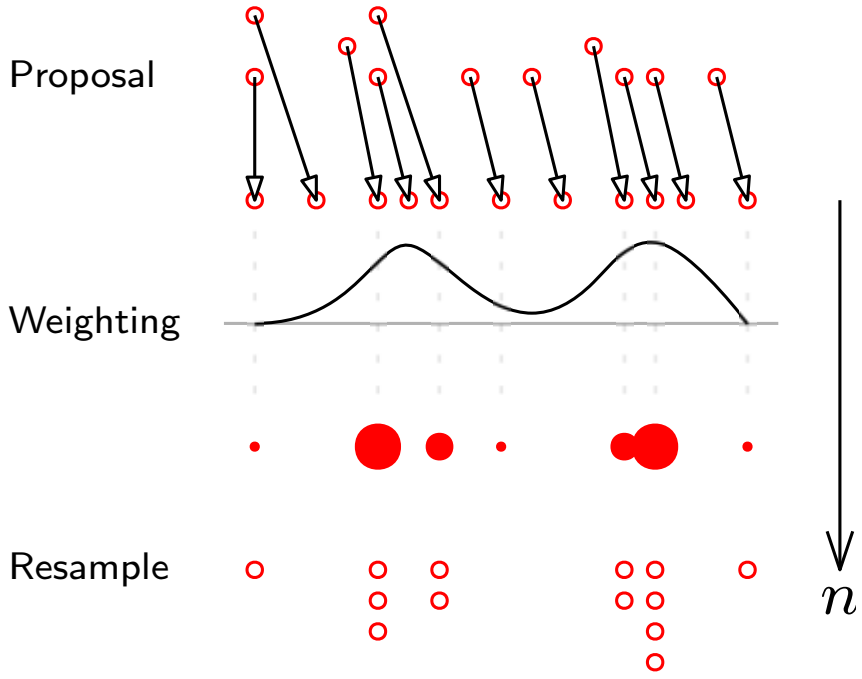


Figure 2.8: Illustration of particle filter: simulation based computation for recursive Bayesian inference. Illustrated is one time-step n of the particle filter, from the top we start with a proposal from the previous time-step (the previous step results from the resampling step, shown at the bottom). We represent the posterior as a mixture distribution (Bishop, 2006) of particles represented as circles ○, such that sampled particles have diameters proportional to their importance weight. New samples are then drawn from these weighted particles, and thus the simulation continues for a fixed number of cycles. Mechanically the steps are: particles are drawn from a proposal, re-weighted according to importance and then new particles are sampled for the next time-step.

Tomczak, 2014) outlined a fully Bayesian approach to articulated human motion tracking from video sequences, using a manifold regularised particle filter. The low-dimensional manifold is treated as a regulariser which restricts the space of all possible distributions to the space of distributions concentrated around the manifold itself. In later sections we shall consider other dimensionality reduction schemes used for motion modeling.

2.6 Automatic approximate inference

A model is a simplified representation of reality, and the simplifications are made to discard unnecessary detail and allow us to focus on the aspects of reality that we want to understand. Depending on the problem, it is important to assess the trade-offs between speed, accuracy and complexity of different models and algorithms and find a model that works best for that particular problem. Consequently, the pairing of a suitable inference scheme to a model is a notoriously difficult problem; no one method is likely to generalise

across the board (Wolpert & Macready, 1997). A probabilistic programming system allows us to 'quickly' and accurately iterate over models and inference methods, to find the most optimal pair conditioned on the problem domain.

For our problem domain, typically bespoke inference schemes have used to understand the latent space (Fox et al., 2007; Teh, 2011; Johnson & Willsky, 2010; Wallach et al., 2010), of the complex class of models under investigation. We propose instead the usage of probabilistic programming in lieu of the methods used hitherto. In this section, we will be presenting an extended exposition of the probabilistic programming paradigms, its vices and virtues, and how we exploit *them* to yield robust inference for our model family of choice.

To frame this discussion, consider first of all *what* probabilistic programming actually purports to be. The importance of this discussion cannot be understated since probabilistic programming is still not prevalent enough, in industry or academia, to allow it to be immediately identifiable to non-experts (compared to e.g. neural-nets, the importance of which has not escaped anyone, intentionally or not). Hence, we shall refer to the inestimable Pilon (2015) and his discussion of the probabilistic programming insights of Beau Cronin, where the latter frames the subject as so

> Another way of thinking about this: unlike a traditional program, which only runs in the forward directions, a probabilistic program is run in both the forward and backward direction. It runs forward to compute the consequences of the assumptions it contains about the world (i.e., the model space it represents), but it also runs backward from the data to constrain the possible explanations. In practice, many probabilistic programming systems will cleverly interleave these forward and backward operations to efficiently home in on the best explanations.
>
> –Beau Cronin

A formal qualification follows in definition 2.6.1.

2.6.1 Probabilistic programming systems

Probabilistic programs are regular programs extended by two constructs (Gordon et al., 2014):

I) The ability to draw random values from probability distributions; and

II) The ability to condition probability distributions on values computed in the programs.

A probabilistic programming system (PPS) unifies techniques for formal description of computation with the representation and use of uncertain knowledge. A PPS' main advantage is in separating the modelling and the inference problems, which allows us to focus on the on the former without worrying about the latter. Recently PPS languages have seen interest from fields as diverse as artificial intelligence, cognitive science as well as natural languages communities Goodman & Stuhlmüller (2014). Tolpin et al. (2015) formalises the concept:

Definition 2.6.1. (Probabilistic program). A probabilistic program is a stateful[5] deterministic computation $\mathcal{P}$. That computation $\mathcal{P}$ has the following properties:

- When commencing the computation, the computation does not expect any arguments
- On every call i, $\mathcal{P}$ returns either a distribution (F_i), a distribution and a value $(G_i, \mathbf{x}_i)$, a value $(\mathbf{z}_i)$ or a termination
- Upon returning a distribution, $\mathcal{P}$ expects a sample $\mathbf{x}$ drawn from the distribution as the argument to continue
- Upon returning a pair $(G, \mathbf{x})$ or a value $\mathbf{z}$, $\mathcal{P}$ is invoked again without any arguments
- Upon returning a termination, $\mathcal{P}$ terminates.

A program is run by calling $\mathcal{P}$ repeatedly until termination.

Every run of the program implicitly produces a sequence of pairs $(F_i, \mathbf{x}_i)$ of distributions and values of latent random variables (Tolpin et al., 2015). That sequence is called a *trace* and is denoted by $\mathbf{X}$. The probability of a trace is proportional to the product of the probability of all random choices $\mathbf{X}$ and the likelihood of all observations $\mathbf{Y}$

$$\mathbb{P}_{\mathcal{P}}(\mathbf{X} \mid \mathbf{Y}) \propto \prod_{i=1}^{|\mathbf{X}|} \mathbb{P}_{F_i}(\mathbf{x}_i) \prod_{j=1}^{|\mathbf{Y}|} \mathbb{P}_{G_j}(\mathbf{y}_i). \tag{2.74}$$

The objective of inference in probabilistic program $\mathcal{P}$ is to discover the distribution of the program output. *Observe*: we have employed the notational convention of the PPS field in this definition, and used $\mathbf{x}$ for the latent variable. We will use this convention when discussing algorithms in the preliminary section, but θ otherwise, mainly when we talk about Bayesian nonparametric state-space model.

[5]Stateful computation means that the model of computation has got a memory storage, and it uses this information to make computations.

Throughout this chapter, our PPS choice will be *Anglican* (Wood et al., 2014). It is an open source, compiled probabilistic programming language integrated with Clojure, which in itself is a general purpose functional programming language that just-in-time compiles to a Java virtual machine (JVM)[6]. In other words, Anglican is a subset of Clojure, extended with a few special forms that make it a probabilistic programming language. Anglican, like all probabilistic programming languages, differs substantially from traditional programming. Probabilistic programs are written with parts not fixed in advance that instead take values generated at runtime by random sampling procedures. Inference in probabilistic programming characterises the conditional distribution of such variables given observed data assumed to have been generated by executing the probabilistic program Wood et al. (2014). Anglican has implementations of several importance sampling based methods such as sequential Monte Carlo (SMC) and particle Markov chain Monte Carlo sampling procedures (Andrieu et al., 2010).

For demonstration purposes, consider the generative hidden Markov model. Using this, we briefly discuss the idea behind SMC inference in a probabilistic programming context. Consider the generative model $\mathbb{P}(\theta_{1:T}, y_{1:T})$ with hidden variables $\theta_{1:T}$ and observations $y_{1:T}$. In PPS, we let the observing random variable y_t be the value of the t^{th} observation, and the hidden variables $\theta_{1:t}$ be the execution trace before this observation (see fig. 2.9). The goal of SMC inference in Anglican is to approximate

$$\mathbb{P}(\theta_{1:T} \mid y_{1:T}) = \frac{\mathbb{P}(\theta_{1:t}, y_{1:t})}{\mathbb{P}(y_{1:t})} \; \forall t \in \{1, \ldots, T\}. \tag{2.75}$$

The weighted approximation to $\mathbb{P}(\theta_{1:T} \mid y_{1:T})$, is achieved by generating a set of particles (indexed by p) and importance weights at each time step: $\{(\theta_{1:t}^{(p)}, w_t^{(p)})\}_{p=1}^{P}$. The approximation to the target distribution is then given by

$$\mathbb{P}(\theta_{1:t} \mid y_{1:t}) \approx \sum_{p=1}^{P} w_t^{(p)} \delta_{\theta_{1:t}^{(p)}}(\theta_{1:t}). \tag{2.76}$$

At each time step particles are generated using a chosen proposal density $q_t(\theta_{1:t} \mid \theta_{1:t-1})$. This is used to propose new particles, given the set of particles $\{\bar{\theta}_{1:t-1}^{(p)}\}$ re-sampled from the previous $t-1$ steps of the SMC estimate of $\mathbb{P}(\theta_{1:t-1} \mid y_{1:t-1})$.

[6]The JVM is an abstraction layer between a Java application and the underlying platform. As the name implies, the JVM acts as a virtual machine or processor. To the bytecodes comprising the program, they are communicating with a physical machine; however, they are actually interacting with the JVM.

The weights of the corresponding particles are given as follows:

$$W_t^{(p)} = \frac{\mathbb{P}\left(\theta_{1:t}^{(p)} \mid \bar{\theta}_{1:t-1}^{(p)}\right) \mathbb{P}\left(y_t \mid \theta_{1:t}^{(p)}\right)}{q\left(\theta_{1:t}^{(p)} \mid \bar{\theta}_{1:t-1}^{(p)}\right)}, \tag{2.77}$$

and $w_t^{(p)} = W_t^{(p)} \left(\sum_{i=1}^{P} W_t^{(p)}\right)^{-1}$ for $p \in \{1, \ldots, P\}$. Finally, for each t in the proposal, re-sampling and re-weighting steps are iterated. This gives us a particle estimate of our target posterior along with an estimate of the marginal likelihood $\mathbb{P}(y_{1:T})$.

We now give a more detailed explanation for the understanding of the algorithms underpinning Anglican's inference engine. Further, it is common in the approximate inference literature to use notation $\mathbf{x}$ as the latent variable, we shall do so too in the coming discussion, in order to keep ourselves aligned with convention.

2.6.2 Markov chain Monte Carlo

Our principal concern, typically, and primary reason for using MCMC methods, is the difficulty of performing inference on high-dimensional distributions. To formalise our approach with respect to MCMC methods, assume that there is some multivariate distribution such that

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \mathbb{P}^*(\mathbf{x}) \tag{2.78}$$

where $\mathbf{x} \in \mathbb{R}^N$ and $N \gg 1$, $\mathbb{P}^*(\mathbf{x})$ is the unnormalised distribution and $Z = \int_{\mathbf{x}} \mathbb{P}^*(\mathbf{x})$ is the normalisation constant (which is intractable) (Barber, 2012). We can evaluate the unnormalised distribution for any state, but not the normalised because of the intractability of Z. To this end then, the main idea behind MCMC sampling is to sample not directly from $\mathbb{P}(\mathbf{x})$, but from a different distribution such that, in the *limit of a large number of samples* (illustrated by considering a simple sampling procedure from $\mathcal{N}(1, 1)$ as shown in

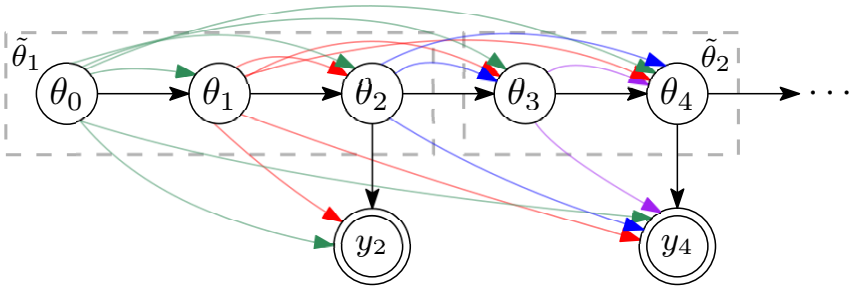
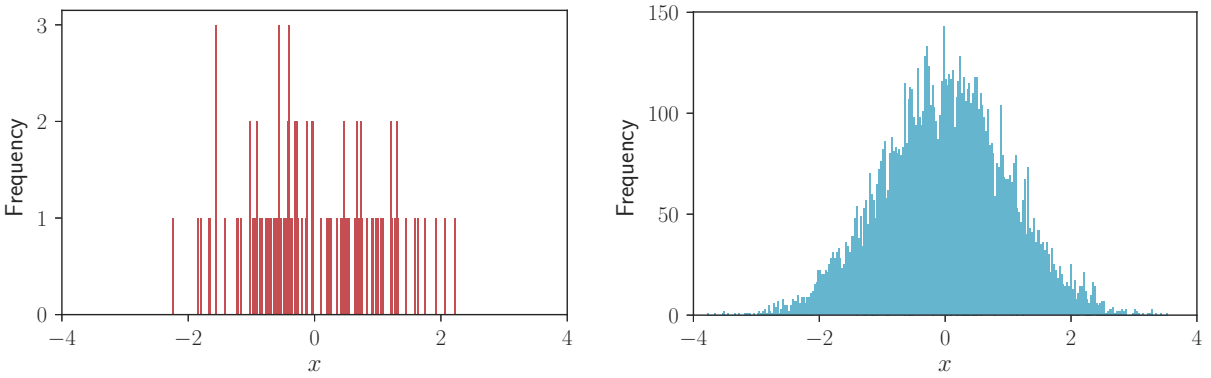


Figure 2.9: Model traces illustration, adapted from (Wood, 2015). The execution trace is bounded by the dashed box, wherein $\tilde{\theta}_1 = \theta_0 \times \theta_1 \times \theta_2$ for example.

figure fig. 2.10), effectively the samples will be from $\mathbb{P}(\mathbf{x})$. Secondly, as the name implies, MCMC forms a Markov chain of dependent samples in order to simulate the expectation of a statistic in a complex model (such as a state-space model). Then, successive random selections form a Markov chain, which has an stationary transition density independent of the time-step. Therein lies the power of inference by simulation.



(a) 100 samples. **(b)** 10,000 samples.

Figure 2.10: As the number of samples increases, the relative frequency of the samples tends to the distribution $\mathbb{P}(x) = \mathcal{N}(1, 1)$ i.e. the univariate normal distribution.

The most general algorithm in the MCMC family is the Metropolis Hastings (MH) method (Chib & Greenberg, 1995) - see algorithm 2. Though it is an MCMC technique within the realm in which we have discussed, it is worth noting that it is not the same as e.g. Gibbs sampling. In fact Gibbs sampling is a special case of MH. This is true because in Gibbs sampling the proposal distribution is given by $\mathbb{Q}(\mathbf{x}_t \mid \mathbf{x}_{-t})$, which, when applied to the Metropolis-Hastings acceptance criterion, reduces to one – hence we always accept under this scheme. Conversely, for completeness, importance and rejection sampling methods are not MCMC algorithms because they are not based on Markov chains. Indeed, importance sampling does not produce a sample from the target distribution, but only importance weights, which are used in Monte Carlo approximations of distributions related to the target.

Thus, more generally speaking, at each step of MH we propose to move from the current state $\mathbf{x}$ to a new state $\mathbf{x}'$ with a probability given by $\mathbb{Q}(\mathbf{x}' \mid \mathbf{x})$, where $\mathbb{Q}$ is the proposal distribution (Murphy, 2012). The user is free to choose any $\mathbb{Q}$ subject to some conditions. Here, it is worthwhile to note that for every distribution $\mathbb{P}(\mathbf{x})$ there will be more than one transition $\mathbb{Q}$ with $\mathbb{P}(\mathbf{x})$ as its stationary distribution, hence the large number of algorithms

in the MCMC family, each corresponding to a specific instance of $\mathbb{Q}$. Murphy (2012) shows that having proposed a move to $\mathbf{x}'$ this move is either accepted or rejected according to the Metropolis-Hastings acceptance criterion (Bishop, 2006)

$$a = \min\left(1, \frac{\mathbb{P}(\mathbf{x}')\,\mathbb{Q}(\mathbf{x}' \mid \mathbf{x})}{\mathbb{P}(\mathbf{x})\,\mathbb{Q}(\mathbf{x} \mid \mathbf{x}')}\right). \tag{2.79}$$

If the proposal is accepted, the new state is $\mathbf{x}'$, otherwise the new state is the same as the current state $\mathbf{x}$. The power of MH can be realised by noting that we only need to know the target density up to a normalising constant, this is because if we substitute (2.78) into (2.79) the normalisation factors cancel. Thus we can sample from the unnormalised distribution $\mathbb{P}^*(\mathbf{x})$ even for very high-dimensional distributions, the full procedure can be found in algorithm 2 (Barber, 2012) in appendix A.

Alternative Monte Carlo methods, called particle filters, can be more efficient for inference about the unobserved process given known parameter values. We continue with a brief review, since they are found widespread use in this thesis.

2.6.3 Particle filters

Particle filtering has appeared in the literature under various names including sequential Monte Carlo (SMC), the bootstrap filter and the condensation algorithm (Blake & Isard, 1997). They are all Monte Carlo algorithms that can be used to approximate posterior distributions for state-space models. Within we outline a basic particle filter with a view to their use within particle MCMC algorithms, which is introduced in section §2.6.4.

Sequential importance sampling is the basic algorithm underlying simulation based (Murphy, 2012) recursive Bayesian filtering (particle filtering). Thus, consider distributions with a hidden Markov structure (Barber, 2012) such as the HMM, where $\mathbf{x}_{1:T}$ are the latent variables and $\mathbf{y}_{1:T}$ the observations, with a joint distribution given by

$$\mathbb{P}(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = \mathbb{P}(\mathbf{x}_1) \prod_{t=2}^{T} \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \prod_{t=1}^{T} \mathbb{P}(\mathbf{y}_t \mid \mathbf{x}_t). \tag{2.80}$$

Our interest lies in drawing sample paths $\mathbf{x}_{1:T}$ given the observations, where the basic idea is to approximate the belief state (which is to say the entire state trajectory) using a weighted set of particles. Please refer §2.6.1 for the full details, as it was outlined there first.

Although particle filters can be more efficient for inference than MCMC methods with respect to the unobserved process with known parameters values, they struggle when dealing with unknown parameter values (for a more thorough overview of particle filters see e.g. Kitagawa & Gersch (2012); Särkkä (2013); Doucet et al. (2001a)). A natural extension then of both MCMC methods and particle filters, is the marriage of the two, yielding particle MCMC (PMCMC) (Doucet & Johansen, 2009; Andrieu et al., 2010).

2.6.4 Particle Markov chain Monte Carlo

The PMCMC approach is to embed a particle filter within an MCMC algorithm. The particle filter will then update the unobserved process given a specific value for the parameters, and MCMC moves will be used to update the parameter values (Wood et al., 2014; Fearnhead & Meligkotsidou, 2014). To put it simply, PMCMC is a Metropolis-Hastings sampling scheme with SMC updates. Furthermore, there are three generic implementations of PMCMC; particle independent Metropolis Hastings, particle marginal Metropolis Hastings (PMMH) and particle Gibbs (PG). For a brief synopsis of each see (Fearnhead, 2012), for a more robust and in-depth treatment see (Andrieu et al., 2010).

As noted earlier, an alternative approach to parameter inference by particle filters is to use MCMC updates within the particle filter algorithm to generate new parameter values. Unfortunately, it is not as easy as this due to inherent problems with the SMC algorithm. First, the particle filter fails after a few steps because most of the particles will have negligible weight. This is called the degeneracy problem, and occurs because we are sampling in a high-dimensional space (Murphy, 2012). This is typically countered by adding a resampling step and using a good proposal distribution (neither of which are trivial solutions).

Going a bit further, the commanding principle is that the particle weight represents the relative probability with respect to the other particles. In the resampling step, we draw from the set of particles such that for each particle, the normalised weight times the number of particles represents the number of times that particle is drawn on average. There are any number of ways of doing this properly, for example Hol et al. (2006) compare four frequently encountered resampling algorithms for particle filters (they find that systematic

resampling works best). Second, whilst resampling helps with degeneracy it introduces problems of its own; since the particles with high weight will be selected many times, there is a loss of diversity amongst the population. This is known as sample impoverishment. For the standard Gibbs sampler these problems become especially acute.

The idea behind the basic Gibbs sampler is to sample iteratively from $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})$ and $\mathbb{P}(\mathbf{x}_{1:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$ where the state-space model is parametrised by $\boldsymbol{\theta}$. Hence, we do not require the specification of a proposal density as in algorithm 2 and indeed for PMMH, because it it often possible to sample $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})$. This is typically *impossible* for $\mathbb{P}(\mathbf{x}_{1:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$ − however we can use an approximation for this update, known as the conditional SMC update (Andrieu et al., 2010) − see algorithm 3, which allows for SMC to be used as a proposal distribution in a Gibbs sampling algorithm (Neiswanger et al., 2014). Conditional SMC's power is that it ensures that a given particle with a specific ancestral lineage will survive all the resampling steps, where the remaining particles are generated as usual. Before presenting the conditional SMC update for state-space models, the notion of ancestral lineages is best explained through a simple example shown in example 2.2.

Example 2.2: Ancestral lineage

Figure 2.11 shows ancestral lineages generated by a conditional SMC algorithm. One of the lighter paths is e.g.

$$\mathbf{x}_{1:6}^4 = (\mathbf{x}_1^3, \mathbf{x}_2^3, \mathbf{x}_3^5, \mathbf{x}_4^5, \mathbf{x}_5^5, \mathbf{x}_6^4)$$

i.e. the bottom one, which has an ancestral lineage given by

$$B_{1:6}^4 = (3, 3, 5, 5, 5, 4)$$

and for emphasis; *these are the states.*

The lineage, $B_{1:T}^{(l)}$, of a particle is then defined recursively such that $B_T^{(l)} = l$, with the backward recursive relation (Andrieu et al., 2010), given by $B_t^{(l)} \triangleq A_t^{B_{t+1}^{(l)}}$. As such A_t^l represents the index of the parent at time t of particle $\mathbf{x}_{1:t}^{(l)}, \forall t \in \{2, \ldots, T\}$.

As noted the difficulty lies in sampling $\mathbb{P}(\mathbf{x}_{1:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$, and it is this step we are approximating in algorithm 3 in chapter A. The full particle Gibbs algorithm is presented in algorithm 4 in chapter A.

The idea of particle Gibbs is to use a particle filter to approximate $\mathbb{P}(\mathbf{x}_{1:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$ as shown in algorithm 3. Hence, informally, if the current value for $\mathbf{x}_{1:T}$ is labelled as $\mathbf{x}_{1:T}^*$, then we can approximate the update step $\mathbb{P}(\mathbf{x}_{1:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$ by implementing a particle filter

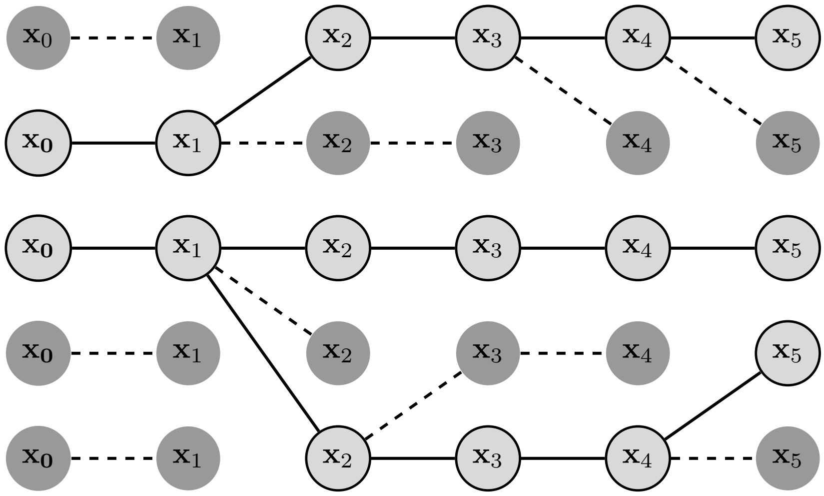


Figure 2.11: Example of $L-1=4$ ancestral lineages generated by a conditional SMC algorithm for $L=5$ particles and time $t=1,\ldots,6$ where $\mathbf{x}_t^{(l)}$ is particle l at time t. Each level corresponds to a state sequence.

which conditions on one of the particles $\mathbf{x}^*_{1:T}$ at time T (Fearnhead & Meligkotsidou, 2014). Thus we sample one of the particles at time T from the conditioned particle filter (i.e. algorithm 3), and update $\mathbf{x}_{1:T}$ to the value of this particle. The particle Gibbs algorithm can be shown to satisfy detailed-balance (for details see Andrieu et al. (2010)), is ergodic under mild assumptions, and will, given sufficient amount of time, admit the stationary distribution.

2.6.5 Bayesian optimisation for probabilistic programs

In some experiments, we are required to choose domain-specific model parameters, parameters which are typically only known by experts in that field. By seeking to make our models and methodology as general as possible we want to extract optimal model parameters from the observations themselves. However, in general (though exceptions exist see, e.g. Rainforth et al. (2016)), PPS inference engines currently do not provide such optimisation, particularly if the objective function is expensive to evaluate, as is typically the case when it takes the form of an intractable integral. This is especially true in our case since our objective functions are full nonparametric Bayesian state-space models. Hence to overcome the drawbacks of probabilistic programming, we turn to Bayesian optimisation (BO).

The appropriateness of the hyperparameters $\boldsymbol{\theta}$ can be described by the marginal likelihood. To this end, we use the aforementioned particle-based inference algorithms in Anglican to

provide us with a noisy estimates of $\mathbb{P}(\mathbf{y}_{1:T})$ (an intractable integral). This is expensive since inference must be performed on large datasets, hence it fits well with the Bayesian optimisation framework which allows us to find the global maximum of some expensive black-box function $f : \mathbb{R}^d \to \mathbb{R}$ (for further details see the excellent papers by Snoek et al. (2012) and Shahriari et al. (2016)). This is true for probabilistic programming systems, where the objective function is often expensive to evaluate as is typically takes the form of the log marginal likelihood. Specifically in probabilistic programs a program defines a joint density $\mathbb{P}(\mathbf{x}_{1:T}, \mathbf{x}_{1:T} \mid \boldsymbol{\theta})$. Rainforth et al. (2016) explain that the aim is to optimise a subset of the variables $\boldsymbol{\theta}$ whilst also integrating out the latent variables. Formally, Bayesian optimsation seeks to find the global maximum, on a d-dimensional space with bounds B:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta} \in B \subset \mathbb{R}^d}{\arg\max}\, f(\boldsymbol{\theta}) && \text{Standard Bayesian optimisation} \\
&= \underset{\boldsymbol{\theta} \in B \subset \mathbb{R}^d}{\arg\max}\, \mathbb{P}(\boldsymbol{\theta} \mid \mathbf{y}_{1:T}) && \text{PPS objective function} \\
&= \underset{\boldsymbol{\theta} \in B \subset \mathbb{R}^d}{\arg\max}\, \mathbb{P}(\boldsymbol{\theta}, \mathbf{y}_{1:T}) \\
&= \underset{\boldsymbol{\theta} \in B \subset \mathbb{R}^d}{\arg\max} \int \mathbb{P}(\boldsymbol{\theta}, \mathbf{y}_{1:T}, \mathbf{x}_{1:T})\, \mathrm{d}\mathbf{x}_{1:T} && \text{BO for probabilistic programs} \qquad (2.81)
\end{aligned}$$

where we may only be able to evaluate f noisily. Bayesian optimisation models the objective function as a random function and uses this model to determine informative sample locations. A popular approach is to model the underlying function as a Gaussian process (see §2.7.1). We incorporate our beliefs about f by placing a prior measure over the space of such possible objectives. By conditioning f on the available data $\mathcal{D}_t = \{\boldsymbol{\theta}_t, \mathbf{y}_t\}_{t=1}^T$, the posterior over functions $\mathbb{P}(f \mid \mathcal{D}_t)$ is retrieved. This allows estimation of the expected value and uncertainty in $f(\boldsymbol{\theta})$, $\forall\, \boldsymbol{\theta} \in \mathbb{R}^d$. Bayesian optimisation then calculates this posterior and uses it to define an acquisition function $a(\cdot)$, which assigns an expected utility to evaluating f at particular $\boldsymbol{\theta}$, based on the trade off between exploration and exploitation in finding the maximum. Each evaluation yields an additional training point $(\boldsymbol{\theta}_t, \mathbf{y}_t)$. After updating the GP with the new observation, BO repeats the cycle until convergence or a budget on the total number of evaluations is exhausted. By interleaving optimisation of the acquisition function, evaluating f at the suggested point and updating the surrogate, BO forms an efficient global optimisation algorithm, in the required number of function evaluations, whilst naturally dealing with noise in the outputs (Rainforth et al., 2016).

2.7 Bayesian nonparametrics

Bayesian nonparametrics (BNP) is an area of statistics that allows model to be more flexible, by making their priors unbounded, or in other words; turning priors into infinite-dimensional objects in which our objects of desire live. A nonparametric Bayesian model is a Bayesian model whose parameter space has infinite dimension. To define a nonparametric Bayesian model, we have to define a prior on an infinite-dimensional space (Teh et al., 2006).

We shall not provide a thorough exegesis of all of BNP, merely the objects and tools which we will be using for our contributions. Chief amongst these is the Dirichlet process (DP) and the Gaussian process, but we also touch upon the Pitman-Yor process. We shall also consider the hierarchical DP as well as a discussion into exchangeability and the validity of our analysis. For completeness we echo the words of Wallach et al. (2010, §1), which goes some way in summarising our choice of processes; any model based on the Dirichlet or Pitman-Yor processes induces a posterior distribution which provides a partition of the data into clusters, without requiring that the number of clusters be pre-specified in advance. But they go on to to highlight that most sequential processes will fail to produce a partition that is exchangeable. Most processes that is, except the Dirichlet and Pitman-Yor processes(Wallach et al., 2010, §2.3).

But we start with the GP.

2.7.1 Gaussian processes

This section will provide a synopsis of the Gaussian processes (GP), primarily to inform §2.3 and consequently some of the experiments we undertake whilst working with Bayesian nonparametric state-space models later on in the thesis.

Definition 2.7.1. (Gaussian process). A Gaussian process (GP) is an infinite collection of random variables, any finite subset of which have a Gaussian distribution. Further, a GP defines a prior over functions, a prior which can be used utilised to find the posterior over the same functions, once the GP has been made privy to a finite set of observations.

The GP specifies that random functions $f(\cdot)$ can be drawn from the former, such that the values $f(\mathbf{X})$ at points $\mathbf{X} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are drawn from a multivariate normal distribution as

$$f(\mathbf{X}) \sim \mathcal{N}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')) \tag{2.82}$$

where $m(\cdot)$ is the mean function and $m(\mathbf{X}) = \mathbb{E}[f(\mathbf{X})]$ is the mean function and $k(\mathbf{X}, \mathbf{X}') = \mathbb{E}[(f(\mathbf{X}) - m(\mathbf{X}))(f(\mathbf{X}') - m(\mathbf{X}'))]$ is the covariance function.

Throughout we assume we have a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$ of n input predictor vectors, aggregated in design matrix $\mathbf{X}$ of size $n \times D$ in which $\mathbf{x}_i \in \mathbb{R}^D$, where $\mathbf{X} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\mathsf{T}$ (where each row contains one example) and the value of the functions at the given input values $\mathbf{X}$ is given by $\mathbf{f}(\cdot)$, with corresponding target values $y_i \in \mathbb{R}$, aggregated in a target vector $\mathbf{y} = [y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)]^\mathsf{T}$. If we let $\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\mathsf{T}$ then GPs provide a flexible class of models in which any finite-dimensional marginal and joint distribution is Gaussian, which is to say if

$$\mathbf{f}(\mathbf{X}) \sim \mathcal{GP}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X})) \tag{2.83}$$

then

$$\mathbb{P}(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)) = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ m(\mathbf{x}_2) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \ldots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}\right).$$

The $n \times 1$ sized mean vector $\mathbf{m}_i = m(\mathbf{x}_i)$ is specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$, which represents an initial guess at the regression function $f(\cdot)$. The covariance function $k(\mathbf{x}, \mathbf{x}') = \mathsf{cov}(f(\mathbf{x}), f(\mathbf{x}'))$ specifies the covariance between the process at any two points, resulting in a $n \times n$ covariance matrix $K_{ij} = k_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j \mid \boldsymbol{\theta})$. The covariance function and its hyperparameters $\boldsymbol{\theta}$, control the smoothness of realisations from the GP and the degree of shrinkage towards the mean (Gelman et al., 2014). For example, as noted by Rasmussen & Williams (2006), a popular choice for the covariance function, is the squared exponential covariance function

$$k_{\mathsf{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2l^2}\right) \tag{2.84}$$

in which the hyperparameter is given by the length-scale l and $||\cdot||$ is the Euclidean norm. Different covariance functions can be used to add structural prior assumptions like smoothness, non-stationarity, periodicity, and multi-scale or hierarchical structures

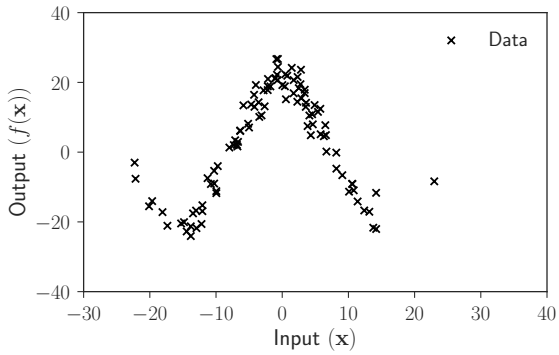
(Gelman et al., 2014). Sums and products of Gaussian processes are also GPs which allows easy combination of different covariance functions.

At the time of writing there is but one opus magnum on the subject of Gaussian processes, so we shall be referencing the book by Rasmussen & Williams (2006) extensively in our perusal of the topic. More importantly, they describe in detail the methodologies, which we exploit in this thesis. As such, for extensive theory on the subject, we refer the reader to them. We shall drive this section with an example courtesy of eq. (2.85)

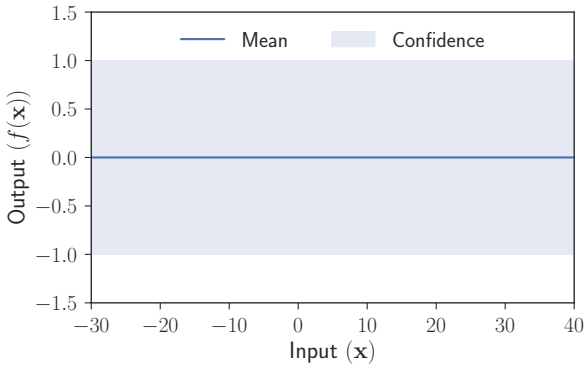
$$f(\mathbf{X}) = 20\cos\left(\frac{\mathbf{X}}{5}\right) + \mathcal{N}(\mathbf{0}, 10 \cdot \mathbf{I}) \tag{2.85}$$

From eq. (2.85) we have drawn a number of samples and applied to them a GP with the aim of locally estimating the underlying dynamics (i.e. without the noise). For this we use a Matérn kernel (Williams & Rasmussen, 2006, p. 84). The choice of this kernel is deliberate and requires further discussion. Typically, in GP modelling the first choice of kernel is almost always the squared exponential. But as the choice of kernel determines almost all the generalisation properties of the GP, in this instance, the SE kernel is unsuitable. The reason being that it is "too smooth" (Rasmussen & Williams, 2006). In fact the SE covariance function is infinitely differentiable, rendering it strictly smooth by definition, and hence imposes dynamics on the Gram matrix (another name for the covariance matrix) which are unsuitable for many natural phenomena (Rasmussen & Williams, 2006). Consequently our choice of the Matérn kernel which does not have the same smoothness assumptions.

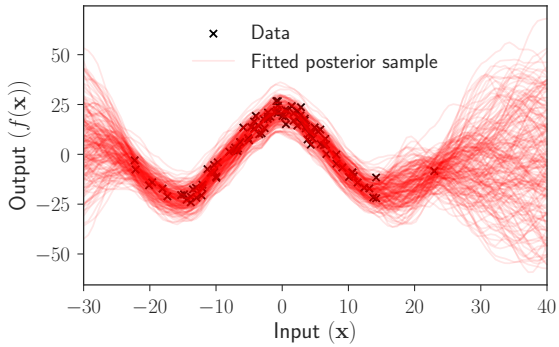
We continue to use the GP as a prior and update it using the available data to obtain a posterior GP which we can then use for prediction – as demonstrated in fig. 2.12. We train the model by minimising the negative log marginal likelihood w.r.t. to the hyperparameters and noise level (Rasmussen & Williams, 2006), using gradient based optimisation – see (Rasmussen & Williams, 2006, §2.2) for further details on gradient descent for this model class. Where the training seeks to find GP hyperparameters for the kernel and the mean function. Learning in the GP setting is discussed further in §2.7.1.2.



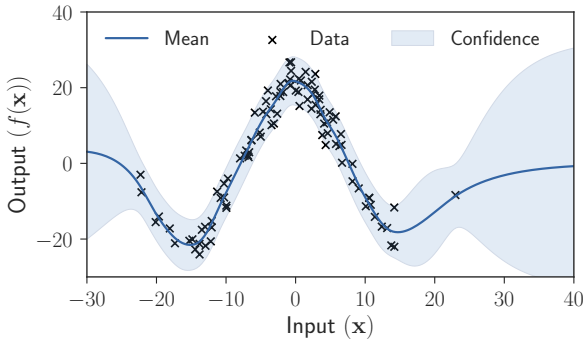
(a) Samples from eq. (2.85).



(b) Prior.



(c) Samples from the posterior.



(d) Confidence depicted at twice the standard deviation around the predictive mean.

Figure 2.12: A graphical illustration of the GPR methodology. Figure 2.12a shows observations from eq. (2.85) followed by fig. 2.12b depicting the prior Matérn kernel prior and fig. 2.12c show samples from the posterior. Finally fig. 2.12d shows the posterior with confidence bounds around the predictive mean.

An alternate way of understanding GP is by considering its graphical model, where we can consider the interconnected nature of the Gaussian field, by understanding a subgraph of the graphical model in fig. 2.13. The Gaussian finite field is a complete graph K_n, where a complete graph is a simple undirected graph in which every pair of distinct vertices n is connected by a unique edge. For our purposes, the edge strengths represent the covariance terms $\sum_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ (Murphy, 2012, §15.1). We shall depict the K_3 subgraph from fig. 2.13 denoted by blue edges. Then, by invoking the third dimension, the K_3 complete graph is more clearly displayed in fig. 2.14.

We can add a test point $\mathbf{x}_*$ to the subgraph in fig. 2.14 and thus attain the GP regression model for K_4. Then, because of the marginalisation property of GPs, when adding more nodes to the graph, the distribution on the other nodes remains invariant to change (Rasmussen & Williams, 2006). Thus, the learning problem boils down to finding the edge

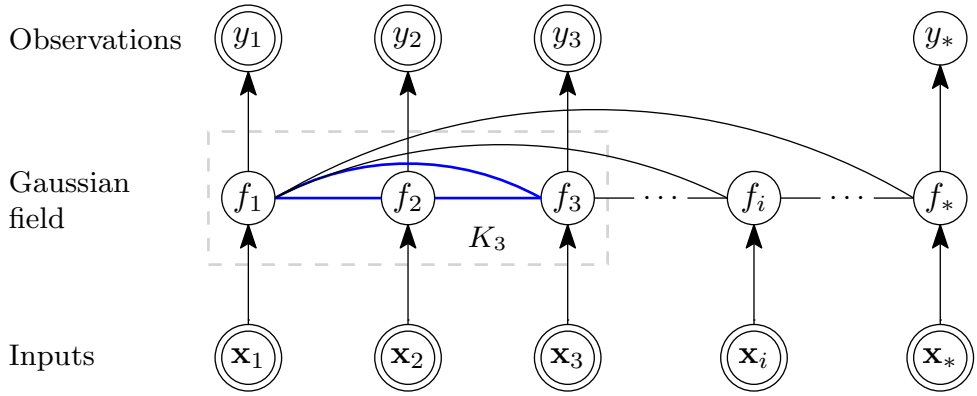
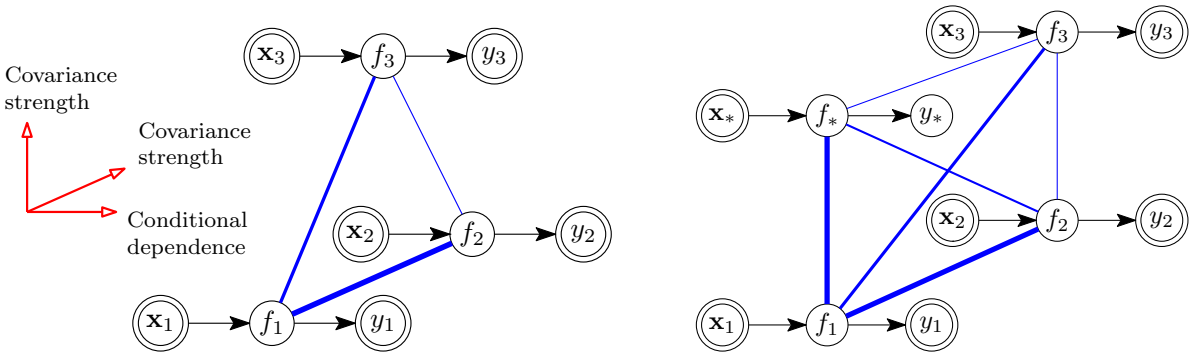


Figure 2.13: Gaussian process graphical model. All hidden nodes $f_i = f(\mathbf{x}_i)$ (for $i = 1, 2, 3$) are interconnected by directed edges forming the Gaussian process. For brevity, and to avoid clutter, we only show conditional dependencies of f_1 s.t. $f_1 \mid f_2, f_3, f_i, f_*$. Adapted from (Rasmussen & Williams, 2006, p. 17). Special care should be taken regarding the stochastic nature of the observed nodes. Specifically note that the inputs are deterministic i.e. under this model, but the observations are stochastic per design through the Gaussian process, as they are sampled.



(a) A Gaussian process regression graphical model, with three training points and zero test points. The latent nodes are fully connected as per definition in the K_3 subgraph.

(b) A Gaussian process regression graphical model with three training points and one test point.

Figure 2.14: Oblique projection of Gaussian process K-complete (nomenclature borrowed from graph theory) sub-graphical models with and without test points. Note that edges carry different weights, corresponding to different strengths of covariance. The 'vertical' and 'horizontal' strengths (as noted by coordinate system) are in reference to other nodes that lie in the same plane. The final, orthogonal axis, of the coordinate system demonstrates conditional dependencies or alternatively, information flow, in the sub-graphs.

weights, which most faithfully represent the information dependencies between inputs and predictors, which in consequence, means finding the hyperparameters that achieve that goal. This rests upon the notion that in GP modelling 'kernel is king' – i.e. that the kernel determines almost all the generalisation properties of a model. Hence, conditioned on the assumption that the kernel with best modelling capacity has been selected, the parameters of that same covariance function are tuned (via gradient descent) to maximise the information yield (measured through the marginal log likelihood) of the model (under

that kernel). Pictorially this is equivalent to tuning the weight of the edges, connecting the latent nodes in fig. 2.14.

2.7.1.1 Multivariate Gaussian process regression

Our intent is to learn a function which maps gait-cycle percentage (i.e. fraction of the stride period completed) and speed, to a locomotion-variable (e.g. joint-angle) trajectory. One way to approach the multivariate function learning problem is to place a prior distribution on the regression function using Gaussian processes (GP) (Rasmussen & Williams, 2006). Usage of GPs for regression is also known as *spatial kriging* or just *kriging* (Krige, 1951; Kleijnen, 2009) – a method originally developed in spatial statistics by Krige (1951).

Ultimately our goal is to infer target values at n_* test points not in the design matrix $\mathbf{X}$. Equivalently, we would like to estimate or predict the value of y_* at novel inputs $\mathbf{x}_* \notin \mathbf{X}$. We do this by assuming that y_* also originates from a multivariate Gaussian likelihood. Consequently, given a Gaussian observation model: $y(\mathbf{x}) \mid f(\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$; its predictive distribution (Rasmussen & Williams, 2006, §2.2) over test points $\mathbf{X}_*$ has the form

$$\mathbb{P}(\mathbf{f}(\mathbf{X}_*) \mid \mathbf{X}, \mathbf{y}, \mathbf{X}_*, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\bar{\mathbf{f}}_*, \mathsf{cov}(\mathbf{f}_*)\right). \tag{2.86}$$

For further details see (Rasmussen & Williams, 2006, p. 16).

Typically we consider GPs with a zero prior mean function. But there are many reasons why one may want to explicitly model the mean function, including interpretability of the model and convenience of expressing prior information (Rasmussen & Williams, 2006, §2.7). For example, in (Wahlström et al., 2013) the authors make the assumption that their mean function is constant, and simply place a Gaussian prior on it, i.e. $m(\mathbf{x}) \triangleq \boldsymbol{\beta}$ when $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \sigma_\beta^2 \mathbf{I})$. Indeed, §2.7 of (Rasmussen & Williams, 2006) deal specifically with instances where the mean function is non-deterministic. But it should be noted that a deterministic mean function is *not* the same as a zero mean function (Rasmussen & Williams, 2006, §2.7).

That being said, the mean function conundrum does beg further discussion. The closed form properties of the predictive distribution of the GP, certainly are an attractive prospect when the function is fixed. But that alone should not be enough, and indeed we can still receive closed-form predictions with a non-deterministic mean function see (Rasmussen &

Williams, 2006, §2.7). Indeed, we can consider the analysis that we shall be considering later on in this chapter, for an example where a mean function could benefit the analysis (it was not done this time, as this methodology was primarily intended as a proof of concept, and future work will contain explicit non-deterministic basis function selection). Consider now fig. 2.15. In this figure it can be seen that the difference between the maximum and minimum joint-angle decreases, the slower the speed. The gradient decrease in angle is steeper for a reduction in speed than for an increase in speed, where we see that whilst an increase in speed certainly increases the angular difference, it starts to plateau as we increase the speed. There are a number of reasons for this: first, the ankle plantarflexion angle has a maximum anatomic range, the limits of which we approach as a we increase the speed. Second, when the speed approaches zero, the joint-angle will be zero across the gait cycle. This suggests two different constant behaviours, far away from our observations, in opposite directions (the direction here being referred to is the speed axis in fig. 2.15). As we approach zero speed, the joint-angle will also tend towards zero degrees across the board, hence a zero mean function is appropriate. But as we increase the speed, a zero mean function would betray our analysis, and instead a more suitable choice should be made – such as a constant mean function tailored to maximum range of ankle plantarflexion angle. This being said, though he mean function is important, the kernel function and resulting Gram matrix, arguably play a larger goal for successful regression. Consequently, we return to the covariance function.

Picking a kernel implies specifying a prior distribution over functions and hence an encoding regarding our beliefs about its nature. Just by changing the kernel, the realisations of the Gaussian process change drastically, from the very smooth, infinitely differentiable, functions generated by the squared exponential kernel demonstrated in eq. (2.84). Hence, depending on the kernel and on the number of training points, it can be a very flexible model, able to learn complex patterns – without having to specify a non-deterministic mean function (though the most common choice, as previously noted, is to use a zero mean function). Going back to the above paragraph, Osborne (2015) notes that a GP is effectively a nonparametric model combined with a parametric model, the prior mean. If learning is pursued through maximum likelihood estimation, for this mean, it can lead to overfitting – if overparametrised. But the same is not true for the hyperparameters of the covariance function. Which is typically also why we pick simple prior mean functions.

However, sometimes the mean function does matter. Generally speaking, far away from the observations, the GP modelling formalism will lead to asymptotic uncertainty. Hence, the further we get from the observations, the closer we get to mean, and a zero mean, for many natural phenomena, is not realistic nor accurate methodology for conducting modelling – naturally *some* phenomena may follow this behaviour. Hence, if notions of long-range phenomena behaviour do exist, then these can be incorporated into the mean function for more accurate long-range predictions. Specifying an informative mean-function does also leave one open to problems of overfitting, much like the GP formalism itself. One way of preventing overfitting is to use Bayesian regularisation i.e. place priors on the mean-function parameters.

We shall give the mean function a more incisive treatment in §5.7 with respect to the problem at hand. For now, we turn our attention to the actual tuning of this model class.

2.7.1.2 Model selection

Typically, we do not a priori know the values of the hyperparameters, but might know broadly where they should feature in order to return a model of high accuracy. We receive this model by first considering the foundations of GPs, to arrive at the objective function of our learning (i.e. the optimisation we will undertake to find the values of the hyperparameters best suited to the problem at hand).

Hence, we echo the excellent work by Saatci (2012), by restating that a GP places a prior on functions (Rasmussen & Williams, 2006). Consequently, we can state the problem in terms of Bayes' law

$$\mathbb{P}(\mathbf{f} \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \frac{\mathbb{P}(\mathbf{y} \mid \mathbf{f}, \mathbf{X}, \boldsymbol{\theta})\, \mathbb{P}(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{2.87}$$

where the evidence (normalising constant or more commonly called *marginal likelihood* in the GP literature) is given by

$$Z(\boldsymbol{\theta}) = \mathbb{P}(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = \int \underbrace{\mathbb{P}(\mathbf{y} \mid \mathbf{f}, \mathbf{X}, \boldsymbol{\theta})}_{\text{Gaussian likelihood}} \underbrace{\mathbb{P}(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta})}_{\text{GP prior}} \mathrm{d}\mathbf{f}. \tag{2.88}$$

Because we are marginalising over function values rather than parameter values $\boldsymbol{\theta}$, under a GP model (Rasmussen & Williams, 2006, §2.2), the prior simply reduces to

$$\mathbb{P}(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta})) \tag{2.89}$$

where, as noted before, we have assumed a zero mean function. Taking logarithms, this can be written as

$$\mathbb{P}(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta})| - \frac{1}{2}\mathbf{f}^{\mathsf{T}} K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta})^{-1}\mathbf{f}. \tag{2.90}$$

Much like the prior, the likelihood, as explained by Saatci (2012) and Rasmussen & Williams (2006), is simply just a factorised Gaussian of the form

$$\mathbb{P}(\mathbf{y} \mid \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}, \sigma_n^2\mathbf{I}) \tag{2.91}$$

which, using logarithms, becomes

$$\log \mathbb{P}(\mathbf{y} \mid \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\sigma_n^2\mathbf{I}| - \frac{1}{2}(\mathbf{y}-\mathbf{f})^{\mathsf{T}}(\sigma_n^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{f}) \tag{2.92}$$

where n are the number of data points. Combining eqs. (2.88), (2.90) and (2.92) renders an analytic form of the marginal likelihood, wherein the targets are conditioned only on the observations and the model hyperparameters

$$\begin{aligned}\log \mathbb{P}(\mathbf{y} \mid \boldsymbol{\theta}) = &-\frac{n}{2}\log(2\pi) \\ &-\frac{1}{2}\left[\mathbf{y}^{\mathsf{T}}(K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta}) + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} + \log|K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta}) + \sigma_n^2\mathbf{I}|\right]\end{aligned} \tag{2.93}$$

Rasmussen & Williams (2006) explain that eq. (2.93) separates automatically into calibrated model fit and complexity terms which are optimised to learn the model hyperparameters $\boldsymbol{\theta}$. This fundamental property of GPs, make them particularly powerful since it allows one to balance between the capacity of the model and suitability (fit) to the observations (Duvenaud, 2014, §1.1.1). Equation (2.93) helps us select an appropriate covariance function, based on its fit to the available observations i.e. model selection. There are also other ways, more automatic, of finding a suitable covariance function, such approaches were presented in (Grosse et al., 2012; Duvenaud et al., 2013; Lloyd et al., 2014). In these systems the authors explore an open-ended space of statistical models, to discover a good explanation for a set of observations. One large part of what these papers propose is to automatically search for a kernel combination to take into account things such as smoothness, trends, periodicity and change points, that may be found in the observations. Selecting a kernel is, they concede, more of "black art" rather than a principled science (Duvenaud et al., 2013). Hence, the need for compositional and principled approaches for automatically selecting a *basis* set of kernels, which best fit the regression task under the GP model.

Further, structure can also be learned using a parametric mean function since its parameters would form part of $\boldsymbol{\theta}$ and then optimised under the same guise, i.e. to balance model fit with complexity. Indeed, Rasmussen & Williams (2006) discuss this in §2.7.1 and demonstrate just such a non-zero basis function approach.

The learning process is summarised in algorithm 1 (inspired by (Rasmussen & Williams, 2006, Algorithm 2.1) and (Saatci, 2012, Algorithm 1)) and an illustrative example of GPR is shown in fig. 2.15.

Algorithm 1: Gaussian process regression

Input : $\mathcal{D} \triangleq \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$ :: Training observations
$\mathbf{X}_*$ :: Test inputs
$K(\cdot, \cdot)$ :: Kernel
$\boldsymbol{\theta}$ :: Hyperparameters

$\mathbf{K} \leftarrow K(\mathbf{X}, \mathbf{X} \mid \boldsymbol{\theta})$ ▷ *Calculate base covariance matrix*
$\mathbf{K} \leftarrow \mathbf{K} + \sigma_n^2 \mathbf{I}$ ▷ *Add measurement noise*
$\mathbf{L} \leftarrow \mathbf{K}^{-1}$ ▷ *Compute using Cholesky factorisation*
$\boldsymbol{\alpha} \leftarrow (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$
▷ *Where* $\boldsymbol{\alpha}$ *evaluates to* $\mathbf{L}\,\mathbf{y} - \mathbf{L}\sigma_n^2\,\mathbf{I}(\mathbf{I} + \mathbf{L}\sigma_n^2\,\mathbf{I})^{-1}\mathbf{K}\,\mathbf{y}$ *– a special case of Woodbury's matrix identity (Woodbury, 1950)*
$\mathbf{K}_* \leftarrow K(\mathbf{X}, \mathbf{X}_* \mid \boldsymbol{\theta})$ ▷ *Covariance matrix between train and test inputs*
$\mathbf{K}_{**} \leftarrow K(\mathbf{X}_*, \mathbf{X}_* \mid \boldsymbol{\theta})$ ▷ *Covariance matrix between test inputs*
$\bar{\mathbf{f}}_* \leftarrow \mathbf{K}_*^\mathsf{T} \cdot \boldsymbol{\alpha}$ ▷ *Predictive mean – see (Williams & Rasmussen, 2006, p. 16)*
$\mathsf{cov}(\mathbf{f}_*) \leftarrow \mathbf{K}_{**} - \mathbf{K}_*^\mathsf{T}\,\mathbf{L}\mathbf{K}_*$ ▷ *Predictive covariance – see (Williams & Rasmussen, 2006, p. 16)*
$\log Z(\boldsymbol{\theta}) \leftarrow -\frac{1}{2}\left(\mathbf{y}^\mathsf{T}\boldsymbol{\alpha} + \log \sum_i^N [\mathbf{L}]_{(i,i)} + N \log(2\pi)\right)$
▷ *Log marginal likelihood– see eq. (2.93)*

Output : $\log Z(\boldsymbol{\theta})$ :: Logarithmic evidence (marginal likelihood)
$\bar{\mathbf{f}}_*$ :: Predictive mean
$\mathsf{cov}(\mathbf{f}_*)$ :: Predictive covariance

It is worth noting that model selection is more complex than merely maximising the Bayesian marginal likelihood, i.e. the statement in eq. (2.93). Consider that there can be multiple optima of the marginal likelihood, and that these correspond to different interpretations of the data. *Which one is best?* It depends. First, what questions are we asking? We can estimate the generalisation error of our selected model (e.g. using cross-validation). But we are also at liberty to bounds on how much generalisation we allow at all, this can be done through probably approximately correct (PAC) -Bayes for example (Seeger, 2003). PAC-Bayesian theory is a theory of machine learning that blends both frequentist and Bayesian theory of the world (McAllester, 2013). Typically one

says that with a probability of at least $1 - \delta$ (the *probably*), any classifier (for example) from a hypothesis class which has a low training error, will have a low generalisation error (the *approximately correct* bit). The Bayesian view says that we assume a prior distribution over functions or classifiers (as we are in this chapter, with GPs) and then use Bayes' theorem to update the prior, based on the likelihood over our observations, for each function (MacKay, 2003). PAC-Bayes theory assumes probability distribution on situations with a prior weighting on rules expressing a learners preference for some rules over others. This is manifestly different from the Bayesian standpoint, wherein the starting point a potentially biased joint distribution on rules and situations, which through Bayes' theorem, induces a conditional distribution on rules conditioned on situations (McAllester, 2013).

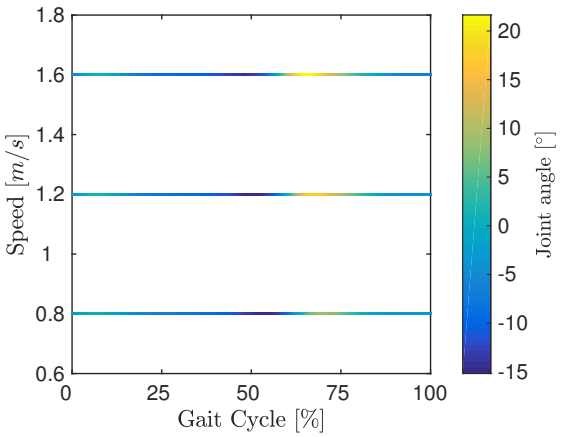
Rasmussen & Williams (2006) state that *learning* in a GP setting concerns the structural form of the covariance function and secondly its parameters – as much as has been explained thus far. But much as we have dealt with learning principally, we have only touched upon the idea of learning parameters, given some observations. Presently, this assumes a chosen covariance function. This will yield a posterior over our parameters certainly. But what of the model? The full Bayesian pipeline requires us to have a posterior over our model choice. These levels of inference are discussed in intimate detail Rasmussen & Williams (2006, p. 109). Really though, it comes down to *Occam's razor* (MacKay, 2003).

The Principle of Parsimony (Vandekerckhove et al., 2015):

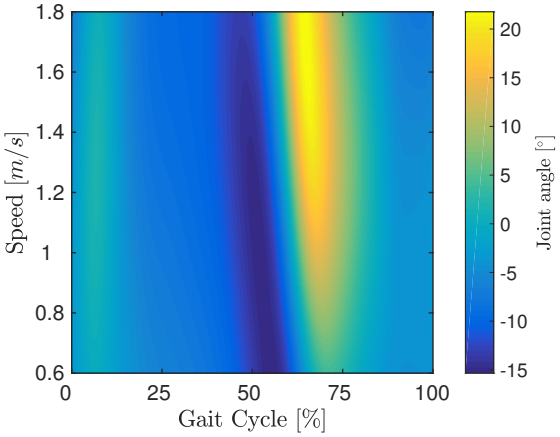
> Throughout history, prominent philosophers and scientists have stressed the importance of parsimony. For instance, in the Almagest – a famous 2nd-century book on astronomy – Ptolemy writes: "We consider it a good principle to explain the phenomena by the simplest hypotheses that can be established, provided this does not contradict the data in an important way". Ptolemy's principle of parsimony is widely known as Occam's razor; the principle is intuitive as it puts a premium on elegance. In addition, most people feel naturally attracted to models and explanations that are easy to understand and communicate. Moreover, the principle also gives ground to reject propositions that are without empirical support, including extrasensory perception, alien abductions, or mysticism. In an apocryphal interaction, Napoleon Bonaparte asked Pierre-Simon Laplace why the latter's book on the universe did not mention its creator, only to receive the curt reply "I had no need of that hypothesis".

For Gaussian processes, this has direct mathematical meaning. A 'simple' model (e.g. one with a structurally simple covariance function such as a linear kernel, with few parameters)

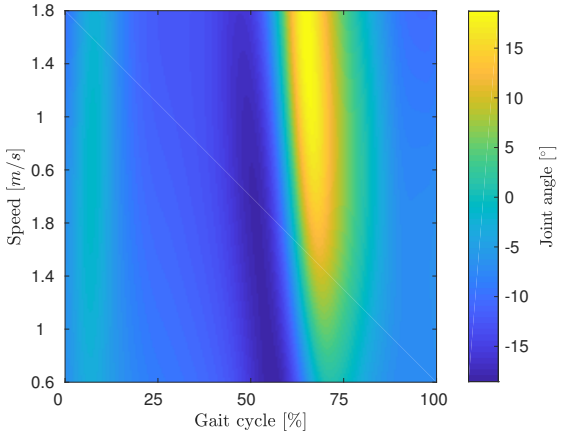
can only countenance a few possible datasets, but these will have high likelihood under this simple model. A complex model on the other has a lot of capacity, rendering it amenable to many datasets, which have likelihood under the model. But since there are now many datasets, and the marginal likelihood is still a probability distribution, probability mass must be shared around, for this to happen the model has to be complex, to take into account the many and varying datasets in its domain. Consequently, whilst Gaussian process learning most certainly can cause overfitting, there are ways (as discussed) to overcome this – but to emphasise: Bayesian inference with Gaussian processes does not automatically avoid overfitting (Rasmussen & Williams, 2006).



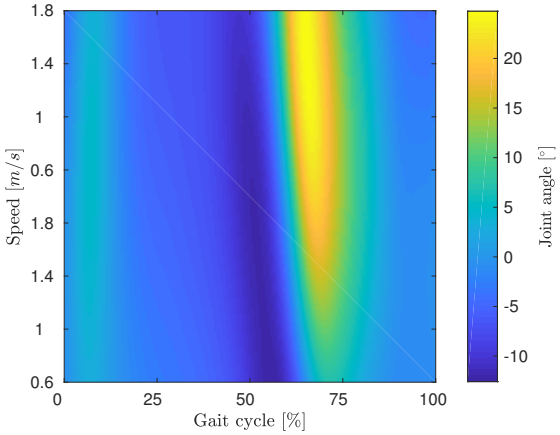
(a) Training inputs $\mathbf{X}$ with targets $\mathbf{y}$.



(b) Posterior predictive mean function as heatmap.



(c) Posterior mean function with variance removed.



(d) Posterior mean function with variance added.

Figure 2.15: Workflow used in obtaining an ankle plantarflexion angle regression manifold, for subject 17, from the Moore dataset (Moore et al., 2015). Figure 2.15a shows the training data used, fig. 2.15c and fig. 2.15d show the posterior predictive uncertainty in the mean function. Refer to fig. 5.7 for the reference frame. Adapted from (Dhir et al., 2018).

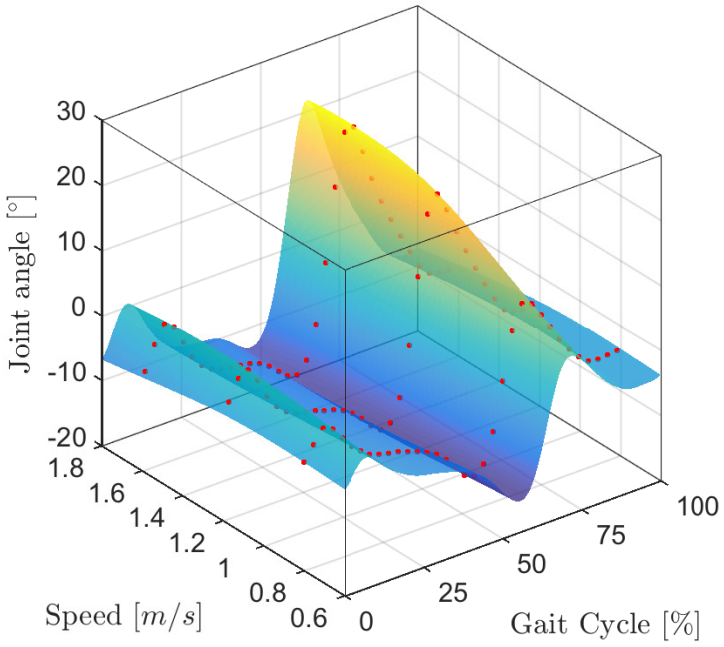


Figure 2.16: The posterior predictive mean function for the tests inputs (shown in red). The result of applying Gaussian process regression. Note that the inputs here are time and velocity (gait cycle and speed on the plot) and we are looking for the response, the joint angle, which we receive through a function, which is estimated through the GP formalism.

2.7.1.3 Other models

It is worth noting that there are other models that could have been used to learn our non-linear multivariate surface functions, present in each locomotion envelope. It is important to note that e.g. splines (a natural contender in our experiments) are just a special case of GP regression, as shown by Kimeldorf & Wahba (1970). Which is to say; if you use a certain type of kernel in Gaussian process regression, you exactly obtain the spline fitting model.

We have chosen to use GPs for our nonlinear multivariate regression because of the substantial domain knowledge that we can incorporate but also because we receive explicit bounds on our posterior uncertainty – something which is useful when dealing with human interaction. That being said, we emphasise that other methods could be of high utility as well. First, we turn to deep learning methods, the advent of which has seen a deluge of state-of-the-art work in virtually all fields of machine learning. Holden et al. (2015) present a method for learning a manifold on the CMU database of human motion (a benchmark motion capture dataset). They note that this manifold can "be treated as a

prior probability distribution over human motion data". The manifold is found through the use of convolutional autoencoders. Incidently the idea of finding manifolds of human motion was touched upon by the GP community as well, notably by GP-based latent variable models, introduced in the preliminary work (see the GPLVM by Lawrence (2003) and the GPDM by Wang (2005)). Given the sheer amount of MOCAP data available, it is likely that this direction of research, will be become more popular. In addition, there are other, more recent approaches from the GP community (Calandra et al., 2016a,b, 2014) for use with locomotion and MOCAP. We employed some of these methods, notably Bayesian optimisation, in the preceding chapter, but otherwise note that their utility for the hybrid controller herein, are outside the scope of this thesis.

Further, there are more commonplace methods (which should naturally not detract from their utility) such as support vector regression. A recent excellent paper by Gu et al. (2015) in which the authors present an regression learning algorithm, which uses a parameter to control the number of support vectors used in the regression task. Naturally we would be amiss if we did not also mention least squares and its many flavours (Bates & Watts, 1988). However, we contend that for our domain, where inputs are highly correlated, and often high-dimensional, GPR is the most natural choice for a regression tool. This is not difficult to see, especially given that GPs scale linearly with dimension (but cubically with number of inputs) (Rasmussen & Williams, 2006). Naturally there are alternatives such as sparse GPs which typically use a low-rank approximation to the covariance matrix, and in so doing reduce the cost from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM + NMD)$ for training (Smola & Bartlett, 2001), where M is the sample size. Naturally there are traditional methods, such as a linear regression, which also work well in high dimensions, provided we account for two of the main shortcomings. First, linear regression has low bias but high variance, this can sometimes be alleviated by removing some bias, and so reduce the variance. Secondly, linear regression, by design, has a lot of parameters, hence we need to encourage it to make use of an eigen-set, if only to be able to interpret the resulting model (MacKay, 2003; Murphy, 2012). This being said, its inherent simplicity poses an interesting venue for further work in this domain.

2.7.2 Dirichlet process

The Dirichlet process is a distribution over distributions. Consider a DP mixture model which enforces that the random parameters governing the observations are drawn from a distribution drawn from a Dirichlet process. Therein we mean to say that the problem is to partition some random variable $\mathbf{X} \triangleq \{X_1, \ldots, X_N\}$ into (meaningful) clusters. This could be e.g. the partitioning of the Iris flower dataset (Fisher, 1936), in which three species of Iris (*Iris setosa, Iris virginica* and *Iris versicolor*) are specified through four features; the length and the width of the sepals and petals. Typically the number of *a priori* flower-class cardinality is specified, but supposing we did not know this in advance, we would be forced to specify an unbounded prior on the size of the flower class. Explicitly the DP ensures that observations (in the flower example, this will be the features) in the mixture model can be factored as

$$\mathbb{P}(X_n \mid \Phi) = \sum_{k=1}^{K} \mathbb{P}(c_n = k)\, \mathbb{P}(X_n \mid \phi_k, c_n = k). \tag{2.94}$$

Alas, what eq. (2.94) demonstrates is that each variable is generated by one of K mixture components, given by parameters $\Phi \triangleq \{\phi_k\}_{k=1}^{K}$. In this exposition c_n is an indicator variable which assumes $c_n = k$ if, and only if, observation X_n was generated by component k by ϕ_k (Wallach et al., 2010). We then take clustering to be the exercise in which we identify which parameters ϕ_k are associated with observations X_n, i.e. those for which $c_n = k$, in other words cluster k. We note, like Wallach et al. (2010), that BNP models then assume that parameters Φ are sampled from some prior distribution $\Phi \sim \mathbb{P}(\Phi)$, whose probabilities that $c_n = k$, are "well defined in the limit" (Wallach et al., 2010), i.e. when the number of clusters $K \to \infty$.

⋆ ⋆ ⋆

Having provided a qualitative notion of the DP, we can now give a more detailed, quantitative, specification of its details.

Definition 2.7.2. (Dirichlet process). The Dirichlet process defines a random measure G in terms of finite dimensional Dirichlet distributions, where $(\Omega, \mathcal{F}, H)$ is a probability space with positive concentration parameter $\alpha > 0$. Then G is distributed according to

a DP, with parameters α and H, written as $G \sim \mathcal{DP}(\alpha, H)$. For every finite partition $\{A_1, \ldots, A_K\}$ of Ω, it follows that

$$\bigcup_{k=1}^{K} A_k = \Omega \quad A_j \cap A_k = \varnothing \text{ when } j \neq k. \tag{2.95}$$

Then, the random probability measure G on Ω, is a draw from a DP if its measure (Fox, 2009) on every finite partition follows a Dirichlet distribution as follows

$$(G(A_1), \ldots, G(A_K)) \sim \text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K)). \tag{2.96}$$

For further details see (Airoldi et al., 2014, §5.3.1) and (Johnson et al., 2014, §2.9.1).

Proof The existence of the Dirichlet process, has been proven many times, in different forms and versions. However, the first, and perhaps most famous was the one by Ferguson (1973), where he invoked Kolmogorov's consistency conditions which confirmed the existence of the DP as a stochastic process which, as noted by Fox (2009), has Dirichlet marginals. The proof in itself is not constructive (nor, it must be emphasised, is definition 2.7.2 – it is but formalism). The importance of constructive objects can be understood by echoing the words of Rocke (1998) where he says that "constructive mathematics is a philosophical doctrine that asserts that objects cannot be shown to exist unless a method is provided for producing them". Such a method was provided by Sethuraman (1994) with his now famous *stick-breaking construction*.

Definition 2.7.3 (Stick-breaking process)**.** The stick-breaking process provides a constructive definition of the Dirichlet process. In it we say that $\beta \triangleq \{\beta_i : i \in \mathbb{N}\}$ is distributed according to the stick-breaking process with parameter γ such that

$$\begin{aligned} v_i &\sim \text{Beta}(1, \gamma) \\ \beta_i &= v_i \prod_{j=1}^{i-1}(1 - \beta_j) \qquad\qquad i = 1, 2, \ldots \end{aligned} \tag{2.97}$$

we typically denote this process $\beta \sim \text{GEM}(\gamma)$.

We demonstrate the process as per definition 2.7.3 described above, in fig. 2.17. Looking at fig. 2.17 notice that for low values of γ, the stick weights are concentrated on the first few weights (meaning that the observation points are concentrated on a few clusters), while

the weights become more evenly dispersed as we increase γ (meaning that we posit more clusters in our observation points). From this intuition we can form the stick-breaking construction of the Dirichlet process; theorem 1.

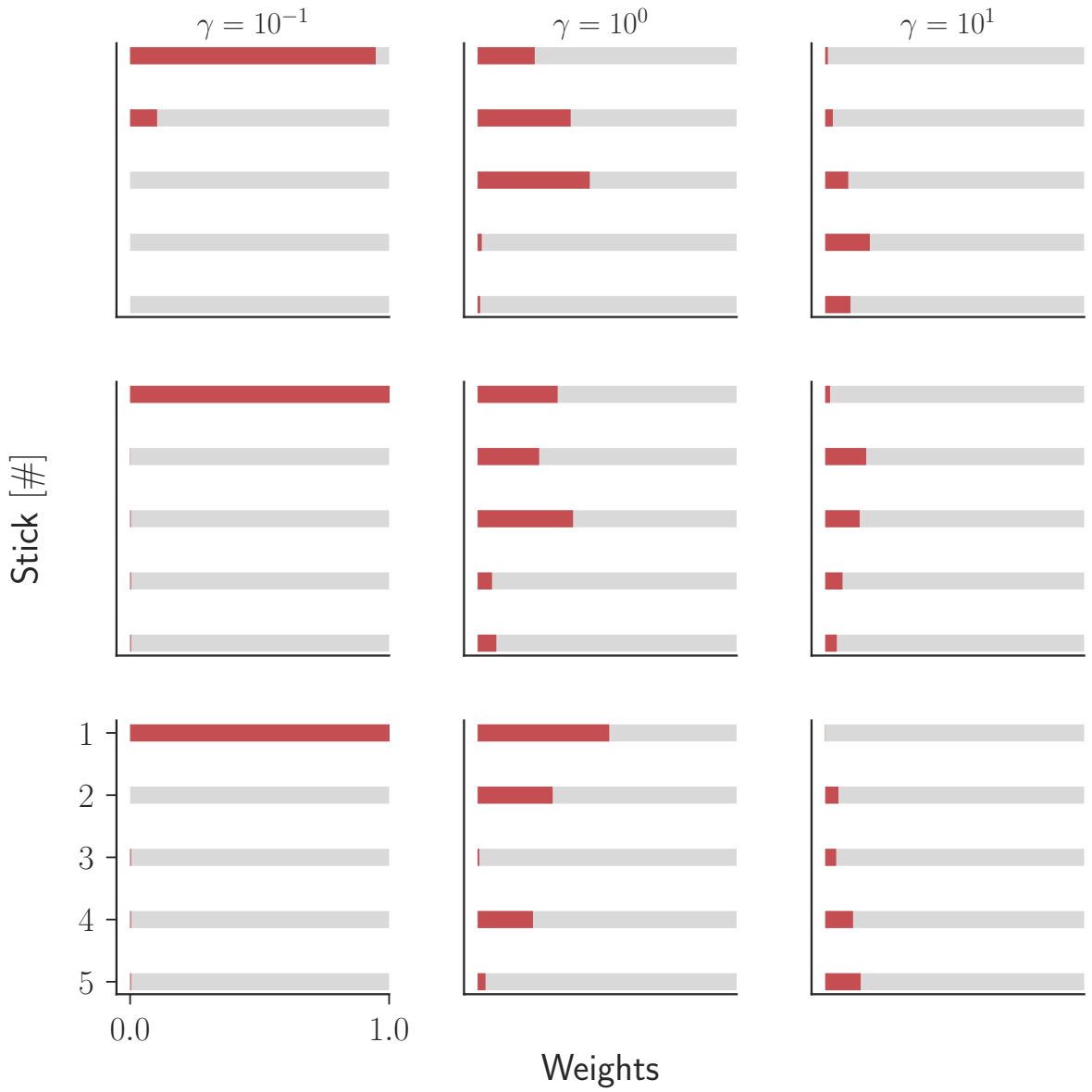


Figure 2.17: Illustration of the stick-breaking construction, where the bottom left panel's axes play the role of legends for the rest of the panels. As demonstrated in definition 2.7.3 we have a unit length stick, which we break at a random point v_1; the length of the remaining piece which we keep is called β_1. Recursively we then break off pieces of the remaining stick, to generate $\beta_2, \beta_3, \ldots$ and so on. We demonstrate the process for three parameters values (on the columns), and demonstrate three samples of each (on the rows) for the first five stick lengths (weights).

Theorem 1 (Stick-breaking construction)**.** *Let* $\beta \sim \text{GEM}(\gamma)$ *and* $\theta \overset{i.i.d.}{\sim} H, \forall i \in \mathbb{N}$*, if*

$$G = \sum_{i=1}^{\infty} \beta_i \delta_{\theta_i} \Rightarrow G \sim \text{DP}(\alpha, H). \tag{2.98}$$

For further details see (Sethuraman, 1994, p. 8)

As with fig. 2.17, we are now in a position to visualise draws from a DP, and better understand their quantitative behaviour. See fig. 2.18. The stick-breaking process is but one of many representations of the Dirichlet process, as we have already noted, Ferguson (1973) was the first to prove the existence of the DP. But since then, as noted, more representations have followed, and we have already touched upon the random-measure view, in terms of the stick-breaking process. However, for completeness the two other, most famous, representations must also be included, both of which take the random-partition view of the problem.

Blackwell & MacQueen (1973) followed the work by Ferguson (1973) and introduced an extension of the Pólya urn scheme, to allow for a continuum of colours. Therein they used de Finetti's theorem (Diaconis, 1977) to prove the existence of the random probability measure, which has since confusingly become the Blackwell-MacQueen urn scheme (but is still also goes by the name of Pólya's or Hoppe's urn scheme (Teh, 2011)). It satisfies the properties of the Dirichlet process and is simple in its construction. In the same vein Aldous (1985) gave us the Chinese restaurant process (CRP), which is another effective way of constructing a Dirichlet process. The CRP goes by the following analogy: assume that there is a Chinese restaurant with infinitely many tables. As customers enter the restaurant they sit randomly at any of the occupied tables or they choose to sit at the first available empty table, with some probability. The CRP defines a distribution on the space of partitions of the positive integers, where tables are clusters and customers are observations, and the final seating arrangement is the found assignment with some inference scheme.

It should be noted that although there are many representations of the DP, they are all equivalent in the sense that they satisfy the properties of the DP as laid out by Ferguson (1973), but their formulation differs in their examination of the problem from alternate representation views. For a detailed discussion on all of the representations see (Fox, 2009, §2.9).

Having established that the DP does exist, a number of important properties flow from definition 2.7.2.

➢ For any measurable set $A \subset \Theta$, the base distribution H specifies the mean of the DP

$$\mathbb{E}[G(A)] = H(A). \tag{2.99}$$

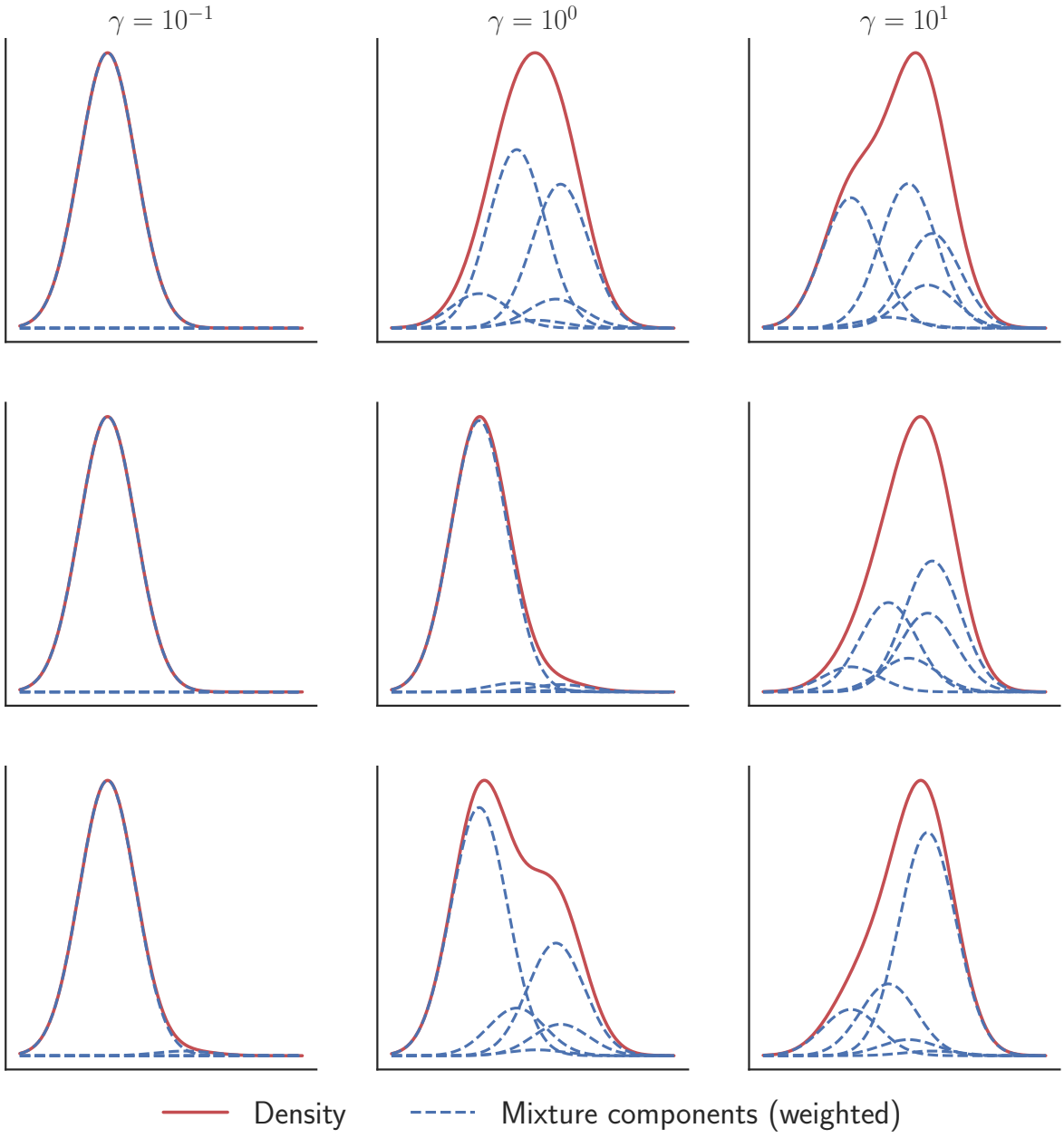


Figure 2.18: We depict density estimation examples using the Dirichlet process mixture model (see fig. 2.21a), for different concentration parameter (γ) values. The DPMM uses component densities from a parametric family and represents the mixture weights as a DP. Note that we have induced a partition on *objects*, as per the CRP analogy, where the objects are our mixture components ϕ_k as per eq. (2.94). The difference being that under this modelling paradigm we have an infinite number of components, but since modelling in the infinite mixture domain, yields a 'rich gets richer' -type behaviour, we only ever use a finite subset of components (this is of course a simplification of eq. (2.102)). Importantly, component count will always be finite and discrete, whereas the content of those components can take on any real of discrete value(s) we so wish under the likelihood model.

➢ Teh (2011) notes that the concentration parameter α can be understood as the

inverse variance, which is to say

$$\mathbb{V}[G(A)] = \frac{H(A)(1 - H(A))}{\alpha + 1} \tag{2.100}$$

Consequently the larger α, the smaller the variance $\mathbb{V}[\cdot]$ and hence the DP will concentrate more mass around the mean $\mathbb{E}[\cdot]$.

➢ Sethuraman (1994) showed that if $G \sim \text{DP}(\alpha, H)$ then, with probability one, that random draw can be written as

$$G = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \quad \text{for} \quad \theta_k \mid H \sim H \quad \text{when} \quad k = 1, 2, \ldots \tag{2.101}$$

Johnson et al. (2014) refer to this property by its measure theoretic name, i.e. that G is atomic.

➢ Ferguson (1973) showed that the posterior distribution of G is found by first assuming a sequence of independent draws (Teh, 2011); $\theta_1, \ldots, \theta_n$ from G (since G is a distribution over Θ, samples $\theta_{1:n}$ take values in Θ). Using the partition given in definition 2.7.2 and counter $n_j \triangleq \#\{j : \theta_j \in A_k\}$ – the number of observed values in A_k, the posterior is given by

$$(G(A_1), \ldots, G(A_K)) \mid \theta_1, \ldots, \theta_n \sim \text{Dirichlet}(\alpha H(A_1) + n_1, \ldots, \alpha H(A_K) + n_K) \tag{2.102}$$

which follows from definition 2.7.2 and the conjugacy between the Dirichlet and the multinomial distributions (Teh, 2011). Then, as explained by Ferguson (1973), the posterior distribution of the random measure G, and G alone, is given by

$$G \mid \theta_1, \ldots, \theta_n \sim \text{DP}\left(\alpha + n, \frac{\alpha}{\alpha + n} H + \frac{1}{1 + \alpha} \sum_{i=1}^{n} \delta_{\theta_i}\right) \tag{2.103}$$

which is a weighted average between the prior base distribution H and the empirical distribution $\sum_{i=1}^{n} \delta_{\theta_i}$ where δ_θ is a point-mass located at θ (Teh, 2011).

All of these properties will become important when we extend the hidden Markov model to the Bayesian nonparametric domain. But, in order to frame that work, we need to consider the models without the temporal component, and thus return to the antecedent model which introduced this section, and which we are now in a position to consider nonparametrically.

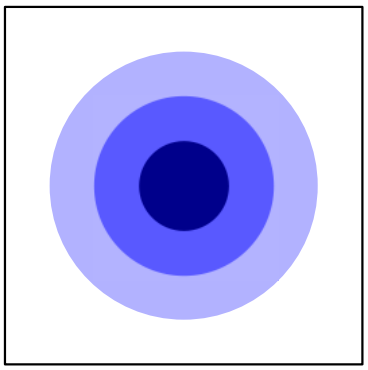
Finally, whilst we have demonstrated the technical proclivities of the DP, we have not yet explicitly spelled out answers to more simple questions such as *how do we receive a discrete distribution from a continuous base measure?*. The base measure H is a measure on a typically a continuous space Θ (see next section for more details) as shown in fig. 2.19a. What the DP gives us then is a distribution over partitions of Θ where α is really tells us *how* many regions our partition will have; a smaller alpha, will give us fewer regions, since it works like an inverse variance, and will consequently have a higher propensity to repeat draws from partitions in the output distribution as shown in fig. 2.19b and fig. 2.19c. In the stick-breaking analogy and construction, a small alpha results result in fewer sticks, but they each individually have higher weight. Large values of alpha of α on the other hand, means that the we spread the probability mass more widely, yielding more sticks, each individually with smaller weights. Sticks, being finite and discrete, by analogy, thus explain why we receive a discrete distribution from a continuous base measure as shown in fig. 2.20. Herein in we see the process in action; we start with a unit-length stick, which is broken at a random point β_1 and the length of the piece that we keep we call θ_1 – this is our first weight (Murphy, 2012). The remaining stick length $(1 - \theta_1)$ is then recursively broken, with stick lengths recorded concurrently with corresponding stick weights $\theta_1, \theta_2, \theta_3, \ldots$

This construction means that samples from a Dirichlet process are discrete with probability one – hence its utility in providing a distribution over probability measures.

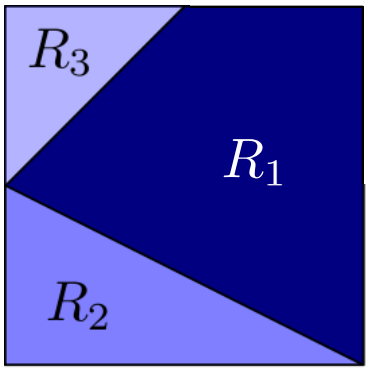
2.7.2.1 Dirichlet process mixture model

The most common application of DPs is not how they have been utilised in this thesis, where we are interested in temporal cluster assignment. Rather, they are typically used to extend mixture models to the infinite domain by making them nonparametric (Teh, 2011), yielding the Dirichlet process mixture model (DPMM).

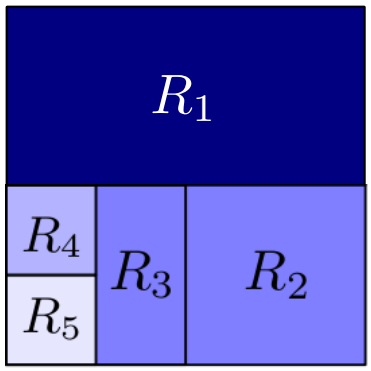
Definition 2.7.4. (Dirichlet process mixture model). The Dirichlet process translates to mixture models with a countably infinite number of components (Teh, 2011) – see fig. 2.21a. Observations $\{y_1, \ldots, y_N\}$ are assigned using latent parameters $\{\theta_1, \ldots, \theta_N\}$. Each latent variable is drawn I.I.D. from G, where each observation is distributed according to $F(\theta_i)$



(a) The base measure H on a two-dimensional space.



(b) One possible partitions with three regions noted by $\{R_i \mid i = 1, 2, 3\}$.



(c) One possible partition with seven regions noted by $\{R_i \mid i = 1, \ldots, 5\}$.

Figure 2.19: Completely random partitions of a two-dimensional space (e.g. Θ) shown in fig. 2.19a. The partition regions R_i are proportional in size and colour to the density in fig. 2.19a. The nature of the Dirichlet process is that in expectation $\mathbb{E}[G(R_i)] = H(R_i)\ \forall i \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ are the natural numbers. The continuous measure in fig. 2.19a has been depicted with discrete stratifications for clarity of exposition, but remains a continuous measure. Figure was inspired by Murphy (2012) and Sudderth (2006).

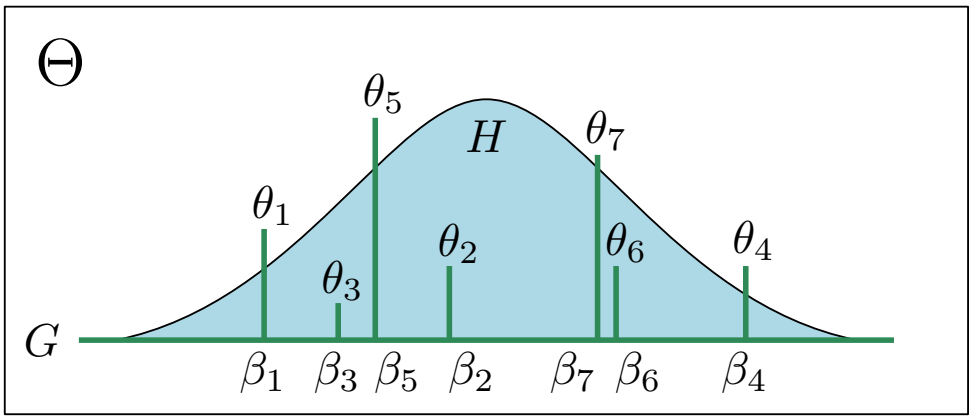


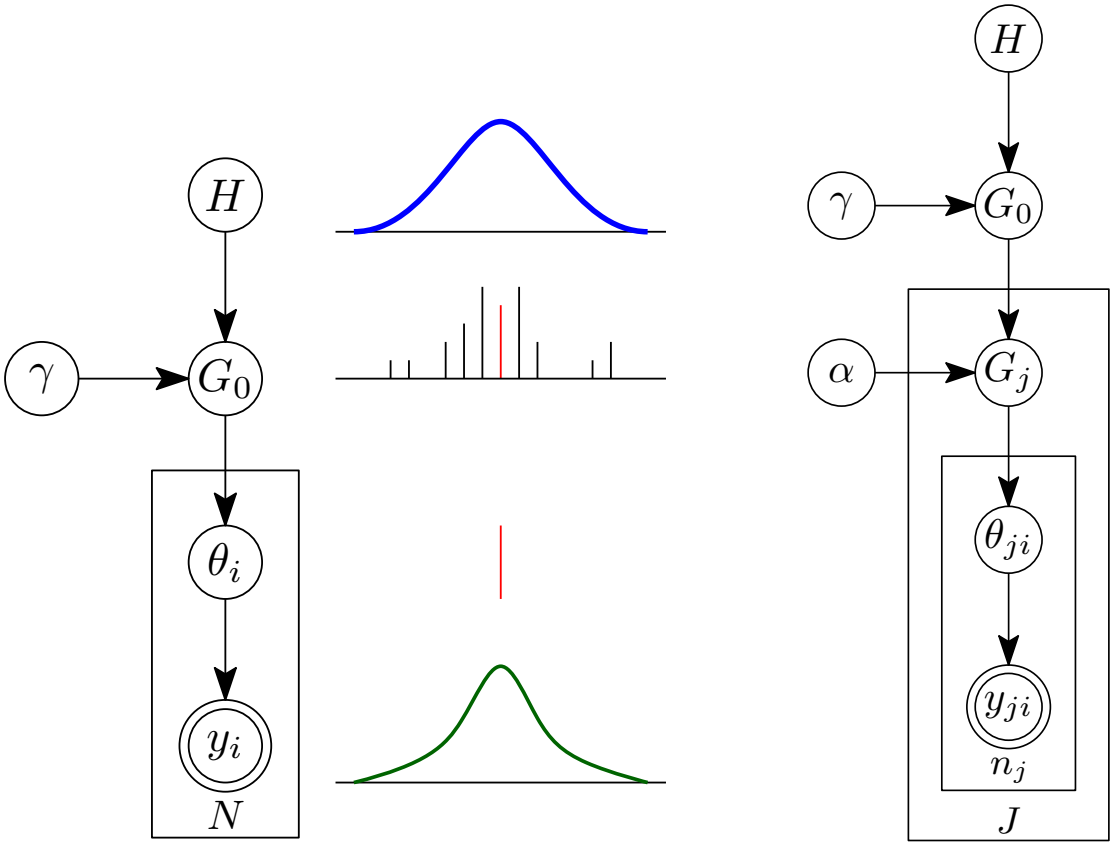
Figure 2.20: A sample draw G from the Dirichlet process, constructed using the stick-breaking process. Where H is the base measure, and the random measure G is a realisation of the DP – the infinite sum of stick or atoms with weights θ_k taken at locations β_k. Through this construction $\sum_{k=1}^{\infty} \beta_k = 1$ and consequently the Dirichlet process specifies a partition over parameter space Θ.

with parameters θ_i where $F(\cdot)$ comes from a parametric family $\mathcal{F} \triangleq \{F(\theta) \mid \theta \in \Theta\}$. The model evolves as:

$$\begin{aligned} G_0 \mid \gamma, H &\sim \mathcal{DP}(\gamma, H) \\ \theta_i \mid G_0 &\sim G_0 \qquad\qquad i = 1, 2, \ldots, N \\ y_i \mid \theta_i &\sim F(\theta_i). \end{aligned} \tag{2.104}$$

For further details see (Teh, 2011).

Consider fig. 2.20, herein when we set the base measure to be the defined on the space of the univariate Gaussian distribution i.e. $H \triangleq \mathcal{N}(\mu, \Sigma)$, a so receive the infinite Gaussian mixture model in the taxonomy of DP mixture models.



(a) Dirichlet process mixture model with graphical illustrations of model components shown on the right.

(b) Hierarchical Dirichlet process mixture model.

Figure 2.21: Dirichlet process and hierarchical Dirichlet process mixture models shown as graphical models. Refer to the definitions for the generative models. Note in fig. 2.21a the inclusion of graphical illustrations showing the hierarchical nature and evolution of model components. The analog is true for fig. 2.21b and is hence not included for brevity.

Further, fig. 2.21 gives a good representation of the model. Because draw G from the DP is discrete, as shown by the second level from the top in fig. 2.21a, the samples $\theta_i \ \forall i \in \mathcal{I}$ can assume the same value with a strict probability of one. Whereupon observation y_i is simulated from cluster i in index set $\mathcal{I}$, under distribution $F(\cdot)$ with parameter θ_i.

DPMMs have been used in a number of clustering applications, where the number of clusters is not known a priori. They are also used in applications in which we believe the number of clusters grows without bound as the amount of observations grows (Teh et al., 2006). It is not just in clustering where the unknown state cardinality problem arises, DPs have also found uses in applications beyond clustering, where the number of latent objects

is not known or unbounded. A number of examples (Teh et al., 2006; Teh, 2007; Murphy, 2012) include:

- Nonparametric probabilistic context free grammars
- Visual scene analysis
- Document clustering
- Haplotype inference
- Phenomena detection

as well as many others.

In many of our listed applications it is important to be able to model the same set of objects in different contexts (Teh, 2007), allowing for the possibility that a higher order process may be responsible for lower-level instantiations of phenomena. As Teh (2007) tells us: this corresponds to the problem of grouped clustering and can be dealt with using hierarchical Dirichlet processes (HDP).

2.7.2.2 Hierarchical Dirichlet process mixture model

Perhaps one of the most useful properties of the HDP, is that it allows us to share clusters among multiple clustering problems (Teh & Jordan, 2010). A simple example to illustrate its utility; posit that one would ask groups of people for their favourite colour. This would result in high counts for popular choices such as red, blue and green, almost certainly across all groups asked. Because we know this to be almost certainly true, we can model colour preference with a HDP, since we want to induce sharing of colours between groups, or in other words to "share statistical strength" (Teh & Jordan, 2010, §6.1).

Definition 2.7.5. (Hierarchical Dirichlet process). A collection of random measures $\{G_j \mid j \in \mathbb{N}\}$ are distributed according to the hierarchical Dirichlet process with base-strength and group-strength parameters γ and α respectively, with a given base distribution H, if

$$\begin{aligned} G_0 \mid \gamma, H &\sim \mathcal{DP}(\gamma, H) \\ G \mid \alpha, G_0 &\sim \mathcal{DP}(\alpha, G_0). \end{aligned} \tag{2.105}$$

See the graphical model in fig. 2.21b. For further details see (Teh et al., 2006, §4).

It is now trivial to form the hierarchical Dirichlet process mixture model (HDP-MM), by combining definition 2.7.4 and definition 2.7.5.

Definition 2.7.6. (Hierarchical Dirichlet process mixture model). By combining our understanding of DPMMs from definition 2.7.4 and definition 2.7.5 it can be seen that the HDP may be used as a prior distribution over the factors in grouped observations. Furthermore, we assume that observations are exchangeable within a group. Let $j = 1, \ldots, J$ index the groups, let $i = 1, \ldots, n_j$ index the number of observations in group j. Teh et al. (2006, §4) explain that θ_{ij} is a factor explaining observation y_{ij}, and so the full HDP-MM evolves as

$$\begin{aligned} G_0 \mid \gamma, H &\sim \mathcal{DP}(\gamma, H) \\ G_j \mid \alpha, G_0 &\sim \mathcal{DP}(\alpha, G_0) \\ \theta_{ij} \mid G_j &\sim G_j \\ y_{ij} \mid \theta_{ij} &\sim F(\theta_{ij}). \end{aligned} \tag{2.106}$$

The corresponding graphical model is shown in fig. 2.21b. For further details see (Teh et al., 2006, §4).

⋆ ⋆ ⋆

Finally, to conclude this section. The reader will be very aware that we have specifically disengaged from the topic of posterior inference. Part of our contribution is specifically to deal with this issue in an *automatic* fashion, which is to say that the observation-fitting task is done via probabilistic programming, which we deal with in intricate detail in §2.6.

CHAPTER 3

Incidence detection

Contents

In this chapter, we will explore what some authors term "good old fashioned AI" (GOFAI) (Bostrom, 2014). The GOFAI taxonomy encompasses a lot of domains, thus for emphasis, completeness (and as a simple reminder) it is worth noting the nature of the systems *we* are considering: *dynamic* ones. The analysis that we perform is comparatively simple, but as we shall see, has high utility in our application domain of interest, namely human incidence detection. This is to mean: we are interested in contiguous segments of activity, *incidences*, for humans. This sort of human activity recognition is required if we are to compose intelligent prostheses that can learn from their surroundings.

Human activity recognition (HAR) is inherently translation invariant and hierarchical (Ronao & Cho, 2016) and thus lends itself well to being formulated as a classification problem, where the classification is over a space of *incidents*. It is a field, which has

received considerable attention in the last few years, in part due to its high demand in various application domains, which make use of time-series sensor data to infer activities (Ronao & Cho, 2016). The interest for us then is natural, and marks the start of our venture into complex time-series analysis with the goal of using it for control. Human activity recognition is a problem, which we believe can be addressed by turning to the humble mobile phone. Modern 'smartphones' incorporate numerous, sensors such as microelectromechanical devices including:

- accelerometers;
- gyroscopes;
- inertial modules;
- microphones;
- magnetometers and many others.

The substantial computing power of these units, coupled with their impressive array of sensors, means that harnessing the power of them could provide an alley for dealing with robust activity recognition. In our case by focusing on *fall detection* – a highly dynamical incident. Rather than dealing with *all* sorts of human activity recognition, for our purposes we focus on the harder task of dealing with a subset class of recognition. Naturally though, fall detection is but a proxy for incidents of interest. Ultimately this is because the prosthesis is required to interface with the user's sensory-motor control system (Tucker et al., 2015; Wang et al., 2012). Consequently an active prosthesis needs a form of perception layer or intention inference, irrespective of how trivial it might seem. Thus the reason why we focus on fall activities, is because this is an action which demands a forceful and immediate response from the control system. Thus, first and foremost, the controller needs to be able to identify a highly dynamic incident such as this.

As we shall see in later chapters, going for a wholesale classification of *everything*, over all length scales, is scarcely possible nor, perhaps, necessary. Choosing what granularity (in essence, what part of the activity hierarchy we consider) of information we want is as much part of the modelling task, as the modelling itself. Hence, our interest in this domain is formulated through the following problem statement, which we qualify with a method motivation followed by our proposed solutions drawing on material from Dhir & Wood (2014).

Problem statement People fall. As they get older they fall more often as the incidence of balance problems and muscle weakness increases. To develop tools and procedures for dealing with falls, certain problems need to be dealt with:

1. Human movement measurements are high-dimensional, highly nonlinear and temporal. To make incidence classification easier, is it possible to reduce the dimensionality to a more manageable space which improves classification accuracy?
2. What effect does missing observations have on the classification performance?
3. Though human movement is nonlinear, can we, with high accuracy, use linear classifiers to robustly identify fall incidents?
4. The study by Luštrek & Kaluža (2009) provide a good baseline for answering the above questions. Can we improve upon their results using elementary machine learning and statistical tools?

Motivation This chapter presents a study of standard machine learning methods to achieve effective fall detection, which can be employed in daily geriatric practise. There is a wealth of available data on essentially people falling. This makes it a good fit for machine learning methods. As far as machine learning goes, the temporal features needed to detect one are not particularly complex, and a number of systems are available (Wang et al., 2017b). But there is a dearth in robust application of machine learning methods. We rectify this by employing Kalman smoothing for in-painting of missing observations and demonstrate that it is enough to use simple multivariate linear discriminant analysis to separate falling incidences from other events.

Contributions Correspondent to the list identified above in the problem statement, we propose the following solutions as our primary contributions:

1. We show that using basic dimensionality reduction methods improves classification performance. This is hardly surprising, but goes to show that much of our observation space has no utility for classification performance.
2. Completing the observation space with a simple generative model, improves classification performance. Again, not surprising, but demonstrates the ease with which improvements can be found.

3. We also demonstrate that reasonable accuracy can be achieved, by considering merely individual motion capture tags, attached to the subject. This further demonstrates the redundancy in the observation space.

⋆⋆⋆

The material presented within is conditional on prior work which can be found in the preliminary material. The following subsections have relevant dependencies:

➢ State-space modelling is employed in §3.3.3 and should be set against §2.4

➢ Section §3.4 on dimensionality reduction, is informed by §2.1

➢ Various classification schemes are applied in §3.5, they are covered in greater detail in §2.2

3.1 Fall detection

Falling represents a major source of anxiety for many groups of people; healthy and those suffering from a disease which affects their balance and gait. Especially for the elderly population, falling presents a particular hazard which can often lead to morbidity or mortality. Tromp et al. (2001) note that each year, one in every three adults aged 65 and older falls. Falls can cause moderate to severe injuries, such as hip fractures and head trauma and can increase the risk of early death (Tromp et al., 2001). Falls are the leading cause of injury deaths and accounted for 83% of all fatal falls in Ireland, in 2005. They are the leading cause of injury-related hospitalisation among people 65 years and older in society (Bourke & Lyons, 2008). Furthermore, Stevens et al. (2012) highlight the fact that among older adults, falls are the leading cause of both fatal and nonfatal injuries. In the United States (US), in the 2006 alone; 2.3 million nonfatal fall injuries among older adults were treated in emergency departments and more than 662,000 of these patients were hospitalised (Stevens et al., 2006). As a consequence of this high incidence rate, the direct medical costs of falls, adjusted for inflation, was $30 billion in 2006 (Stevens et al., 2006). It has been shown that the medical consequences of a fall are highly contingent upon the response and rescue time. Thus, a highly-accurate automatic fall detection system is likely to be a significant part of the living environment for the elderly to expedite and improve

the medical care provided whilst allowing them to retain autonomy for longer (Mubashir et al., 2013).

As the above paragraph shows, falling represents a significant drain on medical resources, as a result of its prevalence in society at large. But also in industrial settings, falling presents a major cause of concern too. The Occupational Safety & Health Administration (OSHA - an agency of the US Department of Labour) states that out of 4,188 worker fatalities in private industry in calendar year 2011, 738 or 17.6% were in construction. In construction, the leading cause of worker deaths on construction sites was falls (259 out of 738 total) (OSHA, 2012). According to the 2008 Liberty Mutual Workplace Safety Index (Liberty Mutual Research Institute, 2008), the annual direct cost of disabling occupational injuries due to slips, trips and falls is estimated to exceed $11 billion. The Index reports that falls on the same level are the second most costly occupational injury (estimated annual cost of $6.7 billion) (Liberty Mutual Research Institute, 2008).

Whilst the relevant authorities have steadily decreased the attrition rate by various means, primarily through changes to the health and safety legislation, rules and regulations can only be passively preventive. This means that while they might have the effect of making the person in question more careful, accidents can never be completely overcome, because of the randomness of their nature. However, using machine learning and signal processing methods, we can seek to actively monitor the equilibrial state of the subject and intervene with appropriate action, when it has determined whether or not the subject has passed from a stable to an unstable (falling) state.

3.1.1 A statistical approach

Noury et al. (2007) note that most state-of-the-art fall detection systems rely on accelerometer instrumentation, like the one presented by Zhang et al. (2006). Although Bourke & Lyons (2008) proposed fall algorithms separately based on thresholds of both signals from a tri-axial accelerometer and a bi-axial gyroscope and reached a performance of 100%. Despite these encouraging figures, fall-detection systems have not been commercialised on a large scale, and there is little or no use of these devices in daily geriatric practice (Noury et al., 2007). There are multiple reasons for this; some devices demonstrated inadequate operation, some had inadequate ergonomics or were not

accepted by the users due to the stigmatisation of the fragility of an old person. The *most* potent reason is rejection of the equipment by both the wearer and the remote monitoring systems was, however, due to the rate of false alarms (Noury et al., 2007). We posit that such false alarms can be reduced using completed training sets for the classifier, by way of smoothing.

We investigate task-specific activity recognition, with the goal of improving classification performance. We identify two areas where improvements can be exploited to improve activity classification accuracy:

1. We propose a smoothing model which can accurately in-paint missing pose data.
2. We suggest transforming high-dimensional activity feature vectors into a low-dimensional eigenspace using supervised dimensionality reduction.

While our ultimate aim is a real-time system that infers pose continuously from low-rank observations; there is immediate practical utility associated with the smoothing methods developed in this chapter, particularly when used in conjunction with the technique of Zhang et al. (2006), for detecting when a serious fall may have occurred. In this system, high confidence fall activity recognition, preceding a period of constant negative gravitational acceleration is necessary.

The literature provides many techniques for activity detection see e.g. (Bourke et al., 2007; Hwang et al., 2004; Noury et al., 2003), but few have focused on smoothing and dimensionality reduction for pose models as we do here. In the literature, feature vectors used for classification are typically very large; Luštrek & Kaluža (2009), use feature vectors ranging in dimensionality from $\mathbf{x} \in \mathbb{R}^{240}$ to $\mathbf{x} \in \mathbb{R}^{2,700}$. With careful use of cross-validation and regularisation, high-dimensional feature vectors can be used for activity recognition with high accuracy. Higher accuracy still can be achieved by using task-specific regularisation. That being said, feature vectors of these sizes are impractical and will lead to problems with overfitting, despite measures, such as pruning and cross-validation. We investigate computationally efficient models for dimensionality reduction for improved classification performance, in conjunction with improved models and methods for in-painting missing pose information for the purpose of decreasing false positives in fall detection and reporting. While the machine learning literature has focused on the development of sophisticated articulated-pose models (Khandoker et al., 2007) we

show that using even the simplest pose models to in-paint missing pose data can have a significant impact on activity classification performance.

3.2 Related work

Activity recognition, with a focus on fall-detection, can be broadly separated into two categories; domain knowledge (DK) and machine learning (ML) based.

Mirchevska et al. (2013) explain that DK-based methods rely on threshold heuristics, whereas ML-based methods rely on recorded fall and non-fall data. Both seek to make the detection automatic. To this end a number of different approaches for the automatic detection of falls, using various sensors, have appeared in recent years – see (Hwang et al., 2004; Diaz et al., 2004; Noury et al., 2003; Doughty et al., 2000). These fall-detection devices use either the near horizontal orientation of the falling person, following the fall, and or the impact of the body with the ground to identify a fall, the typical sensors used for this are accelerometers (Bourke & Lyons, 2008). Current fall detection systems can be classified into three types, based on the type of sensors used:

- vision,
- audio and
- wearable-sensor-based.

An overview of fall-detection taxonomy is shown in fig. 3.1.

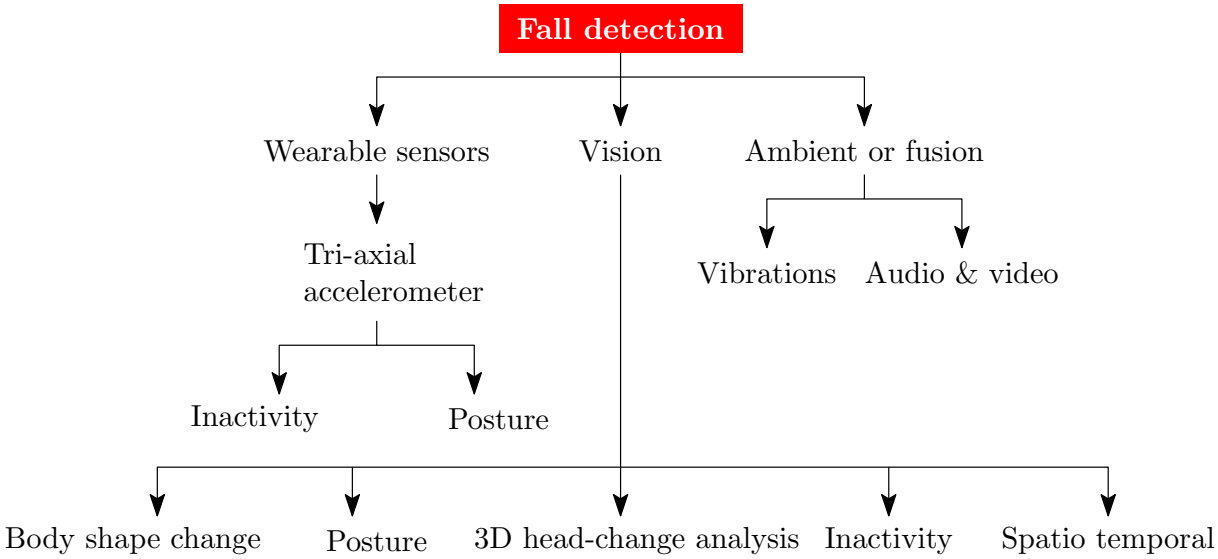


Figure 3.1: Classification of fall-detection methods. Adapted from (Mubashir et al., 2013).

Within the purview of fig. 3.1 it is worth considering a problem that rarely is paid a lot of attention in the HAR literature, namely imbalanced data. The situation with imbalanced data typically presents itself when our dataset has a non-uniform distribution of classes. Ideally we want the classes to be distributed equally. To combat this issue there are some strategies that we can adopt, the simplest of which is to collect more data, though the are any number of reasons why this may be prohibitive. Another, strategy concerns resampling of the dataset where we deliberately under-sample the over-represented classes. It is also possible, though perhaps not best advised, to add synthetic examples of the under-represented class, to balance out the dataset more in its favour. A finally, it is possible to employ classification algorithms that penalise the model for making mistakes on minority classes, and thereby induce it to 'take more care' when dealing with the minority classes during online classification. With this in mind we now consider some related pieces of work.

Zhang et al. (2006) present a system with real-time classification of human movements based on data collected from a smartphone mounted on the subject's waist. Using their algorithm, body motion is labelled with five categories. This classification is achieved by using a combination of the kernel Fisher discriminant (KFD) and the k nearest neighbour (k-NN) algorithm. The k-NN algorithm is explained in §2.2, so we focus instead on the KFD. When linear dicriminant analysis is used with a kernel, the results is the KFD. Consequently, to understand KFD, we need to consider LDA, which is function which seeks to find a projection of the observation, in which space the class separation is maximised. Consequently, in this space, the class separation is maximised and so too the ability to accurately make classifications. When we are dealing with a non-linear observation space, these are mapped under KFD, via a kernel, into a kernel feature space. Then in kernel feature space, a sequence of projection operations again seek to maximise the separation between classes. Hence, in sum, Zhang et al. (2006) use KFD for observations projections and consequently class separation, and then employ k-NN to do the actual classification in this reduced space.

For observation Zhang et al. (2006) measure acceleration, based on a tri-axial accelerometer, where the system is continuously monitoring changes to the gravitational acceleration parameter. If the smartphone acceleration is near absolute gravitational acceleration, for the duration of a second or more, the subject is considered motionless. Once this

is detected the data is backdated for 1.5s (Zhang et al., 2006), and that section of the data is used as the sequence input for two classification algorithms which are employed to determine if there actually was a fall event (one of the five activity labels). But the feature vector used in the backdated period has a cardinality of $\boldsymbol{\phi} \in \mathbb{R}^{192}$, which is large (for GOFAI methods note) and could lead to problems with overfitting, especially as the number of training samples used was low at $N \approx 730$. Contrast this approach against a recent piece of work from the deep learning community. Ronao & Cho (2016) use a deep convolutional neural network to perform activity recognition. They used smartphone accelerometer and gyroscope tri-axial sensor data, from 30 volunteers who performed six different activities while the smartphone was in their pockets. In their study $\boldsymbol{\phi} \in \mathbb{R}^{128}$ represents one activity. They used $N = 7352$ for training and $N = 2947$ features for testing. They recorded an overall performance of 94.79% on the test set with raw sensor data, and 95.75% with additional information of the temporal fast Fourier transform applied to their data.

Consider now why the CNN is in fact a good model for this sort of observation set. Convolutional Neural Networks are similar to normal neural networks, they are made up of neurons which have tunable parameters and biases. Each neuron in the network takes an input, performs the usual dot-product operations and then finished with non-linear function, the CNN however provides a slightly different model flavour. They use a CNN for HAR to learn complex features automatically from the raw accelerometer signal to differentiate between different activities of daily life. This is done by using a 1D CNN which means that observation segments have a height of 1 as one-dimensional convolution (depth wise) is performed over the acceleration signal – this sort of input alteration is what Wang et al. (2017a) call "input adaptation", which they explain is a way of forming a virtual image of acceleration signals, to fit with the more typical CNN input type: images. After convolution, pooling and fully-connected layers follow, and these layers actually perform the classification task (Ronao & Cho, 2016). Wang et al. (2017a) explain that there are two reasons why CNNs are suitable for this task:

1. CNNs impose local dependency which means the nearby signals in HAR are likely to be correlated – this makes for a good classifier.

2. CNNs are scale invariant which refers to the resultant scale-invariance for different paces or frequencies in the dataset, meaning that vastly different gaits can be accommodated with relative ease.

A very different approach is taken by Olivieri et al. (2012). They demonstrate a low-cost, home-based health care system based on automatic imaging recognition from video sequences. They propose a software package based upon a spatio-temporal motion representation, called Motion Vector Flow Instance (MVFI) templates, which capture relevant velocity information by extracting dense optical flow from video sequences of human actions. Automatic recognition is achieved by first projecting each human action video sequence, consisting of approximately 100 images, into a canonical eigenspace (i.e. dimensionality reduction), and then performing supervised learning to train multiple actions from a large video database. The MVFI is approximately 100% accurate in binary classification between fall activities and other actions Olivieri et al. (2012), where they show that their method is robust and can perform in real-time.

The study by Luštrek & Kaluža (2009) provides a thorough machine learning based approach to the problem at hand. Their method is achieved by equipping the user with infrared tags, from which the locations of body parts are determined, thus enabling posture and movement reconstruction. The authors compared eight common classification algorithms. Their goal was was to recognise six different activities. Gaussian noise was added to create clean and noisy data-sets to better reflect realistic signal collection.

For more recent studies, consider the work done by Albert et al. (2012). In their study, 15 subjects were asked to simulate four different types of falls; left and right lateral, forward trips, and backward slips, while wearing smart phones and dedicated accelerometers. Nine subjects also wore the devices for ten days, to provide data for comparison with the simulated falls. Five classification schemes were applied to a large time-series feature set to detect falls. Their results are robust, with both the Support Vector Machine (SVM) and the Sparse Multinomial Logistic Regression classifier, achieving accuracies close to 98% for pooled subject data when using 10-fold cross-validation, while that accuracy decreased to 97% when subject-wise cross-validation was used Albert et al. (2012).

Another recent, but very different approach to activity recognition, is employed by Mirchevska et al. (2013). Their paper presents a method for combining domain knowledge

and machine learning (CDKML) for classifier generation and online adaptation. A general CDKML schema – a method for combining DK and ML for classifier generation and online adaptation contains three phases:

1. Initialisation;
2. Refinement; and
3. Online adaptation.

The authors show that the classifiers developed after the first two phases are already more reliable and robust than ML classifiers built from limited examples from the domain of interest. Another option is to consider fundamental properties of the gait itself, to illicit information about falling.

⋆⋆⋆

We will demonstrate a simple combinations of methods to improve the classification accuracy in the work by Luštrek & Kaluža (2009), whilst using the same body-centered coordinate system and classification schemes. Their work is relevant to this study because their dataset features significant sections of missing information, such that 7.5% of their dataset is irretrievable due to sensor failure. But of principal importance is the dimensionality reduction which we shall investigate in order to improve classification performance, with particular attention being paid to their experiments where a feature vector with dimensionality of $\mathbf{x} \in \mathbb{R}^{720}$ was used. Smoothed data is found from Kalman smoothing (KS) which we describe in §3.3.3. On this data we implement several data-complete canonical transformations, with special focus on linear discriminant analysis (LDA), which are described in §3.4. Our experiments are outlined in §3.5 and finally a discussion and conclusion follow in §3.6.

3.3 Methods

We start by setting out the details of our analysis, demonstrating the variables used and the spaces they live in. Formally the problem can be stated as follows; assume there is a labelled training set $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $|\mathcal{S}| = N$, $j \in \mathcal{J} = \{1, \ldots, N\}$ and $\mathbf{x}_j \in \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are the task-specific feature vectors, with the activity (classes) given by $y_j \in \mathcal{Y} = \{1, \ldots, K\}$. The training set is such that $\mathcal{X} \subseteq \mathbb{R}^D$. A classifier is then

a function $h : \mathcal{X} \to \mathcal{Y}$, that maps an instance $\mathbf{x}_j$ to a label $\widehat{y}_j = h(\mathbf{x}_j)$. The accuracy of a classifier is evaluated using a loss function $\mathcal{L}(h(\mathbf{x}_j), y)$, which measures the disparity between the predicted actual label set (Dekel & Shamir, 2010). The problem at hand has multiple labels (activities) making it a multi-class classification problem. The following sections describe how the feature vectors $\mathbf{x}_j$ are chosen and evaluated.

3.3.1 Dataset

The dataset we use is the same as that used by Luštrek & Kaluža (2009), and is a motion capture (MOCAP) type dataset. The dataset was collected with a real time infrared MOCAP (Luštrek & Kaluža, 2009), consisting of six infrared cameras and infrared light sources. Three volunteers were equipped with 12 infrared reflectors. The markers were attached to the ankles, knees, hips, shoulders, elbows and wrists (see fig. 3.7). They were tracked with the cameras, and their three dimensional (3D) coordinates were measured. Artificial Gaussian noise was added according to the specifications of the system's manufacturer. The standard deviation of the noise was 43.6mm horizontally and 54.4mm vertically (Luštrek & Kaluža, 2009). Data was collected at 60Hz which was downsampled to 10Hz (to simulate typical smart phone sample frequency). The recording coordinate system was right-handed with the y–axis as the vertical axis and the other two axes aligned with the square walls of the room. A coordinate transformation was used to map the exogenous reference frame to an endogenous frame, where the y-axis passes through the two hip tags, the z-axis becomes the vertical axis, with the origin located between the two hip tags and finally the x–axis is normal to the yz–plane. *Eight* different short movement scenarios were repeated ten times by each subject:

- walking in a straight line;
- walking in a straight line whilst limping on the right leg;
- walking with a heavy burden in the right hand;
- walking in a circle, walking then stopping and resuming walking;
- falling in various ways and fashions, particular to each subject;
- lying down (which could be mistaken for falling);
- sitting down (which also could be mistaken for falling).

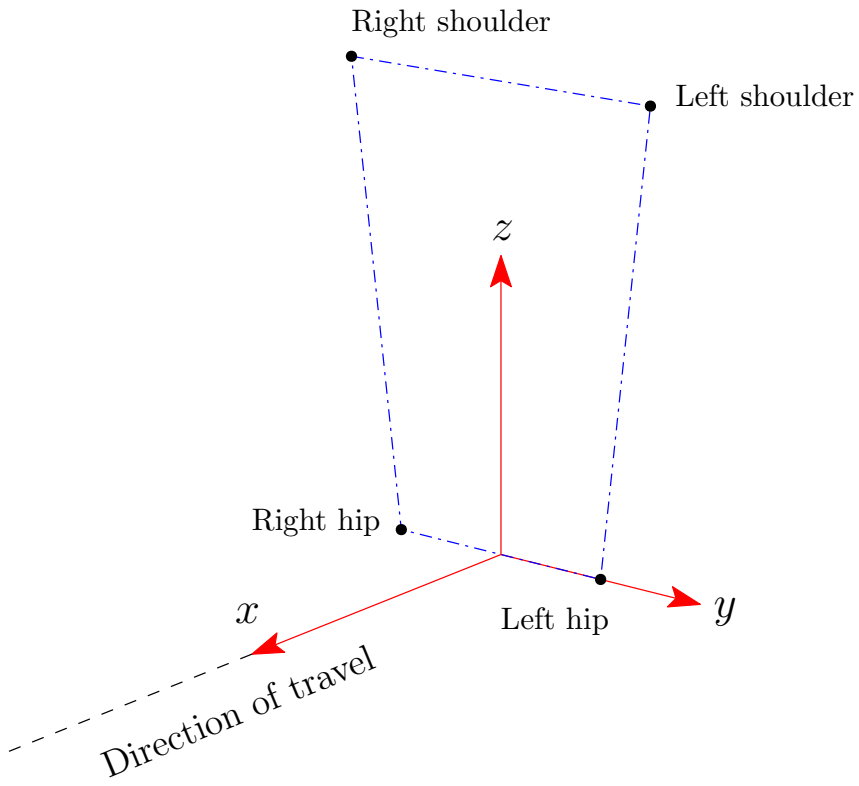


Figure 3.2: Body centred coordinate system, where the direction of travel is concurrent with the sagittal plane and the yz–plane is aligned with the frontal plane.

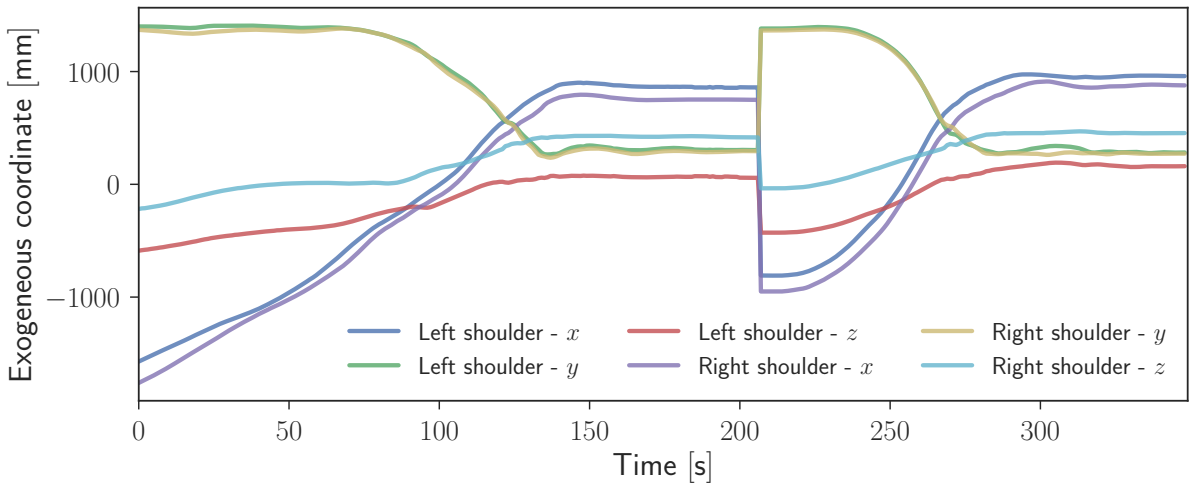


Figure 3.3: Recordings made from an infrared motion capture system. Window shows marker trajectories of the coordinates of two (out of 12) markers attached to the bodies of three volunteers (Luštrek & Kaluža, 2009). The scenario depicted includes three activities, enacted in the following order: *walking* → *falling* → *lying*.

Each scenario was labelled with one or more activities: falling, the process of lying down, the process of sitting down, walking, sitting (stationary) and lying (stationary).

3.3.2 Attribute Set

Let the collection of body tags be in the set $i \in \mathcal{I} = \{1, \ldots, 12\}$, where the attribute vector, from which the classifier infers the subject's activity, consists of ten consecutive snapshots of the subject's posture, describing one second of activity. We tried multiple snapshot counts (which is another way of saying which downsampling frequency we use), but it was found that ten worked best without adding lots of unnecessary information to the state-vector. Because there was little variability in the state, even under one second, there was little use in having a higher sampling rate.

Further, Luštrek & Kaluža (2009) used several attribute sets, clean and noisy, exogenous and endogenous. We focus our work on noisy observations of joint positions in endogenous coordinates because we envision ultimately, inferring pose from sensors attached to the body. Note that human behaviour needs to be captured by simple and general attributes, such that the classifier can generalise based on the general features used for training. More specific and complicated attributes could have been chosen, and they would have been classified with high accuracy but they would have not have generalised well, because of the high dimensionality required by the former. That being said, also in this domain, deep learning has had a profound impact. Consider the work by Ji et al. (2013) where the authors employ convolutional neural networks which enables them to automate the process of feature construction and hence present state of the art results, on representative classification tasks. Methods such as these have their own problems of course, data efficiency being one of them; where some deep learning methods require orders of magnitude more data than GOFAI methods. Data which typically needs to be labelled as well. An aside for sure, but worth bearing in mind whilst we continue with our hand-crafted features.

Hence, each feature consists of ten consecutive time-frames. Where it should be noted that each iteration of each scenario are not of the same length. Thus where the scenario time-length T, was non-integer, a floor function f was used to map the real-valued T to the largest previous integer value i.e. $f(T) = \lfloor T \rfloor$. Where the labels were mapped accordingly such that each 1s feature vector had one label, and any feature vector where T <1s was dropped. The training data set comprises N feature vectors. Sensor information is collected in an endogenous reference frame.

Let $\mathbf{e}_i^t$ denote the coordinates of the arbitrary tag i at time-frame t. The feature vector $\mathbf{x}_j$ is then designed by letting

$$\boldsymbol{\psi}_{i,j}^t = \left[\mathbf{e}_i^t, \|\dot{\mathbf{e}}_i^t\|, \alpha_i^t, \beta_i^t\right]_j^{\mathsf{T}} \tag{3.1}$$

$$\mathbf{x}_j = \underbrace{\left[\boldsymbol{\psi}_{1,j}^1, \ldots, \boldsymbol{\psi}_{12,j}^1, \ldots \boldsymbol{\psi}_{i,j}^t \ldots, \boldsymbol{\psi}_{1,j}^{10}, \ldots, \boldsymbol{\psi}_{12,j}^{10}\right]}_{\text{1s of activity}}. \tag{3.2}$$

Each sub-feature in the larger $\mathbf{x}$ vector is designated by $\boldsymbol{\psi}_{i,j}^t$ where the superscript t denotes which time-frame is included, and subscript i denoting which tag is included and finally j shows which feature vector they all belong to. In detail

$$\mathbf{e}_i^t \in \mathbb{R}^3 \tag{3.3}$$

$$\|\dot{\mathbf{e}}_i^t\| = \frac{\sqrt{(\Delta\mathbf{e}_i)^{\mathsf{T}} \cdot (\Delta\mathbf{e}_i)}}{\Delta t} \in \mathbb{R}^1 \tag{3.4}$$

$$\dot{\mathbf{e}}_i^t = \frac{\Delta\mathbf{e}_i}{\Delta t} \in \mathbb{R}^3 \tag{3.5}$$

where α and β are the angles of movement between the tag and the z–axis, and the tag and the xz–plane respectively. A summary description is given in table 3.1

Table 3.1: Description of the sub-features used to create the feature vectors.

Feature	Sub-feature description
$\mathbf{e}_i^t$	Spatial coordinate of tag i at time t
$\|\dot{\mathbf{e}}_i^t\|$	Length of vector $\dot{\mathbf{e}}_i^t$ divided by change in time
$\dot{\mathbf{e}}_i^t$	Change in components of $\mathbf{e}$ divided by change in time
α	Angle of movement between the tag and the z–axis
β	Angle of movement between the tag and the xz–plane

Having constructed the feature vectors, we solve the problem of missing data, by smoothing the original dataset, from which the feature vectors are generated.

3.3.3 Kalman smoothing

For an incisive background of state-space modelling, the reader should refer to the preliminary material in §2.4. Therein we explain that Kalman filters are typically used for online inference problems. In our problem domain, we can go one step further and condition on past *and* future observations (i.e. the tag coordinates), leading to our uncertainty being significantly reduced and our posterior state beliefs (i.e. the missing tag coordinates

due to sensor failure) improved (Murphy, 2012). For a visual intuition of the filtering and smoothing problem, consider the visual depiction adapted from Särkkä (2013) in fig. 3.4.

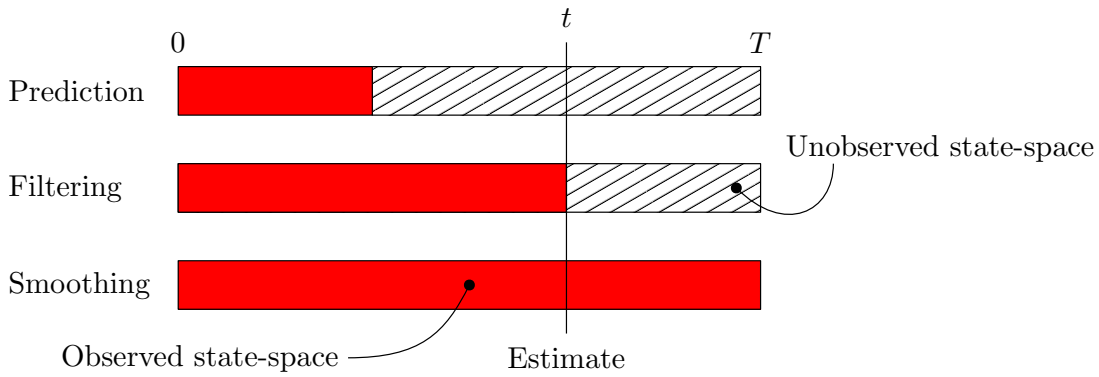


Figure 3.4: The optimal state estimation problem, visually depicted. The state estimation problems can be divided into optimal prediction, filtering and smoothing. The full domain is indexed between 0 and T. Adapted from (Särkkä, 2013).

Because linear Gaussian state-space models (also known as linear dynamical systems) can be represented by a tree-structured directed graph, inference problems are solved efficiently using the sum-product algorithm (Bishop, 2006), the forwards and backwards recursions which are known as Kalman smoothing. We begin by introducing the set

$$\mathcal{E}_t \triangleq \{\mathbf{e}_1^t, \ldots, \mathbf{e}_i^t, \ldots, \mathbf{e}_{12}^t\} \tag{3.6}$$

which constitutes the full set of raw coordinates as measured by the MOCAP system s.t. $\mathcal{E}_t \in \mathbb{R}^{36}$. The smoothing exercise seeks estimate $\mathbb{P}(\mathbf{z}_t \mid \mathcal{E}_{1:T})$, where $\mathbf{z}_t$ is the latent set of tag coordinates at time t (i.e. the missing ones whose magnitude is of interest).

Because the model has linear-Gaussian conditional distributions, the transition and emission distributions (which define a first order Markov model), of the state and observations can be written (Bishop, 2006) in the general linear form

$$\begin{aligned} \mathbf{z}_1 &= \boldsymbol{\mu}_0 + \mathbf{q} & \mathbf{e} &\sim \mathcal{N}(\mathbf{q} \mid \mathbf{0}, \mathbf{P}_0) \\ \mathbf{z}_t &= \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t & \mathbf{w} &\sim \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \boldsymbol{\Gamma}) \\ \mathcal{E}_t &= \mathbf{C}\mathbf{z}_t + \mathbf{v}_t & \mathbf{v} &\sim \mathcal{N}(\mathbf{v} \mid \mathbf{0}, \boldsymbol{\Sigma}) \end{aligned}$$

conditioned on the available data $\{\mathcal{E}_t \mid t = 1, \ldots, T\}$. We determine the parameters of the model $\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{P}_0\}$, using maximum likelihood through the expectation-maximisation (EM) algorithm. We briefly review it here for completeness, it aims to

maximise the likelihood function

$$\mathcal{L}(\boldsymbol{\theta} \mid \mathcal{E}_T) = \max_{\boldsymbol{\theta}} \mathbb{P}(\mathcal{E}_T \mid \boldsymbol{\theta}). \tag{3.7}$$

where $\mathbf{Z}_T = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ and $\mathcal{E}_T \triangleq \{\mathcal{E}_t \mid t = 1, \ldots, T\}$. The EM algorithm then works by iteratively evaluating

$$\mathbb{P}(\mathbf{Z}_T \mid \mathcal{E}_T, \boldsymbol{\theta}^{\mathsf{old}}) \tag{3.8}$$

by maximising

$$\boldsymbol{\theta}^{\mathsf{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}|\mathcal{E}_T, \boldsymbol{\theta}^{\mathsf{old}}} \left[\log \mathcal{L}(\mathcal{E}_T, \mathbf{Z}_T \mid \boldsymbol{\theta})\right] \tag{3.9}$$

until convergence is reached, if not, the old parameter values are updated by the new ones. Where the initial state mean and covariance matrices were initialised as identity matrices (Duckworth, 2012). The EM algorithm is discussed in much greater detail (Bishop, 2006, p. 440).

3.4 Dimensionality reduction

For each scenario iteration (ten for each subject) $\boldsymbol{\theta}$ was found through likelihood maximisation. Missing information, i.e. tag coordinates due to sensor-failure, were inpainted from the linear dynamical system, and feature vectors were created from this smoothed dataset. As noted in the previous section, we focus our attention on a specific feature set used by Luštrek & Kaluža (2009), where $\mathbf{x}_j \in \mathbb{R}^{720}\ \forall j$, and $N = 1,302$. The authors avoid overfitting by using cross-validation and regularisation. We will investigate the latter further by investigating task-specific regularisation by way of dimensionality reduction (DR) through canonical transformations. Six methods were investigated:

- Multiclass linear discriminant analysis (MLDA)
- Principal component analysis (PCA)
- Factor analysis (FA)
- Truncated singular value decomposition (TSVD)
- Gaussian random projection (GRP) and
- Partial least squares regression (PLSR)

We provide a synopsis of MLDA, as it was found to increase classification accuracy the most (see §3.6), the other methods are briefly recounted in §2.1. We seek projection vectors $\mathbf{w}_k$, $k \in \{1, \ldots, |\mathcal{Y}| - 1\}$, arranged by columns in a projection matrix $\mathbf{W}$ s.t.

$$\overline{\mathbf{x}}_j = \mathbf{W}^\mathsf{T} \mathbf{x}_j, \forall j.$$

We are looking for a projection that maximises the ratio of between-class to within-class scatter. It can be shown (Farag & Elhabian, 2008) that the optimal projection matrix $\mathbf{W}^*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues λ_k, of the following generalized eigenvalue problem

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^\mathsf{T} \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^\mathsf{T} \mathbf{S}_W \mathbf{W}|},$$

where $\mathbf{S}_W$ is the within-class scatter and $\mathbf{S}_B$ is the between-class scatter. The projections with maximum class separability information are the eigenvectors corresponding to the largest eigenvalues of $(\mathbf{S}_W^{-1}\mathbf{S}_B - \lambda_k)\mathbf{w}_k^* = 0$, where $\mathbf{w}_k^* \subset \mathbf{W}^*$ are the columns on the optimal projection matrix. We can also briefly consider PCA, which performs a linear transformation of the variables into a lower dimensional space which retains the maximal amount of information about the variables, that is to say that PCA relates where the largest variance in the dataset lies. Thus, using PCA it was found that the first four principal components, of the smoothed coordinate data, accounts for 81% of the variance (the first five: 88%).

3.5 Experiments

Using our smoothed data and dimensionally reduced feature vectors, we compare performance with Luštrek & Kaluža (2009). In their study the authors used eight different classification schemes:

- Pruned C4.5 Decision Tree (C4.5)
- Propositional Rule Learner (PRL),
- Naive Bayes Classifier using Estimator Classes (NB)
- 3-Nearest Neighbours (3-NN)
- multiclass Support Vector Machine (SVM)

- Random Forest (RF)
- Bagging of the fast decision tree learner (Bag.) and
- Boosting of the fast decision tree learner (M1).

We sought to improve the worst performing classification schemes as implemented by Luštrek & Kaluža (2009), in order to train classifiers to correctly label six different kinds of activities. A synopsis for each classifier is provided in §2.2.

3.5.1 Raw observation classification

We also demonstrate DR operations on the raw Kalman smoothed observations directly, and then use the same for classification, without creating any feature vectors. Here we also note that there is an argument for using an autoregressive (AR) model in this for imputation. The AR model imposes that the output variable depends *linearly* on its own previous values and on a stochastic term. Imputation with the AR model is a common approach due to its inherent simplicity and ease of implementation but given the choice of a Kalman filter (or smoother) and an AR model, the latter should be the preferred option. Before going into the theoretical reasons for this, consider some empirical evidence, based on the work of our peers.

Choudhry & Wu (2008) compared four different generalised autoregressive conditional heteroskedasticity (GARCH) models and the Kalman filter. GARCH models are more complex instantiations of the humble AR model. Forecast errors based on company daily stock return forecasts are employed to evaluate out-of-sample forecasting ability of both GARCH models and the Kalman filter. Choudhry & Wu (2008) find that "measures of forecast errors overwhelmingly support the Kalman filter approach". Fulton et al. (2001) focused instead explicitly on the missing data problem, where the authors compared the Kalman smoother to the AR on several difficult imputation problems, and found much the same results as Choudhry & Wu (2008). Finally, Anava et al. (2015) also consider the problem of time series prediction in the presence of missing data, they are interested in the online learning problem in which the goal of the learner is to minimise prediction error. Even in this very complex domain, the Kalman filter is still competitive, especially as the authors have to derive new algorithms for AR predictions, as these are not well defined when data is missing.

These results are not unexpected since the Kalman filter is *an optimal estimation algorithm* (Kalman, 1960). The estimate is optimal in the sense that the mean value of the sum (or any linear combination) of the estimation errors produce a minimal value. This is under the assumption that our observations are linear and Gaussian (which they are). But since these are our assumptions, the estimate (or imputed value) will be optima under this model, which is why an AR process cannot supersede the estimation quality provided by the Kalman filter (nor the Kalman smoother).

⋆⋆⋆

We investigate the raw data on its own, without building 1s long feature vectors, in order to determine how informative the endogenous coordinate measurements are on their own, and if they can in any way inform us how to adequately enable informative feature selection for classification. Recall, that Luštrek & Kaluža (2009)'s feature selection was somewhat arbitrary. Theirs was designed to generalise over multiple activities, stretching beyond the activity set demonstrated in their paper. They did not, however, demonstrate if this capability was achieved by e.g. holding out activities from the training dataset.

3.5.2 Single tag classification

In the second part of our experiments, information redundancy was investigated and physical dimensionality reduction studied. Now $\mathbf{x}_j \in \mathbb{R}^{60}$, instead of using the full set of tags, each tag was classified individually in order to ascertain which tags were most informative in terms of activity recognition, upon which MLDA was implemented, see fig. 3.7. For all experiments accuracy was computed using ten-fold cross-validation, regularisation and each classification scheme was repeated ten times, yielding 100 folds for each algorithm. Where N remained the same for all experiments in this paper.

3.6 Results

Consider the classification results for the raw Kalman smoothed data, where each observation $\mathbf{x}_j \in \mathbb{R}^{36}$ and $N = 13,050$, fitted with dimensionality reduction models as discussed earlier. Figure 3.5 shows a scheme-by-scheme *max* accuracy classification comparison, between the six dimensionality reduction methods. The choice of max accuracy is deliberate, because this is the type of score that Luštrek & Kaluža (2009) report and as will be clear,

this is not a very useful metric, since we do not receive a distribution but only a point estimate. Point estimates, for tasks such as ours, are not useful because most of the methods involved have an element of randomness. If these were fully deterministic methods, then it would make more sense to report point estimates but since they are not, reporting distributions is the only useful measure, and this what we go onto report in fig. 3.6.

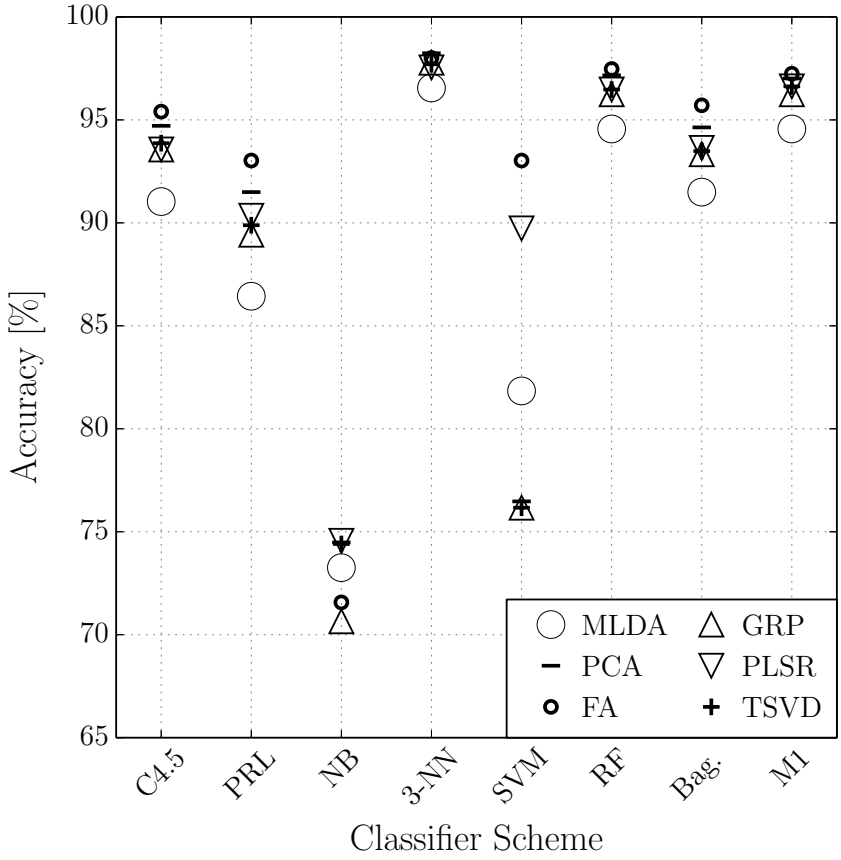


Figure 3.5: Classification of dimensionality reduction models, applied to raw Kalman smoothed 3D coordinate data (no incorporation into feature vectors). Only max results are reported as discussion as the start of §3.6.

The results classify the pose with a frequency of 10Hz, instead of concatenating ten consecutive poses into one feature vector. We must consider if endogenous 3D coordinates are representative of the user's activity. Although coordinates are simple attributes, they are possibly too simple. Because our activity labels are comparatively simple motions, it is unclear whether coordinate attributes would generalise well if e.g. an additional label of running was added. The classifier would then have to separate walking and running activities in pose space. Hence although it is an attractive option to directly classify the raw data using merely a DR model (computationally very cheap), it is possibly too simple

a parametrisation of human pose, which will most likely not generalise well if the activity space were to be increased.

Consider now the results from dimensionally reducing the feature vectors. Results are summarised in fig. 3.6, where the full distribution of the classification accuracies have been summarised in a box plot for MLDA, because it showed that the best performance, compared to the others schemes (of which only the max accuracy is shown in fig. 3.6). A like-for-like comparison is only possible by considering the best accuracies in our experiments. These results can be seen at the maximum whiskers of each MLDA box plot in fig. 3.6, they are summarised in table 3.2.

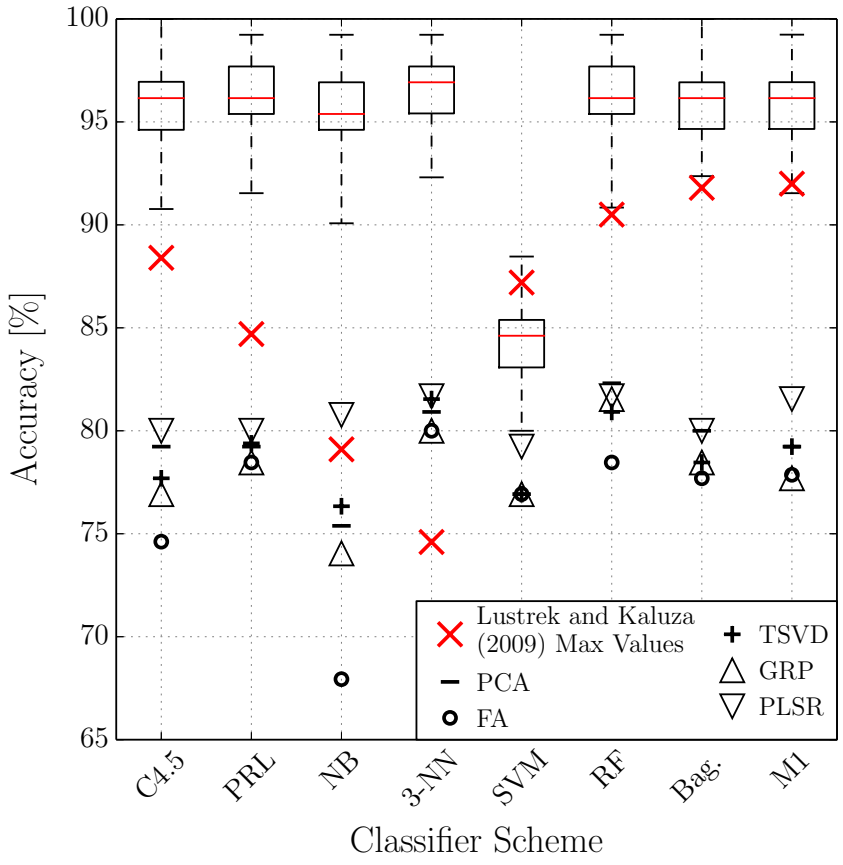


Figure 3.6: Classification performance shown for all eight schemes, for all six dimensionality reduction methods. MLDA results, being the best, are shown as box-plots over all folds. Luštrek & Kaluža (2009) only reported the best accuracies for their experiments, and not the distribution over all their folds, which is why only one point per scheme is shown.

It should be noted though that this form of comparison is not representative nor statistically significant. Ideally Luštrek & Kaluža (2009) should have reported the distribution over their classification accuracy, over the various algorithms, and then quoted the median of the latter. In these experiments 100 cross-folds were used, which allow for a median to be

Table 3.2: Classification scheme (max) accuracy [%] comparison , where the best results are shown in bold.

Study	C4.5	PRL	NB	3-NN	SVM	RF	Bag.	M1
Luštrek & Kaluža (2009)	88.4	84.7	79.1	74.6	87.2	90.5	91.8	92.0
Our methods	**100.0**	**99.2**	**99.2**	**99.2**	**88.5**	**99.2**	**100.0**	**99.2**

quoted, a median which is more representative of the true accuracy of each classification algorithm. This is because the accuracy is reported across *all* folds, not just the best one, thus taking account the variance in the dataset.

Luštrek & Kaluža (2009) only reported the best accuracies for their experiments, and not the distribution over all their folds, why only one point per scheme is shown in fig. 3.6. As is seen in table 3.2, the LDS, which also doubles as a generative model, used for inferring missing data, produces data of high accuracy which validates its use as a generative motion model, the outputs of which function well as viable substitutes for classification. An accurate generative model which can be sampled accordingly to infer pose continuously from low-rank observations, has immediate practical utility since data collection becomes easier, faster and negates the use of complex feature selection to facilitate high classification accuracy. This means that equipping the user with the simplest of collection devices (e.g. a mobile phone), could be enough to infer complex motion and pose, and hence if they are falling or are about to.

That being said, the increased classification accuracy is more likely to have been derived from dimensionality reduction methods. As can be seen multiclass MLDA performs particularly well where MLDA: $\mathbf{x}_j \in \mathbb{R}^{720} \rightarrow \mathbf{x}_j \in \mathbb{R}^5 \; \forall j$. Recall again that the size of projection space is chosen by selecting the eigenvectors with the largest eigenvalues as explained in §3.4 – in this case we selected eigenvectors which explained 90% of the variance. MLDA preserves as much of the class discriminating information as possible, by explicitly modelling the difference between them, and thus finding a linear combination of features which separate the activities. By using a new basis we project the dataset onto a dimensional space with more powerful data representation. We are performing offline inference, hence the means and covariances are known, making this method particularly suitable for our chosen application domain. Complex data structure is preserved since the distributions of the attributes in the feature vectors, are significantly Gaussian (a

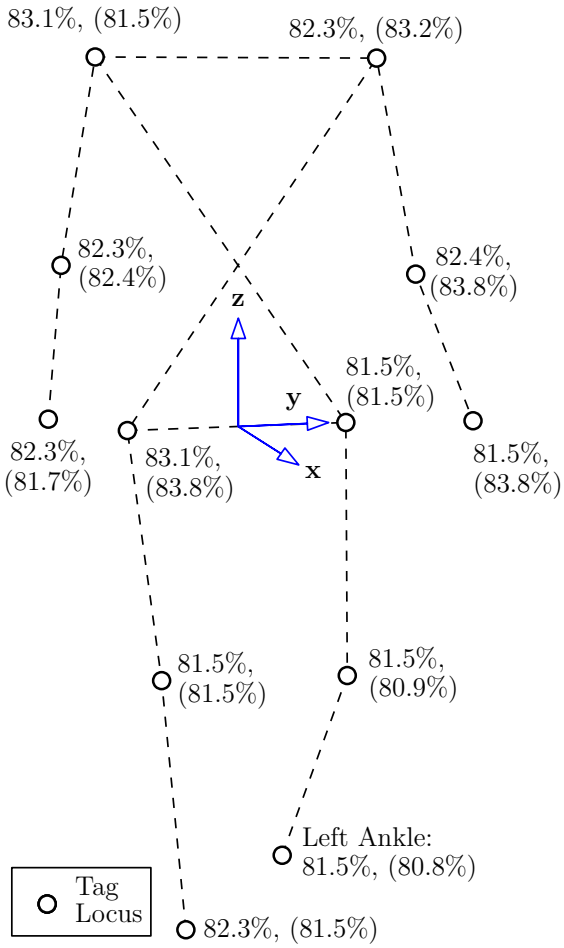


Figure 3.7: Time frame illustration from a walking activity, with best individual tag classification accuracy quoted with each tag for feature vector of size $\mathbf{x}_j \in \mathbb{R}^{60}\ \forall j$, and of size $\mathbf{x}_j \in \mathbb{R}^{5}\ \forall j$, within parentheses. Adapted from (Dhir & Wood, 2014).

requirement for MLDA to perform well and a key assumption of the MLDA method) and discriminatory information is found in the mean of the data and not in the variance. Further, as can be seen from fig. 3.6, other smoothing methods have been adopted and tested.

However, since our method performs offline inference (recall: we are trying to determine if a fall has *happened* not if it is *happening*), in order to use both the forward and backward pass of the sum-product algorithm, we use backdated data. But this requires the use of several body-poses at several (ten in our case, leading to one second of activity) points in time, which is ultimately what makes the feature vectors high dimensional. We overcome over-fitting by using ten-fold cross-validation (CV), where the data is used efficiently by varying our learning and testing sets, through CV, and thus selecting a model that is general and but not exact.

What happened to the SVM? As fig. 3.6 shows and table 3.2 emphasises; the support vector machine performs very badly on this classification task, compared to the other methods. First though, note that we are still comparable to the results of Luštrek & Kaluža (2009) so we can be fairly certain that here are no methodological mishaps that have entered our experimental pipeline. Rather it is more persuasive to consider the SVM in itself, though we caution that this is a difficult question to answer. First, SVMs not usually good for imbalanced datasets such as ours, where instead k-NN show much greater promise. This could be because having an under represented class, will result in a poorly defined class frontier which the k-NN can handle much more robustly (i.e. it can generate a class separator well, with few examples). Secondly, we are using a baseline SVM which assumes that our data is linearly separable – this may not be the case, and instead we could see improved results by using an SVM with a (e.g. polynomial) kernel. It is likely that the true underlying cause is a combination of both of these issues, and using our suggestions for imbalanced data discussed earlier, and a non-linear kernel, we stand a greater chance of getting good results with the SVM.

Further, we have yet to discuss the implications of the coordinate frame that we have chosen to investigate. As stated earlier in this chapter, we operate on coordinates recorded in an endogenous (body-centric) reference frame – but the coordinates were recorded in an exogenous (space-based) reference frame. Consider first; *would classification accuracy improve for either reference frame?*. The answer, simply, is 'no'. This is because ultimately what yields good classification accuracy are the quality of the observations and consequently the quality of the sensors (in this instance, cameras). The reference frame from whence they are recorded has little or no bearing on that quality since ultimately, it is merely a transformation of observation reference, not a transformation of the observations themselves. Consequently we can conclude that placing a reference endogenously is beneficial from a user perspective (because that reference is more natural alongside the human control frame), but confers more ease of collection from an exogenous perspective.

In the second part of our experiments, our feature vectors are still high dimensional ($\mathbf{x}_j \in \mathbb{R}^{60}\ \forall j$). But we investigate what can be considered physical dimensionality reduction, by treating each tag as independent and running the smoothing model and the classifiers on each independently, where no information was passed between tags. The aim of this exercise is to answer the question: *which tag yields best falling incidence information?* The

original and dimensionally reduced classification results are shown in fig. 3.7. The results are not as good as in part one. First, the LDS does not perform as well, owing to the lack of information passed to the model from the other tags, resulting in less exact inferred pose predictions. Moreover, the amount of information contained in one second of activity, or ten sequential body poses, is not enough to produce classifiers which are discriminative enough to accurately categorise the activities. Having an average generative model coupled with an average discriminative classification performance, even with MLDA (the minor difference between classification accuracy between MLDA and the original feature vector size, would suggest that dimensionality reduction is not the foremost problem), suggests that other features need to be considered for single tags, or more tags used for these features in order to maximise the utility of the information used for classifying human motion behaviour.

3.7 Conclusion

In this chapter, we have adopted standard classification and dimensionality reduction methods to solve a simple but important problem. We sought to improve fall detection mechanism by treating our problem domain as one of HAR. This has been addressed many times before, as we demonstrated in §3.2. We demonstrate that this engineering challenge can be better solved, by first completing the dataset and then removing the unnecessary parts of it, allowing the classifiers create more appropriate class boundaries.

Further, it is clear that certain tag locations are better than others for activity classification (see fig. 3.7), and decision boundaries in $\mathbb{R}^{60}$ are naturally easier to define with a hyperplane than in $\mathbb{R}^{720}$ which is why classification drastically improves. But these results must be viewed in the light of their dependency on information from the other tags, owing to the nature of Kalman smoothing. Nonetheless two scenarios can be considered: first all tags are used but then the majority of the information is used to smooth only one tag which will be the primary location for activity recognition. Secondly, a robust and highly reliable registration sensor could be envisioned, if such reliability was to be had then multiple tags for activity recognition would become redundant. Another scenario could be considered whereby only a small tag subset is used (e.g. the combination of the highest accuracy tags shown in fig. 3.7). Finally, it is clear that the amount of training data was not enough,

given the disparity of classification accuracy between the left and right-hand tags as seen in fig. 3.7. If the amount of training data was to increase then both sides would most likely see more similar accuracy results. Moreover it was found that various algorithms classified various individual tags more accurately than others. For example, the right-shoulder tag had the best accuracy using the Random Forest scheme but the left-elbow tag had better accuracy using the 3-NN algorithm.

Further, it is worthwhile noting that the simple methods that we present trivially generalise across subjects in the dataset. Figure 3.7 displays the results for one subject, but under this sensor modality the results are comparable for all subjects in the dataset since the dimensionality reduction methods and the classification schemes, are invariant to subject morphology. This would suggest ample opportunity for transfer learning, where one may posit that we learn classifiers for one subject, and then re-use these on another subject. Given that we discussed the invariance of the employed methods to subject morphology, this suggests that transfer learning is imminently possible as classifiers should, with ease, be re-usable for multiple subjects. This further suggests a fast strategy for learning a more comprehensive set of classifiers which can be learned individually, for multiple activities, and then combined to create a model able to differentiate between multiple activities (more than the eight considered herein).

⋆⋆⋆

Having dealt with an instance of relevant incidence detection and classification, we must now turn to its relevance for an active prosthesis. As noted in §3.1, there is utility in a prosthesis being able to partake in the HAR loop, of which fall detection is part. But that is under the proviso that a classifier has a priori been set with a fixed number of classes. If we seek to develop adaptive prostheses, which have the ability to learn from their surroundings (irrespective of the sensor modality), then we cannot bound the available activity space. Much like humans learn new skills, so too must anthropomorphic prostheses. As such the device must be able to adapt nonparametrically. In other words, it needs to have the functionality to do *unsupervised* segmentation of time-series measurements (not necessarily online, learning can be offline for later usage) and assign utility and labels to those observations. We shall deal with the former problem in the next chapter, where we consider Bayesian nonparametric time-series segmentation.

CHAPTER 4

Dynamics identification via time-series segmentation

Contents

'Proprioception' refers to the human ability to sense movement within joints and joint positions (Mosby, 2013). It is a remarkable property that allows us to know where our limbs are in space – which is to say our sense of the relative position our body parts. Proprioception also extends to our muscles, where it is described as the strength of effort employed by our muscles to effect locomotion (Mosby, 2013). Modern 'intelligent' systems, such as robots (e.g. Honda's Asimo) do not have proprioception, as they do not 'sense' the relative position of their 'limbs' (parts) and joints, instead they measure where they

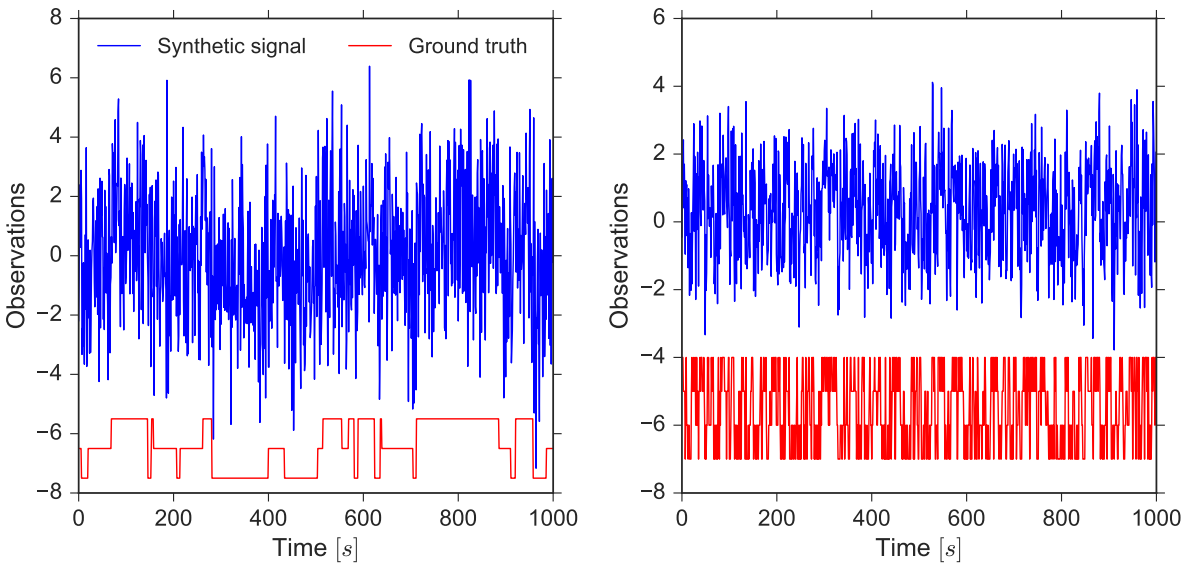
are in their space of operation. Asimo and its kin, have position sensors which allow them to determine where their limbs are at any point in time. But, they cannot as yet perceive where their limbs are as humans do, nor do their actuators follow a strict proprioceptive means of operation, instead predominantly employing trajectory matching.

Proprioception is inimical to human learning. When humans learn through touch, they effect automatic labelling (Han et al., 2011b) of their operating environment, and in so doing pursue a strategy of active learning. In active learning the learner interactively chooses which data points to label (a human agent may choose not to re-label objects which he or she already has sufficient knowledge of).

On the other hand, in order for an intelligent system to make sense of the world, they would be required to maintain a full set of labels, each of which corresponds to a single stimulus, or to a combination of sensory stimuli. Having to maintain such a look-up table is not a realistic requirement for any agent, operating in this complex modern world, nor is it strictly feasible purely from an information theoretic and engineering point of view. Thus, what we are really interested in, are methods for automatically segmenting a time-series (of e.g. force sensory measurements) into a set of time-intervals that have some useful interpretation in some basal domain (Fox et al., 2008, §1). That domain, in our aforementioned example, would be the agent's understanding of the world. That domain, for our proposed application, is a prosthesis able to segment and identify new movement, fully unsupervised, with an unbounded state-space.

Time-series appear in far more domains than robotics though, they are as ubiquitous as the phenomenon that gives rise to them. Hence, there is considerable benefit to be gained from being able to automatically 'understand' (or in the very least, segment) complex sequential data, such that its interpretation is rendered useful. Consider, e.g. the noisy synthetic observations shown in fig. 4.1, the ground-truth in fig. 4.1a is not easily extracted from that simple univariate signal, and harder still in fig. 4.1b. This problem arises time and time again in as varied fields as genomics, finance, machine translation, activity recognition to mention but a few (Barber, 2012; Murphy, 2012; Bishop, 2006; MacKay, 2002; Beal & Krishnamurthy, 2012). Furthermore, there is of course the option of disregarding the time-domain altogether, as noted by Taylor (2009, §2). We posit however, that it would be most unwise to do so given the integral part that it plays in most things, from human behaviour to weather systems. Applying static models to time-series measurements would

disregard, and indeed throw away, much of the richness of the observations themselves.



(a) Slow dynamics **(b)** Fast Dynamics

Figure 4.1: Slow and fast evolving synthetic observations in blue (top), where the true evolution of the process is shown in red (bottom).

Without being too restrictive on the sort of tasks we seek to accomplish in this chapter, it is worth echoing the traditional goals in signal processing w.r.t. time-series analysis. As noted by Weigend & Gershenfeld (1994), Barber (2012, §23.2.1) and Taylor (2009, §2), there are typically three goals:

- ➢ Prediction or forecasting — short term inference over the future evolution of the system;
- ➢ Characterisation or identification — finding the minimal underlying state cardinality and inferring the system parameters of the underlying model; and
- ➢ Modelling — being able to describe, as accurately as possible, the long-term behaviour (or steady-state) of the system under study.

We seek to address these issues using *discrete-state* Bayesian nonparametric state-space models (and related methods), the specifics of which deal with the two most cited drawbacks of the hidden Markov model (a discrete-time state-space model):

I *State duration distributions are necessarily restricted by the geometric shape* $\mathbb{P}(d) = a^{d-1}(1-a)$ (and, in particular, monotonically decreasing), where d denotes the

duration of a given state and a denotes its self-transition probability, which is not appropriate for many real-world problems like animal behaviour modelling (for instance, a human being usually sleeps for roughly 7-8 hours and, certainly, the probability of those sleep durations are not monotonically decreasing in the durations).

II The *number of hidden states must be set a priori* while one of the main objectives of behavioural research is to discover new or more detailed behavioural patterns. This is also known as the *model selection* problem for HMMs.

With that in mind, our interest in this domain is formulated through the following problem statement, which we qualify with a method motivation followed by our proposed solutions.

Problem statement In an unsupervised segmentation setting, we aim to learn a set of meaningful discrete states inherent from the time-series observations. Better yet, we aim to learn a set of states that have well defined discriminators, making it easy for the user to distinguish one state from another. One model, which does this well, is the hidden Markov model. But it is too restrictive in its fixed form as demonstrated in item **I** and item **II**. There is immediate practical utility in being able to define hidden Markov models with unbounded state cardinality which admit complex duration distributions. In a human activity recognition task, this would mean being able to classify human-motion classes, without specifying their prior number (which we had to in chapter 3) or their prior respective duration. Hence we want to answer the following question:

1. How can we more efficiently address the shortcomings of the: basic HMM and recent advances in Bayesian nonparametric HMMs (Teh et al., 2006; Fox et al., 2008; Johnson & Willsky, 2013)?

2. We want to use the probabilistic programming paradigm to address approximate inference in our state-space models. How well does this inference strategy work?

3. Unsupervised learning through the methods of Bayesian nonparametrics, is an elegant modelling framework, but how well does it work in large-scale real applications?

Motivation To address item **II** we adopt the hierarchical Dirichlet process (HDP), which enables examination of HMMs with an unbounded number of discrete states. This is a tried an tested approach (Teh et al., 2006; Fox et al., 2007; Johnson & Willsky, 2010). However, the HDP introduces problems of its own. The HDP creates an unbounded state prior, which usually creates too many states and introduces switching dynamics that are usually too fast (Teh & Jordan, 2010). One way to get around this is to make state-switching parameters *stateful*, which ties them to a specific state, rather than making them global, and then learn them on a per-state basis. That improves performance for item **II**. To address item **I**, one can explicitly impose a parametric duration distribution, or a nonparametric duration distribution. We adopt the latter approach since it has the effect of making the model more flexible. For inference and learning we adopt the probabilistic programming paradigm and use particle Markov chain Monte Carlo methods, or general-purpose inference. This enables to quickly and reliably iterate over different model specifications, without having to derive a new inference scheme for each iteration.

Contributions To address the above problems in item **I** and item **II**, we derive the following models: the stateful HDP-HMM (Dhir et al., 2016b), the infinite duration HMM (Dhir et al., 2017c) and the stateful infinite duration HMM (Dhir et al., 2017c). For learning and inference we used the probabilistic programming paradigm and specifically *Anglican* (Wood et al., 2014) and also use Bayesian optimisation to find model parameters. We demonstrate a comprehensive set of experimental results in §4.4. In the latter section, we apply our models to novel application domains including HAR and to demonstrate utility outside that domain, we also investigate lion ecology modelling.

⋆ ⋆ ⋆

The material presented within is conditional on prior work, which can be found in the preliminary material. The following subsections have relevant dependencies:

- ➢ State-space modelling is employed in §4.3 and should be set against §2.4.
- ➢ Bayesian nonparametrics is heavily used in §4.3, the details of which are introduced in §2.7.
- ➢ Probabilistic programming is used throughout but demonstrated primarily in §4.4. Background work and algorithm exposition is shown in §2.6.

➢ In the empirical evaluation §4.4.2, Bayesian optimisation is applied, which is explained in §2.3.

4.1 Related work

Our work builds directly on top of that by Teh et al. (2006); Fox et al. (2008); Johnson & Willsky (2010) and the models therein. As such we provide an incisive exegesis of those models directly in the following sections. Herein, we shall deal with other, related, Bayesian nonparametric state-space modelling techniques, which have recently appeared in the literature as well as the foundational works that led to this point.

⋆⋆⋆

The first notion of *infinite* hidden Markov models (iHMM), was introduced by Beal et al. (2001). In this work, the authors were the first to extend the standard HMM to have a countably infinite[1] number of hidden states. The authors use a hierarchy of Pólya urn constructions of the DP to describe a generative model with an unbounded state space. In particular, they use they model each row of the transition and emission matrices of the HMM as a DP. Inference and learning was provided by sets of Gibbs samplers, which inferred the latent state sequence and the model hyperparameters. This is the model that Teh et al. (2006) re-interpreted as a strict hierarchical DP (HDP), leading to new stick-breaking and Chinese-restaurant process (Chinese restaurant franchise) representations, and hence the introduction of the HDP-HMM. They, too, used various forms of Gibbs samplers for their inference and learning. Interestingly in the years following the publication of (Teh et al., 2006) it was not clear if the iHMM of Beal et al. (2001) and the HDP-HMM of Teh et al. (2006), were in fact the same model. Fortunately that conundrum was settled by Van Gael (2011, §3.1.3), when he proved that they were in fact the same.

Around the time of the publication of those works, there were also a number of other approaches used, which specifically sought to address item **II** in the aforementioned list. Their general approach was to train several HMMs, with different numbers of states, the best of these being then chosen according to some criterion (Siddiqi et al., 2007, §1).

[1] We say that a set is *countably infinite* if its elements can be put in one-to-one correspondence with the set of natural numbers $\mathbb{N}$. Hence, one can count off all elements in the set in such a way that, even though the counting will take forever, one will get to any particular element in a finite amount of time (Weisstein, 2000).

This approach is unappealing for two reasons. First, it is computationally expensive to iterate over multiple model topologies, in search for the 'best' one (though of course, this task can be appropriately parallelised). The second problem arises precisely as a consequence of the first; because parameter learning is really just an optimisation in some high-dimensional space, it will be prone to local minima, thus leading to inconclusive results when comparing models. Other approaches of similar type seek to estimate the marginal likelihood and then perform model selection conditioned on those results. Van Gael (2011, §3.3.1) has an excellent discussion on this approach, but the main take-away from it is that because the marginal likelihood of Bayesian HMMs is intractable, we have to resort to advanced simulation techniques such as MCMC methods to compute it. One of the primary purposes of this thesis is to champion general-purpose inference methods, which is why advanced and bespoke inference schemes, such as beam sampling (Van Gael et al., 2008), are unsatisfactory for our purposes.

This review would be completely amiss if it did not continue with the thesis of Van Gael (2011), where the author's work *Bayesian nonparametric hidden Markov models* is pertinent to the current discussion. In this thesis the author contributes a "family of fast and exact Monte Carlo inference algorithms" for the aforementioned models. Furthermore, he introduces a new BNP building block called the Markov Indian buffet process, used to build a nonparametric extension of the factorial hidden Markov model (see fig. 4.2), called the infinite factorial hidden Markov model (iFHMM). Despite using bespoke inference, this is an impressive model, which produces state-of-the-art results on the blind-source separation problem. Similar to this work is the contribution by Heller et al. (2009), where the structure of the model is instead made flexible. Their infinite hierarchical HMM, allows sequential time-series modelling to be made more malleable by making the number of hierarchies unbounded. Inference is derived from a Gibbs sampler and a modified forward-backwards algorithm. They demonstrate impressive results on synthetic data, as well as on a number of real tasks, where they have shown that their model learns appropriate model complexity (i.e. the number of hierarchies) given the task at hand.

We shall examine in detail the sticky HDP-HMM by Fox et al. (2008), a more general version of it was introduced by Stepleton et al. (2009). The block-diagonal iHMM, presents an unsupervised method that learns HMMs with block-diagonal dynamic structure from time series data (Stepleton et al., 2009, §1). As with all the models under discussion here,

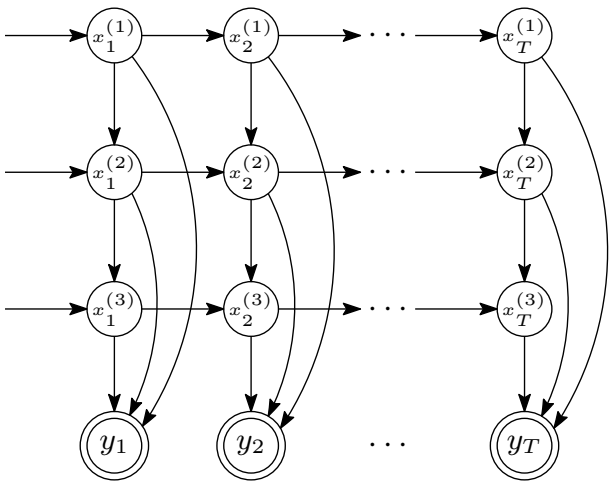
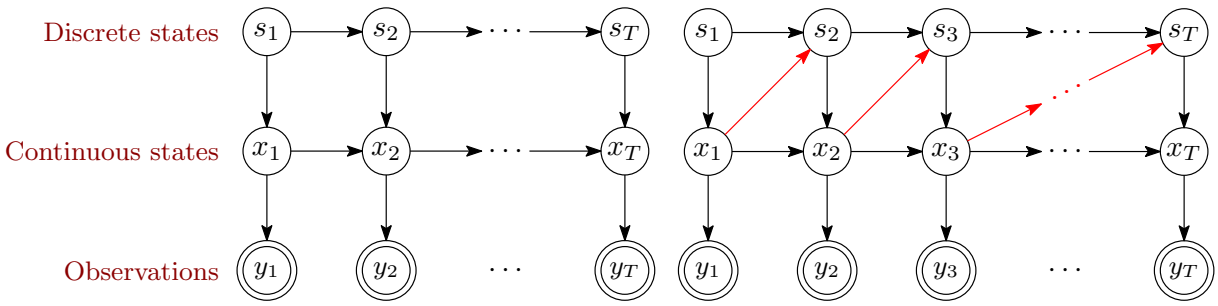


Figure 4.2: A factorial hidden Markov model where the latent node superscripts denote which factor layer they belong to. Continuous and unobserved states are denoted by x_t and observations by y_t as before.

they do not, in advance, specify the state cardinality, thus addressing item **II**. When the block is of size one, the sticky HDP-HMM is received. But like the state cardinality, they do not specify in advance the block-size either. Like Teh et al. (2006) they derive a bespoke Gibbs sampler for inference, and apply their methods to the problems of unsupervised learning of gestures in video clips as well as unsupervised learning of themes in music.

Finally, we shall consider some more recent models and inferences, some of which are of a rather different type than those we have encountered hitherto, manifested through the work by Saeedi et al. (2016); Johnson et al. (2016); Linderman et al. (2016). In the former, Saeedi et al. (2016) present the segmented iHMM (siHMM) the purpose of which it to identify transitions in time-series, wherein regime changes take place in the dynamics.In other words, identify changes between combinations of fast and slow dynamics as demonstrated in fig. 4.1. This can typically be done well with hierarchical HMMs (such as the HDP-HSMM) but at a great inference cost. Saeedi et al. (2016)'s innovation is to maintain a simple inference scheme, and still outperform more complex models such as the HDP-HSMM (Johnson & Willsky, 2013). 'Simple' here is relative of course, since they employ the stochastic variational inference algorithm by Hoffman et al. (2013) and then use inference ideas introduced by Johnson & Willsky (2013), to compose a tuple of model and inference that displays impressive performance on a variety of segmentation tasks.

A rather different but very interesting modelling and inference paradigm is introduced by



(a) SLDS (Murphy, 2012, §17.6) **(b)** rSLDS (Linderman et al., 2016)

Figure 4.3: Graphical models of switching linear dynamical systems. Continuous states are denoted by x_t, discrete by s_t and observations by y_t as before. We have omitted conditioning on model parameters to avoid clutter, but also to highlight the model dynamics.

Johnson et al. (2016) where the authors demonstrate a framework "that combines the complementary strengths of probabilistic graphical models and deep learning methods". By introducing what they call *structured* variational autoencoders (SVAE), the authors utilise a framework which provides them with rich latent representations via graphical models, and fast variational inference via conditional random field structured approximating distributions. Consequently, they demonstrate some impressive results, using their framework, and nonparametrically apply the switching linear dynamical system (Fox et al., 2011) SVAE to the problem of temporal behavioural clustering in video sequences.

Finally, Linderman et al. (2016) present Bayesian learning and inference for a class a models which they call *recurrent* SLDSs (rSLDS). They are interested in decomposing complex nonlinear dynamics, manifested through time-series observations, into segments that are explained by "simpler dynamic units". Their main contribution is to make recent advances in (bespoke) approximate inference, fast and scalable to the aforementioned model class, for large datasets. In the normal SLDS the conditional density which, governs discrete-state switching, is independent of the continuous state (see fig. 4.3). Thus, if the continuous state were to enter a particular dynamics region of state-space, that is vastly different from the preceding time-step, then the SLDS will not be able to capture this dependency. The rSLDS, on the other hand, can, by modelling these dependencies explicitly. In doing so they demonstrate their methods on a number of complex problems with difficult switching dynamics, such as the Lorenz attractor.

Alas we have reached the end of relevant prior work, and now move onto our own contributions, or better put: "once more unto the breach" (Shakespeare, 1598).

4.2 Hierarchical mixture models

In order to infer state cardinality from observations and to flexibly model the distribution of continuous data, we adopt Bayesian nonparametrics. It requires the specification of a prior model for continuous distributions. A helpful and general approach for defining such a prior model was first suggested by Lo et al. (1984) in terms of an infinite dimensional mixture model:

$$\begin{aligned} P &\sim \mathcal{P} \\ X_i \mid P &\overset{i.i.d.}{\sim} P && i = 1, 2, \dots \\ Y_i \mid X_i &\overset{ind.}{\sim} F(\cdot \mid X_i) && i = 1, 2, \dots \end{aligned} \tag{4.1}$$

where P is a discrete random probability measure (RPM) with distribution $\mathcal{P}$, $Y_{1:n}$ are a collection of continuous and possibly multivariate observations and $X_{1:n}$ are the corresponding collection of latent random variables from an exchangeable sequence directed by P. $F(\cdot \mid X_i)$ is some continuous distribution parametrised by X_i. The nonparametric hierarchical model (4.1) defines a mixture model (MM) with a potentially countably infinite number of components. Because the RPM in equation (4.1) is discrete, this means that the pair of consecutive values of X take on the same value with a strictly positive probability. This value is a mixture component. By setting the RPM to the Dirichlet process (DP) (Ferguson, 1973) we obtain the familiar DPMM. But note that the Pitman-Yor process or any other discrete RPM are valid alternatives to the DP. The Dirichlet process, denoted by $\mathcal{DP}(\gamma, H)$, is a stochastic process over countably infinite random measures on parameter space Θ. It is uniquely defined by a base measure H on Θ and a concentration parameter γ (Teh et al., 2006).

The DP is typically used as a prior on the mixture components θ of a mixture model of unknown complexity resulting in the aforementioned DPMM.

$$\begin{aligned} G \mid \alpha, H &\sim \mathcal{DP}(\gamma, H), \\ \theta_i \mid G &\sim G, \\ y_i \mid \theta_i &\sim F(\theta_i). \end{aligned} \tag{4.2}$$

There are, however, many scenarios in which *groups* of data are thought to be produced by related, yet unique, generative processes. Indeed, a recurring problem in many areas of information technology is that of segmenting a signal into a set of time intervals that have a useful interpretation in some underlying domain. In such scenarios, we can take a hierarchical Bayesian approach (where we are merely adding another layer to the model described in eq. (4.2)).

$$G_0 \mid \gamma, H \sim \mathcal{DP}(\gamma, H), \tag{4.3}$$

$$G_j \mid \alpha, G_0 \sim \mathcal{DP}(\alpha, G_0) \qquad \text{for } j \in \mathcal{J}, \tag{4.4}$$

$$\theta_{ji} \mid G_j \sim G_j \qquad \text{for } i = 1, \ldots, N_j, \tag{4.5}$$

$$y_{ji} \mid \theta_{ji} \sim F(\theta_{ji}) \qquad \text{for } i = 1, \ldots, N_j. \tag{4.6}$$

We posit that observations can be subdivided into a countable collection of groups. Groups of observations are modelled by considering a collection of DPs $\{G_j : j \in \mathcal{J}\}$, defined on a common space Θ, where $\mathcal{J}$ indexes the groups. By placing a global DP prior $\mathcal{DP}(\gamma, H)$ on the base distribution G_0, from whence we draw group specific distributions $G_j \sim \mathcal{DP}(\alpha, G_0)$, we receive the hierarchical DP (HDP). The HDP induces sharing of atoms among the random measures G_j since each inherits its set of atoms from the same G_0 (Teh et al., 2006). This idea can be used to develop HMMs with unknown, potentially infinite, state spaces (Beal et al., 2001).

4.3 Infinite hidden Markov models

Formally, a hidden Markov model is a doubly-stochastic Markov chain in which a state sequence $\{\theta_1, \ldots, \theta_T\}$ is drawn, according to a Markov chain, on a discrete state space Θ with transition kernels $\{G_\theta : \theta \in \Theta\}$ (Teh & Jordan, 2010). Corresponding observations $\{y_1, \ldots, y_T\}$, conditional on the state sequence, are drawn from a fixed emission distribution $y_t \mid \theta_t \sim F_{\theta_t}\ \forall t \in \{1, \ldots, T\}$. A remark on the notation used here; when we say F_{θ_t} we are using it as short-hand for $F(\cdot \mid \theta_t)$. Continuing, the initial state distribution is given by θ_0. A graphical model is depicted in fig. 4.4.

Up until now we have used x to denote the latent state, but from now on we will use θ and it will soon be clear why. A HMM on K states or in which $|\Theta| = K$, defines a joint distribution $\mathbb{P}(\theta_{1:T}, y_{1:T})$ over a latent state sequence $\theta_{1:T}$ and an observed or measured

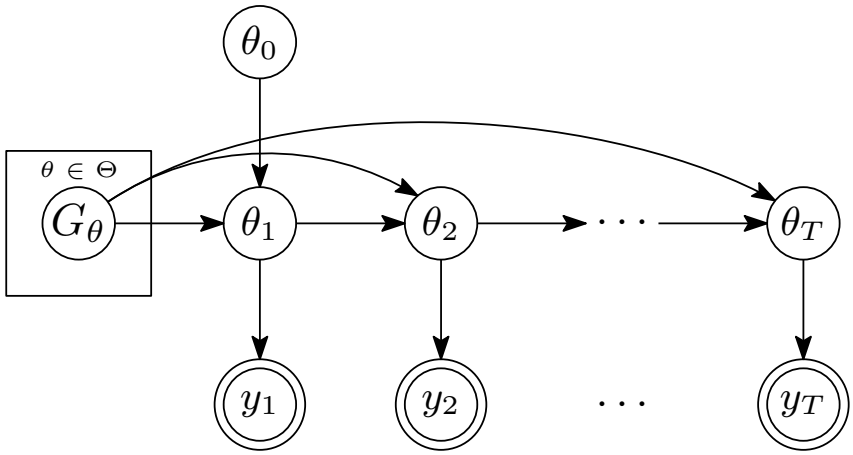


Figure 4.4: A doubly-stochastic first-order Markov chain, wherein the latent states are discrete, i.e. a hidden Markov model.

sequence $y_{1:T}$. Following Bishop (2006, §13.2), we say that the HMM is parametrised by a transition matrix π, which takes as row-entries one probability mass functions per state. Then we let $\pi = \{\pi_k\}_{k=1}^K$ denote the full *Markov* transition matrix and $\pi_k \in \mathbb{R}_+^{1\times K}$ its k^{th} row (Johnson et al., 2014, §2.4) s.t. $\pi \in \mathbb{R}_+^{K\times K}$. The emission parameters are given by $\theta = \{\theta_k\}_{k=1}^K$, where we also use the emission parameters to index the state, which those parameters belong to, thus making clear the reasoning for choosing θ as our latent variable: as well as being the evolution of the latent state sequence $\theta_{1:T}$ it is, by extension, also the evolution of emission parameters and hence the emissions governing the phenomena under investigation. To be fully explicit, the transition function $\mathbb{P}(\cdot \mid \theta_k)$ consists of a vector of K numbers corresponding to the K possible states, where $\sum_k \pi_k = 1$.

We are now in a position to consider the difference between a parametric transition matrix and its nonparametric analogue

$$\pi = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \dots & \theta_{1K} \\ \theta_{21} & \theta_{22} & \theta_{23} & \dots & \theta_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{K1} & \theta_{K2} & \theta_{K3} & \dots & \theta_{KK} \end{bmatrix} \qquad \text{Parametric transition matrix} \tag{4.7}$$

where e.g. the second state is governed by $\pi_2 = [\theta_{21} \quad \theta_{22} \quad \theta_{23} \quad \dots \quad \theta_{2K}]$. Further

$$\pi = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \dots \\ \theta_{21} & \theta_{22} & \theta_{23} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad \text{Nonparametric transition array} \tag{4.8}$$

specifies a countably infinite transition matrix $\pi = \{\pi_k\}_{k=1}^\infty$. This construction is possible thanks to the hierarchical Dirichlet process (Teh et al., 2006).

By employing the HDP in an HMM setting, a prior distribution is defined on transition kernels π_k or G_θ as we will call them from now on, yielding the HDP-HMM (Teh et al., 2006); an HMM with a countably infinite state space. To properly qualify this nonparametric Bayesian approach to HMMs, consider that each G_θ is a DP draw, and is interpreted as the transition distribution over $\theta_t \mid \theta_{t-1}$. All transition distributions are linked by the same discrete measure G_0. Hence, in expectation $\mathbb{E}[G_\theta] = G_0, \ \forall\theta \in \Theta$. Thus, transition distributions *tend* to have their mass concentrated around a common set of states (recall the type of generative process that the DP induces: *rich-get-richer* — hence only a few people can be rich, consequently all wealth flows to these people, much in the same way that all probability mass is located around a few common states), providing the desired bias towards re-entering and re-using a consistent set of states (Johnson & Willsky, 2013). This property means that separate generative processes can be linked by a common measure. In so doing it allows for a higher generative process to be responsible for the *set of common states* shared across all generative processes. This key idea and formalism forms the foundation of the analysis that follows in this section. We provide a brief review of it, and related models, which will lead into our own contributions.

4.3.1 HDP-HMM

The notion of the HDP-HMM is somewhat similar to the paradigm of going from a finite mixture model (MM) to a DPMM, as already described and demonstrated. Teh et al. (2006) made the connection with HMMs by noting that they do not involve one MM, but rather several, indeed "one for each value of the current state". Because the current state θ_t at time t indexes a row of π, the probabilities in that row can then be thought of as the mixing proportions with the emission distribution F_{θ_t}, playing the role of mixture components. It is then natural to consider a scenario in which the finite MM is replaced with a DPMM. A problem arises if these DPMMs are not tied, connected in some way, so that they all operate on the same state-space. If they are disjoint, that means that one state-indexed transition function (row) has a set of states which are inaccessible or disconnected, for some other value of the current state. The elegant solution to this problem was to replace the set of conditional and finite MMs underpinning the vanilla HMM, with a HDP, thus yielding the HDP-HMM. To be perfectly clear about its properties,

the principal one being that it is able to grow the number of partitions on Θ, as more data becomes available.

The generative model is given by

$$G_0 \mid \gamma, H_\Theta \sim \mathcal{DP}(\gamma, H_\Theta) \tag{4.9}$$

$$G_\theta \mid \alpha, G_0 \sim \mathcal{DP}(\alpha, G_0) \qquad \text{for } \theta \in \Theta, \tag{4.10}$$

$$\theta_t \mid \theta_{t-1}, G_{\theta_{t-1}} \sim G_{\theta_{t-1}} \qquad \text{for } t = 1, \ldots, T \tag{4.11}$$

$$y_t \mid \theta_t \sim F_{\theta_t}. \tag{4.12}$$

The graphical model is shown in fig. 4.5. We should also note early on that we assume like Teh et al. (2006), that the base distribution H on Θ is conjugate to the emission distribution F. Non-conjugate mixtures are outside the scope of this thesis. Further note that in our particle-based inference the base distribution and the state-distribution are re-sampled at the same rate as all the other model components.

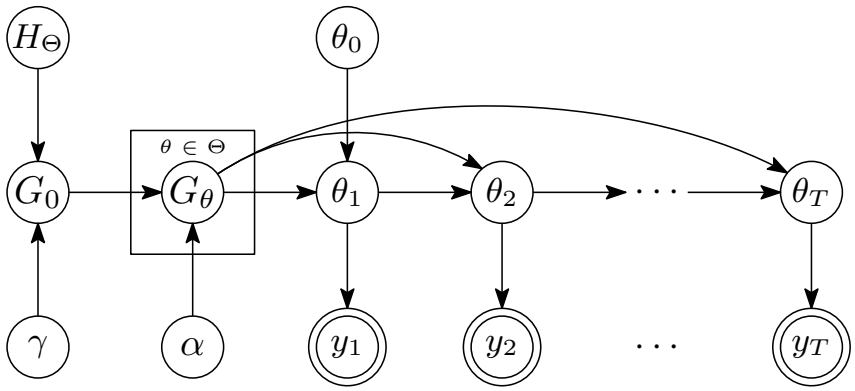


Figure 4.5: The hierarchical Dirichlet process hidden Markov model (Teh et al., 2006). The parameters for the transition, emission and initial state distributions are unobserved random variables depicted by single-barrelled nodes. The measure G_θ is conditioned on α, which represents the hyperparameter for the prior distributions on the state-transition parameters. The double-barrelled nodes indicate observations on which we condition to form the posterior distribution over the single-barrelled latent variables.

Another interesting way to think about the HDP, is as a prior over infinite transition matrices (Johnson & Willsky, 2013). Each draw from a DP is a probability mass function (PMF), and as demonstrated, a discrete distribution, and we can interpret each of G_θ as a transition distribution from state θ, the collection of which form π.

4.3.2 Sticky HDP-HMM

The elegance of the HDP-HMM notwithstanding, it is not without its flaws: specifically there are a number of issues, which disarm the model's utility for time-series evolutions that behave in a certain way. Perhaps, unsurprisingly, these issues relate back to the general problems with the HMM paradigm, described in item **I** and item **II**. Fox et al. (2008) were the first to emphasise the limitations:

- ➢ The HDP-HMM inadequately models the temporal persistence of states, dealt with in item **I**, and in so doing selects models with unrealistically fast state-switching dynamics.
- ➢ It also has a tendency to create redundant states and rapidly switch between them, in essence exacerbating the problem mentioned above.

It should be noted however, that these problems only arise for *some* observations sets. Specifically those that display a non-geometric state-duration. At any rate, when the HDP-HMM is applied to this subset of data, it fails. However, Teh & Jordan (2010) noted that this may not be problematic for applications in which the states are "nuisance variables" and the item of interest is the overall predictive likelihood. When the states have a meaning, in segmentation applications for example, a solution to the limitations of the HDP-HMM was put forward by Fox et al. (2008) who proposed a 'sticky' extension which they say "allows for more robust learning of smoothly varying dynamics".

Alas, the rate at which re-entering and re-using states unfolds in the HDP-HMM is typically too fast for many real world problems. One such real-world problem, which we would like to model, is for example stock-markets which experience vastly different duration dynamics, depending market forces. On slow days some shares will not move at all, but on others they will switch vigorously in and out of what we call different 'states'. The model construction furthermore encourages the creation of redundant states and rapid switching amongst these, too. Enter the sticky HDP-HMM which augments the HDP-HMM with an extra parameter $\kappa > 0$ that biases the process towards self-transitions and thus provides a method to encourage longer state durations. Hence transitions kernels in the HDP-HMM, equation (4.9) above, are instead sampled as follows:

$$G_\theta \mid \alpha, G_0, \kappa, \theta \sim \mathcal{DP}\left(\alpha + \kappa, \frac{\alpha G_0 + \kappa \delta_\theta}{\alpha + \kappa}\right) \tag{4.13}$$

where δ_θ is a point mass at θ. The full generative model of the sticky HDP-HMM is given by

$$G_0 \mid \gamma, H \sim \mathcal{DP}(\gamma, H) \tag{4.14}$$

$$G_\theta \mid \alpha, G_0, \kappa, \theta \sim \mathcal{DP}\left(\alpha + \kappa, \frac{\alpha G_0 + \kappa \delta_\theta}{\alpha + \kappa}\right) \qquad \text{for } \theta \in \Theta \tag{4.15}$$

$$\theta_t \mid \theta_{t-1}, G_{\theta_{t-1}} \sim G_{\theta_{t-1}} \qquad \text{for } t = 1, \ldots, T \tag{4.16}$$

$$y_t \mid \theta_t \sim F_{\theta_t}. \tag{4.17}$$

The graphical model is displayed in fig. 4.6:

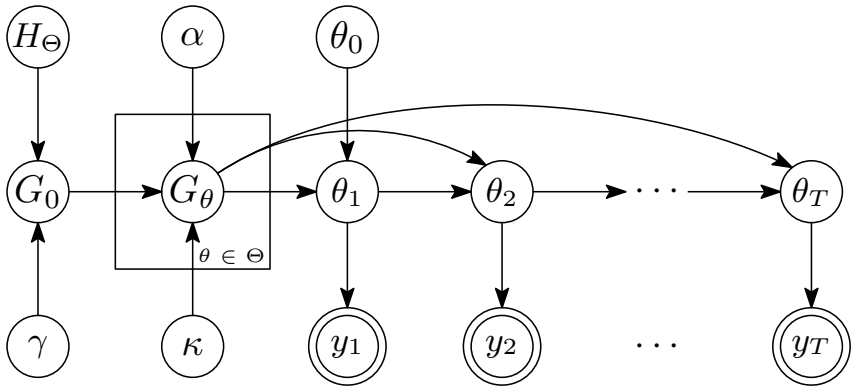


Figure 4.6: The sticky hierarchical Dirichlet process hidden Markov model (Fox et al., 2008). Th parameter κ here induces more state persistence by biasing the transition parameters.

Informally, adding κ increases the expected probability of self-transition by an amount proportional to κ. Formally, over a finite partition $\{\theta_1, \ldots, \theta_K\}$ over the positive integers $\mathbb{N}^+$, the prior in (4.13) on the measure G_θ adds an amount κ only to the arbitrarily small partition that contains θ, which corresponds to a self-transition (Fox et al., 2008, §3). Note that when $\kappa = 0$, the original HDP-HMM is recovered. The sticky HDP-HMM shares the original HDP-HMM's restriction to geometric state durations, thus limiting the model's expressiveness regarding duration structure. More importantly, its global self-transition bias is shared among all states, and does not therefore allow for learning state-specific duration information (Johnson & Willsky, 2013).

4.3.3 Stateful HDP-HMM

Johnson & Willsky (2013) noted the potential utility in specifying a sticky HDP-HMM-type model, but with state-specific parameters. Indeed Huggins & Wood (2014) found the same

thing (last part of §4.2), but also point out that "it is not straightforward to [specify a model with state-specific parameters κ_θ]". We propose such a model herein.

We propose that by allowing for group-specific self-transition biases κ_θ, greater hetereogenicity can be achieved in the dwell-time distribution of the inferred states. We extend this idea further by allowing for group-specific concentration parameters α_θ. Hence we augment the original HDP-HMM by sampling transition functions, in eq. (4.13), like so

$$G_\theta \mid \alpha_\theta, G_0, \kappa_\theta, \theta \sim \mathcal{DP}\left(\alpha_\theta + \kappa_\theta, \frac{\alpha_\theta G_0 + \kappa_\theta \delta_\theta}{\alpha_\theta + \kappa_\theta}\right). \tag{4.18}$$

We refer to this extension as the *stateful* HDP-HMM – in reference to its pronounced usage of memoized groups and their statistics, with parameters indexed by θ. In adopting this approach we imbue the original sticky HDP-HMM with more flexibility w.r.t. modelling the state duration more accurately. We allow for state-specific duration information to be encoded via κ_θ and also admit α_θ to determine the extent of the repetition of the values of G_θ. The generative model of the stateful HDP-HMM is given by

$$G_0 \mid \gamma, H \sim \mathcal{DP}(\gamma, H) \tag{4.19}$$

$$G_\theta \mid \alpha_\theta, G_0, \kappa_\theta, \theta \sim \mathcal{DP}\left(\alpha_\theta + \kappa_\theta, \frac{\alpha_\theta G_0 + \kappa_\theta \delta_\theta}{\alpha_\theta + \kappa_\theta}\right) \qquad \text{for } \theta \in \Theta \tag{4.20}$$

$$\theta_t \mid \theta_{t-1}, G_{\theta_{t-1}} \sim G_{\theta_{t-1}} \qquad \text{for } t = 1, \ldots, T \tag{4.21}$$

$$y_t \mid \theta_t \sim F_{\theta_t}. \tag{4.22}$$

The graphical model is provided in fig. 4.7:

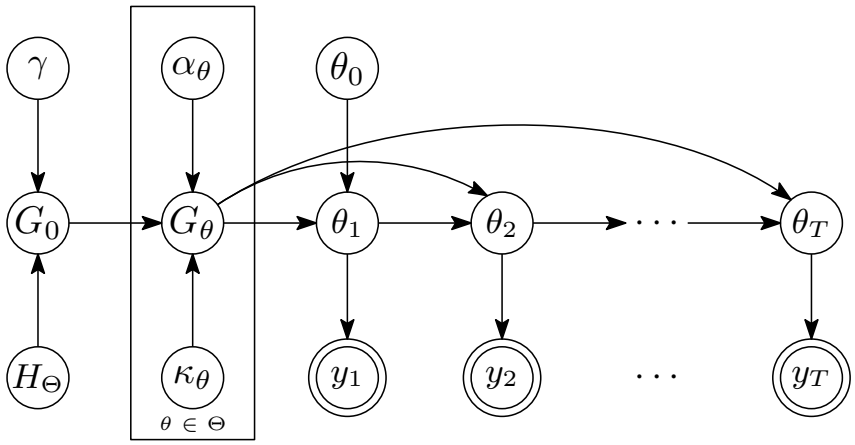


Figure 4.7: The stateful hierarchical Dirichlet process hidden Markov model (Dhir et al., 2016b). Here the plate notation around α_θ and κ_θ ensures that their instantiations are indexed by the state θ.

Whilst the stateful HDP-HMM is an extension of the sticky HDP-HMM, which largely comes from making in more flexible and complex (more parameters), is it better? The stateful HDP-HMM is more parametrised, it still relies on the same nonparametric construction as the sticky HDP-HMM, the improvement comes when we attach explicitly, state statistics to the states, rather than keeping them global as the sticky HDP-HMM does. Hence, if we are modelling a process which has heterogeneous duration statistics, then stateful HDP-HMM should handle it better than sticky HDP-HMM.

That being said, the stateful HDP-HMM and the sticky HDP-HMM still do not deal with item **I**, namely: *state duration distributions are necessarily restricted to be of geometric shape*, which is to say when measurements display a large degree of non-geometric duration in one specific state, these models will switch out of them prematurely. Hence, whilst both the sticky HDP-HMM and stateful HDP-HMM allow for a degree of control over duration statistics, the state duration distributions remain geometric, as the models *build upon* the basic HMM, but do not inherently change its fundamental properties. More desirable would be a model which admits any duration distribution, for global or local usage. We consider models of this flavour in the next section.

4.3.4 Infinite duration HMM

In the classic HMM, the duration of a given state has a geometric (in particular, monotonically decreasing) distribution, because of the Markov property. Geometric duration distributions have been found to be deficient not just in behavioural modelling but also in e.g. speech synthesis (Bilmes, 2006). Thus, we must seek models where the geometric constraint is relaxed, so that models become *semi*-Markovian. One such model is the explicit duration HMMs (EDHMM), see fig. 4.8, developed by Dewar et al. (2012) to make the duration distribution explicit and allow it to have a more general form. Put simply, in an EDHMM, during a (Markov) state transition, a duration is drawn explicitly from a specified duration distribution depending on the new state. After that, the probability of self-transition is one until the duration has elapsed. We prefer to consider the EDHMM over the hidden semi-Markov model (HSMM), which achieves a similar effect through different, less explicit means (Johnson & Willsky, 2010). That being said, state-of-the art work for nonparametric semi-Markovian HMMs comes in the form of the HDP-HSMM by Johnson & Willsky (2010).

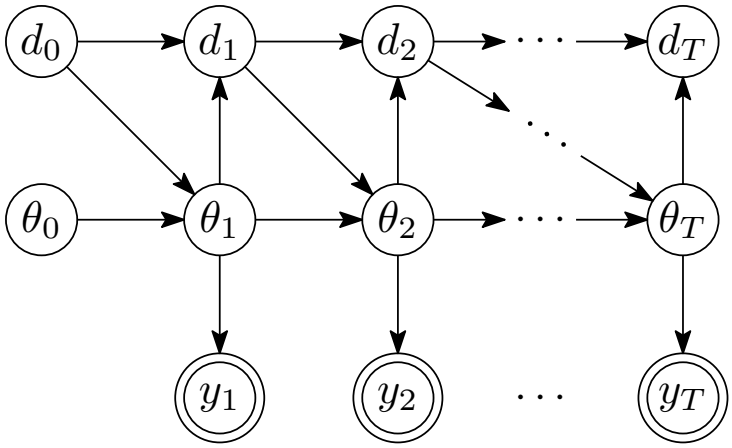


Figure 4.8: The explicit duration hidden Markov model. Here the d parameters control how long the process persists in a particular state θ. It does this by counting down the value of d at each time-step t.

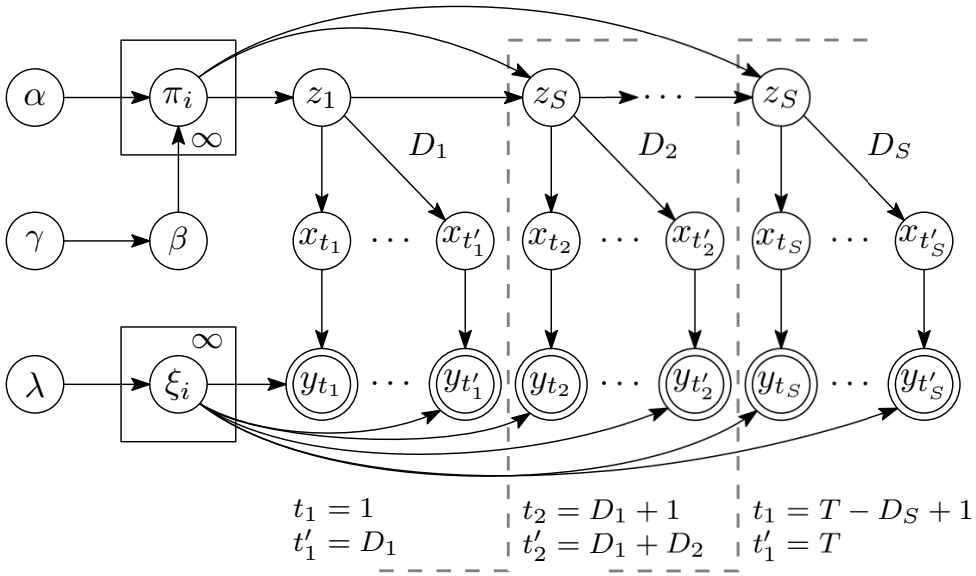


Figure 4.9: The hierarchical Dirichlet process hidden semi-Markov model (Johnson & Willsky, 2010). Note that appearance of global states S, which dictate the evolution of the process.

The HDP-HSMM graphical model structure is shown in fig. 4.9. Two models most resemble the construction, which we propose: the HDP-HSMM of Johnson & Willsky (2010) and the infinite explicit duration HMM (IED-HMM) of Huggins & Wood (2014) – see fig. 4.10 for its graphical model. We briefly describe these models and compare them with our own contributions.

The generative model of the HDP-HSMM (using a stick-breaking construction) is given

by

$$\beta \mid \gamma \sim \mathrm{GEM}(\gamma)$$

$$\pi_i \mid \beta, \alpha \overset{i.i.d.}{\sim} \mathcal{DP}(\alpha, \beta) \qquad (\xi_i, \omega_i) \overset{i.i.d.}{\sim} H \times G \tag{4.23}$$

$$z_s \mid z_{s-1} \sim \bar{\pi}_{z_{s-1}}$$

$$d_s \mid \omega_{z_s} \sim D(\omega_{z_s}) \tag{4.24}$$

$$y_{t_s^1:t_s^2} \overset{i.i.d.}{\sim} F(\xi_{z_s}) \qquad t_s^1 = \sum_{\bar{s}<s} d_{\bar{s}} \tag{4.25}$$

where $t_s^2 = t_s^1 + d_s - 1$. Johnson & Willsky (2010) define $\bar{\pi} \triangleq \frac{\pi_{ij}}{1-\pi_{ii}}(1-\delta_{ij})$ to eliminate self-transitions in what they term their super-state sequence (z_s).

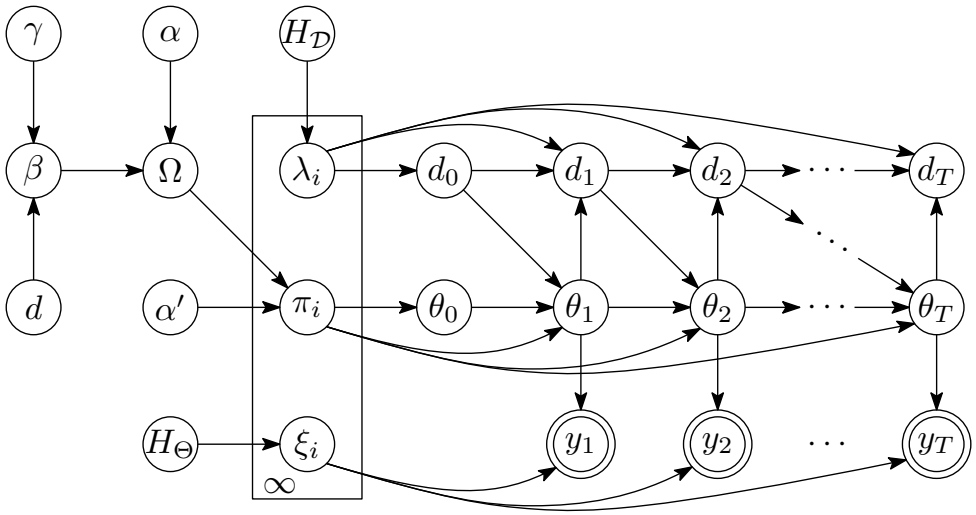


Figure 4.10: The infinite explicit duration hidden Markov model (Huggins & Wood, 2014). This model yields a nearly equivalent model to the HDP-HSMM in fig. 4.9 but its construction gives rise to different inference algorithms.

The IED-HMM arises from the infinite structured hidden semi-Markov model (ISHSMM) Bayesian nonparametric framework (Huggins & Wood, 2014, §5), which the authors note is "very closely related to the sticky HDP-HMM and the HDP-HSMM". This framework, in which the IED-HMM is specified, directly parametrises state duration distributions, allowing for more heterogeneity and specificity, in state dwell durations when compared to the IED-HMM (Huggins & Wood, 2014).

The main difference, relative to the HDP-HSMM, is that the IED-HMM framework gives rise to different inference algorithms. To quote Huggins & Wood (2014, §5): "The IED-HMM yields a nearly equivalent model to the HDP-HSMM (...) [while] the ISHSMM construction gives rise to different inference algorithms". In particular, duration distributions are still

treated parametrically, and the additional inference algorithms that arise are: a version of blocked Gibbs sampling (Huggins & Wood, 2014, §6) which also employs beam sampling. Otherwise the IED-HMM is the same as the HDP-HSMM *except* in the way it constructs dependent, infinite-dimensional transition distributions with structural zeros. Consider the models in figs. 4.9 and 4.10, the HDP-HSMM is recovered from the IED-HMM (fig. 4.10) by using the Pitman-Yor process (PYP) construction of the base distribution H_Θ. Thus, the HDP-HSMM is received by setting $d = 0$, meaning that $\beta \mid \gamma \sim \text{GEM}(\gamma)$. Then, by letting $\alpha \to \infty$, results in a zero variance about β (Huggins & Wood, 2014, §5.4), which forces $\Omega \equiv \beta$.

Consider some tangible examples of the HDP-HSMM; in Johnson & Willsky (2010, §5.2 and §5.3) the authors employed a delayed-geometric duration distribution with Uniform and Beta prior parameter distributions, to spectrogram data from audio of the Morse code alphabet. The authors note that the alphabet has precise properties which allow it to be clustered into 'tone' and 'silence', without "inspecting its temporal structure". But only by including duration information, can short and long tones be differentiated, and consequently the correct learning of the state representation. Huggins & Wood (2014, §7.1.1) replicate these same Morse-code segmentation experiments in their ISHSMM framework. They also consider the well-studied change-point dataset which contains the number of major coal mining disaster in Britain between 1851 and 1962. For this they used Poisson durations with gamma priors.

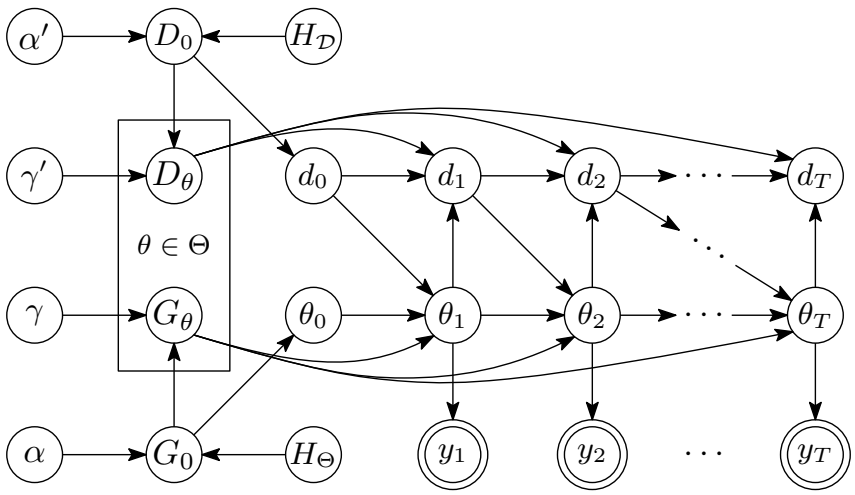


Figure 4.11: The infinite duration hidden Markov model (Dhir et al., 2017c). The IDHMM is similiar to the HDP-HSMM in fig. 4.9 except its durations are chosen from a nonparametric family, whereas the HDP-HSMM's comes from a parametric family. The IDHMM neither jointly samples state and duration, instead keeping them disjoint.

Contrast now these models to our infinite duration HMM (IDHMM), a BNP variant on the EDHMM, which we present herein (see fig. 4.11). We posit that the IDHMM can be preferable as it gives a nonparametric rather than parametric treatment of duration distributions. The HDP-HSMM can be seen as similar to the IDHMM with two key differences

1. The duration distributions are chosen from a parametric family rather than modelled nonparametrically
2. The top level DP for state transitions is implemented through a stick-breaking construction

The IDHMM keeps the state θ and duration d disjoint, instead of sampling the pair as a joint super-state like the HDP-HSMM. This can be particularly valuable when incisive domain knowledge (e.g., environmental or societal factors affecting an animal's ecology) is available regarding the nature of θ and d. The IDHMM models the relationship between state $\theta_t \in \Theta \subseteq \mathbb{N}$, duration $d_t \in \mathcal{D} \subseteq \mathbb{N}$ and observation $y_t \in \mathcal{Y} \subseteq \mathbb{R}^n$, $\forall t \in \mathcal{T} \triangleq \{1, \ldots, T\}$, whilst giving a nonparametric treatment of state cardinality and state duration. The base and group distributions, in the generative model, are drawn as

$$
\begin{aligned}
G_0 \mid \gamma, H_\Theta &\sim \mathcal{DP}(\gamma, H_\Theta) && && \\
D_0 \mid \gamma', H_\mathcal{D} &\sim \mathcal{DP}(\gamma', H_\mathcal{D}) && && (4.26)\\
G_\theta \mid \alpha, G_0 &\sim \mathcal{DP}(\alpha, G_0) && \text{for } \theta \in \Theta && \\
D_\theta \mid \alpha', D_0 &\sim \mathcal{DP}(\alpha', D_0) && \text{for } \theta \in \Theta && (4.27)
\end{aligned}
$$

where G_0 and D_0 have support Θ and $\mathcal{D}$ respectively. Noting that G_θ, being a draw from a DP, is a discrete distribution, we can define $G_\theta[\mathbb{P}(\theta) = 0]$ as the measure obtained from G_θ by setting the probability of drawing θ to 0 and renormalising. Next, a sequence of states, durations and emissions are drawn as

$$
\theta_t \mid \theta_{t-1}, d_{t-1} \sim \begin{cases} \delta_{\theta_{t-1}}, & \text{if } d_{t-1} > 1 \\ G_{\theta_{t-1}}[\mathbb{P}(\theta_{t-1}) = 0], & \text{otherwise} \end{cases} \tag{4.28}
$$

$$
d_t \mid d_{t-1}, \theta_t \sim \begin{cases} \delta_{d_{t-1}-1}, & \text{if } d_{t-1} > 1 \\ D_{\theta_{t-1}}, & \text{otherwise} \end{cases} \tag{4.29}
$$

$$
y_t \mid \theta_t \sim F(\theta_t). \tag{4.30}
$$

where δ_a is a δ-distribution with all its mass on a (Dewar et al., 2012). That is, we keep the state fixed and decrease the duration with one or, if the duration reaches zero, we sample a *different* new state, depending on the old state, and a corresponding new duration, depending on the new state. Using $G_\theta[\mathbb{P}(\theta) = 0]$ rather than G_θ allows us to exclude self-transitions after the duration has expired, to make sure that the duration distribution of θ is actually given by D_θ.

Further, consider that each $G_\theta[\mathbb{P}(\theta) = 0]$ is obtained from a DP draw and is interpreted as the transition distribution over $\theta_t \mid \theta_{t-1}$. All transition distributions are linked by the same discrete measure G_0. Hence, in expectation $\mathbb{E}[G_\theta[\mathbb{P}(\theta) = 0] \mid G_0] = G_0[\mathbb{P}(\theta) = 0],\ \forall\theta \in \Theta$. Since transition distributions tend to have their mass concentrated around a common set of states, a bias towards re-entering and re-using a common set of states is received. Similarly, our choice to extend the representational power of an HDP to durations as well means that the model reuses a common set of state durations.

For the IED-HMM (fig. 4.10), we see that durations arise as draws from the class of duration distribution represented by $H_\mathcal{D}$. Where the homologous measure in the HDP-HSMM is given by $D(\cdot)$ (see eq. (4.24)). Hence, it follows that in these two model frameworks durations always *originate from the same distribution class*; where the main duration inference task consists of drawing accurate samples from the duration prior parameter distribution G (see eq. (4.23)). The IDHMM, as explained, employs a different duration modelling strategy. By levering the statistical strength of hierarchical processes we can model more complex duration phenomena like multi-modal duration distributions.

In summary, our IDHMM differs from these two precedents primarily (ignoring construction and inference details) by modelling durations non-parametrically. By levering the statistical strength of hierarchical processes also for durations, we can model complex duration phenomena like multi-modal duration distributions which may be more difficult to treat in a parametric setting.

4.3.5 Stateful IDHMM

We can also expose the IDHMM to the stateful representation, fig. 4.12, as analogous to the work presented by Dhir et al. (2016b) and the stateful HDP-HMM. Like the stateful HDP-HMM this adds greater heterogeneity in the state and dwell-time distributions of the

inferred states. We will explain in coming sections how general purpose inference allows us to quickly build and experiment with models in this modular fashion, which, in its turn, can give us greater flexibility than using bespoke sampling algorithms such as those employed by Beal et al. (2001); Teh et al. (2006); Fox et al. (2008); Heller et al. (2009); Johnson & Willsky (2010), in settings like ours where the computational efficiency of the latter is outweighed by the programmer efficiency of the former.

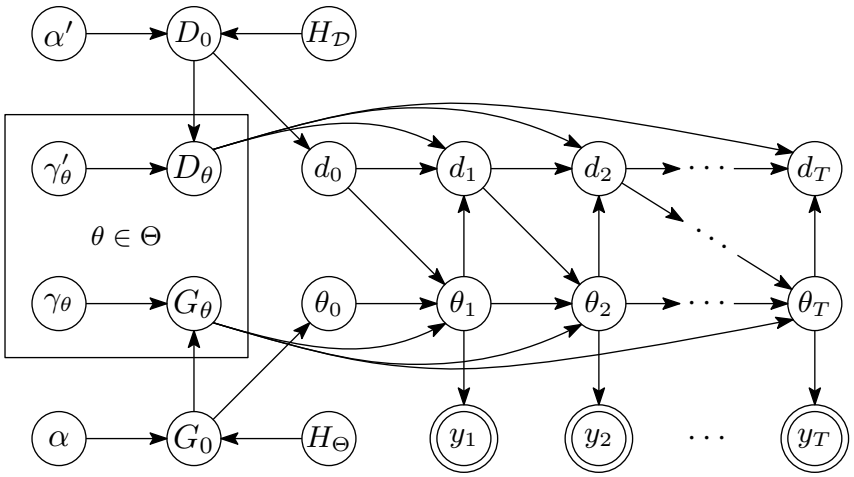


Figure 4.12: The stateful infinite duration hidden Markov model (Dhir et al., 2017c). The stateful extension from the IDHMM in fig. 4.11 is induced on the model since parameters γ'_θ and γ_θ are indexed by the state θ, as shown by the bounding plate.

4.3.6 Other transition matrix priors

Our extensions are readily applicable to the Hierarchical Pitman-Yor Process HMM (HPYP-HMM) (Blunsom & Cohn, 2011). The random probability measure in the stateful HDP-HMM in eq. (4.19) to eq. (4.22) is a Dirichlet process so the state cardinality grows logarithmically with the number of observations (Teh & Jordan, 2010, §2.2). For a large range of real-world problems this kind of logarithmic growth is not a realistic assumption, since many important statistics (for example language modelling) are known to follow power-laws (Teh & Jordan, 2010). By generalising the Dirichlet process we can induce such power-law behaviour via the Pitman-Yor process. The stateful hierarchical prior thus becomes

$$G_0 \mid \gamma, d, H \sim \mathcal{PYP}(\gamma, d, H), \tag{4.31}$$

$$G_\theta \mid \alpha_\theta, d_\theta, G_0, \theta \sim \mathcal{PYP}(\alpha_\theta, d_\theta, G_0) \qquad \text{for } \theta \in \Theta, \tag{4.32}$$

where d is the discount parameter, which takes values in the range $[0, 1)$. For $d > 0$, the model tends to produce a few large and many small clusters (for $d = 0$ we recover the DP), appropriate for e.g. language-modelling (Blunsom & Cohn, 2011) as well as many other complex phenomenon (including the sizes of craters on the moon and of solar flares (Newman, 2005)). The full stateful HPY-HMM generative model is then given as follows

$$\begin{aligned} G_0 \mid \gamma, d, H &\sim \mathcal{PYP}(\gamma, d, H) \\ G_\theta \mid \alpha_\theta, d_\theta, G_0, \theta &\sim \mathcal{PYP}(\alpha_\theta, d_\theta, G_0) && \text{for } \theta \in \Theta \\ \theta_t \mid \theta_{t-1}, G_{\theta_{t-1}} &\sim G_{\theta_{t-1}} && \text{for } t = 1, \ldots, T \\ y_t \mid \theta_t &\sim F(\theta_t). \end{aligned} \tag{4.33}$$

The extension to the IDHMM and stateful IDHMM are trivial.

4.3.7 Bespoke approximate inference

Inference in hierarchical Bayesian nonparametric models is typically achieved using bespoke, model-specific algorithms, typically using one of the various mathematical representations available for nonparametric models including; stick-breaking representations, urn models and truncations (Teh & Jordan, 2010). Teh et al. (2006) present *three* related Markov chain Monte Carlo (MCMC) sampling schemes for the hierarchical DPMM. The first is a straightforward Gibbs sampler based on the Chinese restaurant franchise; the second is based on an augmented representation involving both the Chinese restaurant franchise and the posterior for G_0. The third is a variation on the second sampling scheme with streamlined bookkeeping.

The extension to HDP-HMMs is be done with Gibbs sampling, but can also be done with slice sampling as shown by Van Gael et al. (2008), as well as by truncating the allowable state-space and then using the forward-backward algorithm. The problem with the simple Gibbs sampler is that it will converge slowly due to strong dependencies among the latent states (Teh et al., 2006). A faster algorithm updates the latent states in a block via the forward-backward algorithm for HMMs (Bishop, 2006). The traditional form of this algorithm cannot, however, be applied directly to the HDP-HMM since there is an infinite number of possible states. The solution is to truncate the state cardinality of the

model. Fox et al. (2008) achieve this using a truncation of the stick-breaking process while Van Gael et al. (2008) proposed a slice sampling approach, which adaptively limits the number of states to a finite number thus allowing forward-backwards usage. This can also be dealt with, rather elegantly, using stochastic variational inference. Zhang et al. (2016) derive a variational inference algorithm for the HDP-HMM based on the two-level stick breaking construction. They address posterior inference for the HDP-HMM over all variables by deriving batch and stochastic variational algorithms using the fully conjugate representation originally derived by Wang et al. (2011). In addition to this Fox et al. (2008) employs a blocked Gibbs sampler as well as a direct assignment Rao–Blackwellized Gibbs sampler. Both these schemes deal with the dependencies amongst the latent states. For further details on these algorithms see (Fox, 2009). We add some additional thoughts on the weak-limit sampler as well.

Both Fox et al. (2008) and Johnson & Willsky (2013) use weak-limit samplers to construct *finite* approximations to the HDP transitions with L-dimensional Dirichlet distributions. In this instance L is what we term the 'weak-limit' and we can use it to approximate a bounded Dirichlet process. Here seen using a stick-breaking construction

$$\mathrm{GEM}_L(\gamma) \triangleq \mathrm{Dirichlet}\left(\frac{\gamma}{L}, \ldots, \frac{\gamma}{L}\right). \tag{4.34}$$

Since we can think of the HDP as a prior over infinite transition matrices, the weak-limit approximation means we can do the same thing but for square matrices of maximal size $L \times L$. Johnson & Willsky (2013) explain that the weak-limit approximation represents not only π in finite form, when we also extend the approximation to all model parameters, we can perform full block sampling of the whole latent sequence. This usually results in greatly improved mixing rates. We do not include the full details here, since it is outside the scope of this thesis. But for a full implementation (and a measure of its complexity) on the sticky HDP-HMM see (Fox, 2009, algorithm 10).

Further, by factoring the joint density over latent and observed variables, we get a sense of which densities need approximating:

$$\mathbb{P}(y_{1:T}, \theta_{1:T}) = \mathbb{P}(\theta_0) \prod_{t=1}^{T} \mathbb{P}(\theta_t \mid \theta_{t-1}) \prod_{t=1}^{T} \mathbb{P}(y_t \mid \theta_t). \tag{4.35}$$

Performing inference over these types of models seeks to infer the posterior distributions over the number of states and the persistence of each state. This posterior uncertainty can

be integrated out when making predictions, effectively averaging over models of varying complexity (Fox et al., 2008). This is done typically using one of two approaches; direct assignment sampler or a weak-limit sampler. It is not unusual for articles presenting these types of models, to spend at least one or two pages (sometimes more), deriving the inference algorithms, which is an unfortunate but necessary consequence of using bespoke inference. Conversely the reward is instead reaped in inference speed.

⋆⋆⋆

Hence, unlike other approaches taken at this point with regard to inference algorithms, we presume not to spend the remainder of this chapter discussing and deriving inference algorithms. Instead, we deploy the philosophy of probabilistic programming systems (PPS), wherein we decouple model specification and inference – for a primer see §2.6.

4.4 Empirical evaluation

In this section, we demonstrate some of the features of contributed models alongside state-of-the-art. We will first consider their performance on synthetic data in §4.4.1, followed by two real-world datasets in §4.4.3 and §4.4.6, where Bayesian nonparametric probabilistic programs are used for inferring various properties of the phenomena under investigation.

4.4.1 Synthetic observations

We explore the relative performance between the five models introduced hitherto, by simulating observations $y_t \in \mathbb{R}^{D=2}$, from a very noisy three-state multivariate HSMM (see fig. 4.13) with Gaussian emissions – see fig. 4.14a. Consequently, from the inference perspective, the emission distribution has unknown mean and covariance parameters. The conjugate prior to the multivariate Gaussian is the normal-inverse-Wishart distribution – see (Murphy, 2012). It is denoted by $\mathcal{NIW}(\mu_0, \lambda_0, \Psi, \nu)$. Through conjugacy we seek the posterior distribution of $\{\mu_\theta, \Sigma_\theta\}\ \forall\theta \in \Theta$, where we index group-specific (i.e., behaviour-specific) parameter samples by θ, given a set of observations $y_t \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$. For convenience of notation let $\mathbf{Y} = [y_1, \ldots, y_T]^\intercal$. For brevity, results are only shown for sequential Monte Carlo (SMC), chosen for its superior performance for this model class. In addition, particle Gibbs (iterated conditional SMC) and lightweight Metropolis-Hastings were

all tested as part of our experiments. For background on these methods see §2.6. Model and emissions parameter priors are shown in table 4.1. We place non-informative hyper-priors on model parameters and then condition the models on the observations and sample state trajectories; i.e. we a state-membership to each time-step $\theta \to t,\ \forall\theta \in \Theta \wedge \forall t \in \{1, \ldots, T\}$. We use synthetic observations of size $T = 1000$ and use 100 samples for each particle count (see fig. 4.14).

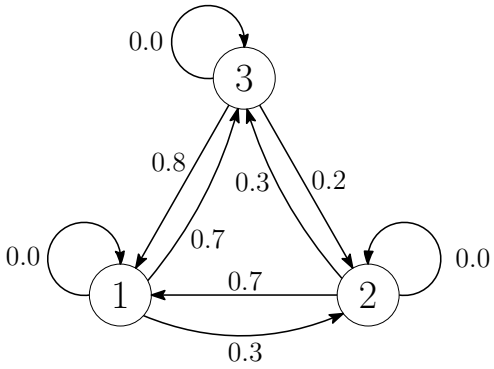


Figure 4.13: State transition diagram for a three-state, multivariate emission HSMM. All states have a zero probability of self-transition.

Table 4.1: Common model and emission priors used for synthetic data experiments.

Model parameter	**Prior**
$\alpha,\ \gamma,\ \alpha',\ \gamma'$	$\Gamma(1, 1)$
κ	$\Gamma(2, 1)$
H_θ	$\mathcal{N}(0, 1)$
$H_\mathcal{D}$	$\mathcal{U}(25, 100)$, $\mathcal{N}(\mu, \Sigma)^\dagger$, $\text{Pois}(\lambda)^\dagger$
μ_0	$\overline{\mathbf{Y}}$ (empirical average)
λ_0	$D + 2$
Ψ	$0.75 \times \text{Cov}(\mathbf{Y})$
ν	0.1

† Used as part of parametric mixture duration distribution

Performance is measured with the Hamming distance, a common clustering metric (MacKay, 2003). A formal description is given in definition 4.4.1.

Definition 4.4.1. (Hamming distance). The Hamming distance function

$$\mathcal{D}_{\mathsf{HAM}}(n, d) \triangleq \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\} \tag{4.36}$$

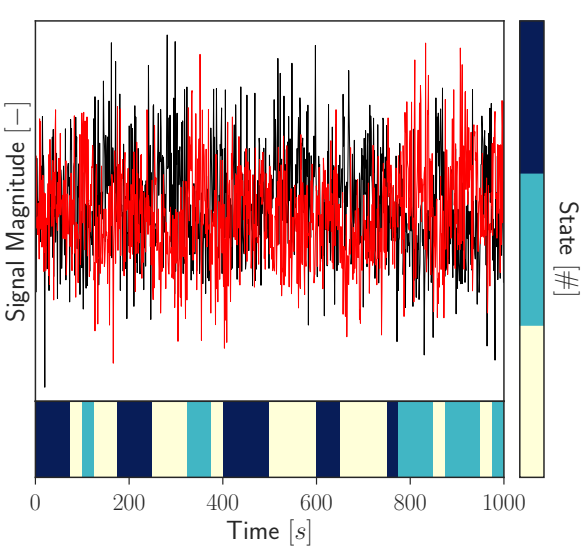
returns 1 on all pairs of inputs $x, y \in \{0, 1\}^n$ that differ in at most d coordinates and returns 0 otherwise (Blais et al., 2014).

This distance metric works by mapping the randomly chosen indices of the estimated state sequence, to the set of indices that maximises the overlap with the true sequence (Fox et al., 2008). Thus the Hamming distance between two vectors is the number of bits we must change to change one into the other. Picking a metric though, is always going to be difficult, especially was we are trying to estimate the quality of a sample drawn from a stochastic process. The Hamming distance is one of many metrics for measuring the 'edit' distance between two sequences. Another example is the *Levenstein* distance which assigns a unit cost to all edit operations (Cohen et al., 2003). We will later on also employ the normalised mutual information (MacKay, 2003) (i.e. the normalised relative entropy between two sequences). However, there are many members of the family of distance metrics (Navarro, 2001). They have in common that they are all types of information theoretic measures, that give some insight into how similar or dissimilar two sequences actually are. In sum; the distance between two sequences, is the number of positions at which the corresponding symbols are different. Hence, the lower, the better.

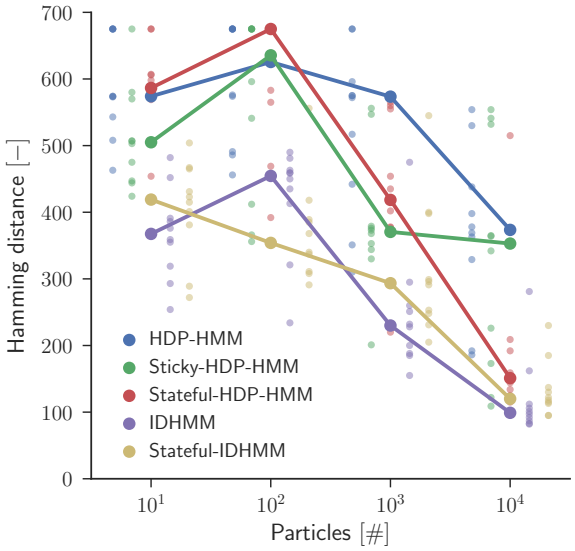
Back to the experiments, the results in fig. 4.14 demonstrate the advantage that our proposed model structure can have for modelling phenomena with non-geometric duration distributions. First, as demonstrated by Dhir et al. (2016b), stateful models yield a clear benefit compared with their contemporaries. This is because we increase the heterogeneity of the dwell-time distribution of the inferred states of the model, by making the model statistics group specific. This is shown in fig. 4.14b where the stateful HDP-HMM outperforms the HDP-HMM and sticky HDP-HMM for some particle counts. The benefits of a stateful representation are less clearcut for the IDHMM, but both IDHMMs outperform other state-of-the-art BNP SSMs.

By extending the HDP-HMM and stateful HDP-HMM, by drawing upon explicit-duration semi-Markovianity (Johnson & Willsky, 2010), as was done in the HDP-HSMM, one allows for the *parametric* construction of highly interpretable models which admit prior information on the state durations (Johnson & Willsky, 2010). We make that model even more flexible by giving a nonparametric treatment to the state cardinality *and* durations. By levering the statistical strength of HDPs for durations, we can model complex duration phenomena, which may be more difficult to treat in a parametric setting. We demonstrate in fig. 4.14e and fig. 4.14f the flexibility that our model structure provides. By employing the duration distributions shown in fig. 4.14d, we gain approximately the same utility in

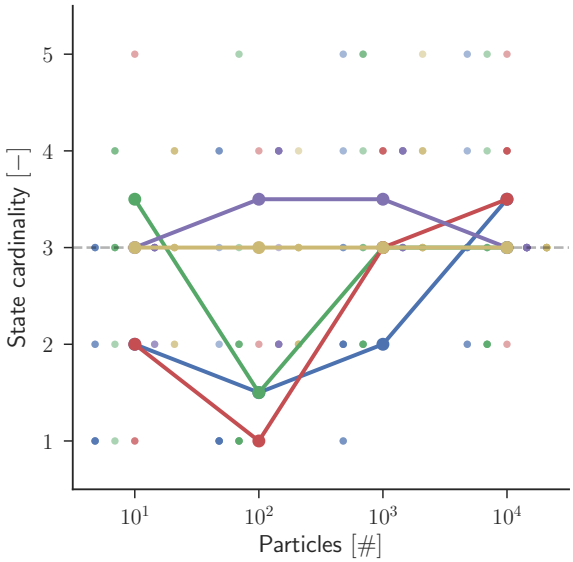
terms of the Hamming distance, when using a uniform-discrete duration prior. However, because we are targeting a duration process of the form shown by the red bars in fig. 4.14d, it is more appropriate to focus our duration prior density on those regions. Hence, we see that by employing two poorly-specified mixture duration distributions (fig. 4.14e: $\mathcal{N}(35, 15) + \mathcal{N}(85, 15)$ and fig. 4.14f: $\text{Pois}(35) + \text{Pois}(85)$ – with equal mixing proportions), the posterior state cardinality is better specified.



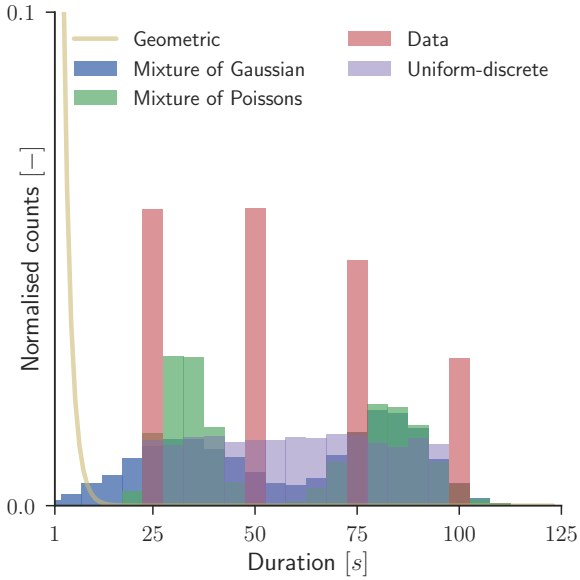
(a) Raw data with ground truth (bottom panel).



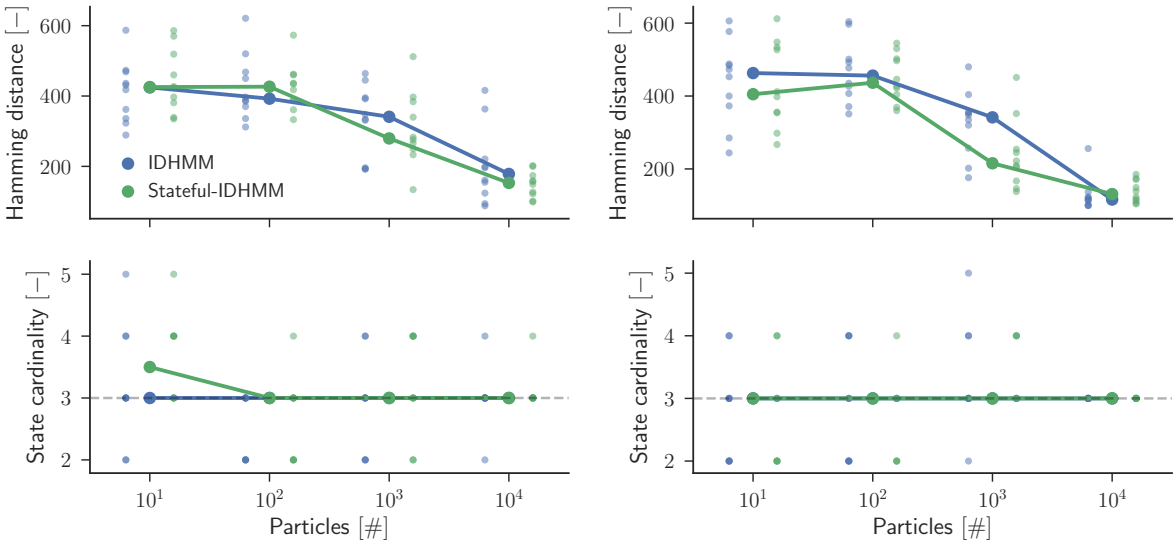
(b) Hamming distance.



(c) State cardinality.



(d) Duration distribution.



(e) Gaussian mixture duration

(f) Poisson mixture duration

Figure 4.14: Results from experiments on multivariate synthetic Gaussian observations with SMC inference. A three-state HSMM with non-geometric duration distributions was sampled to create the observation set in fig. 4.14a. All models under discussion are compared and contrasted in fig. 4.14b and fig. 4.14c. Different durations models used as priors, as well as the observations' duration model, are shown in fig. 4.14d. Our two contributed models are compared, under different durations priors, in fig. 4.14e and fig. 4.14f. Shaded small bullets, correspond to the specific experimental results (a total of ten trials were conducted for each particle count, for each model). The solid large bullets correspond to the median error across all trials in each particle count, for that model. Adapted from (Dhir et al., 2017c).

In this set of experiments we have employed the Hamming distance, as a proxy metric for the quality of the posterior distribution of our models. We, like ours peers; Fox et al. (2008) and Johnson & Willsky (2013), use the Hamming distance because the label-switching problem of Bayesian nonparametrics and MCMC sampling (also known as non-identifiability problem) makes it impossible to receive a posterior distribution over state-sequence estimates. For a formal definition see definition 4.4.2.

Definition 4.4.2. (Label switching problem). The non-identifiability of the permutation of clusters or more generally latent variables, makes interpretation of results computed with MCMC sampling difficult. The non-identifiability poses no problems if the quantities of interest are invariant under permutations of the labels (Puolamäki & Kaski, 2009).

Definition 4.4.2 fits precisely the scenario which we are faced with: samples from the HDP are exchangeable, meaning that any random permutation of the constituent samples in the draw, will yield an equivalent probability. Consequently we cannot distinguish one

sequence from another, because the state-labels have switched (in the Chinese-restaurant analogy, this corresponds to the indices of the tables switching). Hence the quality of the posterior distribution can only be ascertained by employing distance metrics such as the Hamming distance. This means that using such edit distances (Navarro, 2001), we measure the similarity of maximum model negative log-likelihood samples, to the ground truth. But any two separate model log-likelihoods (from different particles) will be disjoint, and hence non-identifiable, hence we cannot estimate the quality of the posterior, like one can for other MCMC methods such as Gibbs sampling, by looking at histograms over all samples. We can however, if we look at histograms of samples, originating from one particle.

Finally, in some instances it is difficult to appreciate the domain in which the model parameters live, hence placing an appropriate hyperprior on said parameters becomes difficult. One way of dealing with this is to employ Bayesian optimisation, introduced in §2.6.5.

4.4.2 Synthetic observations with Bayesian optimisation

In this section we evaluate Bayesian optimisation (BO) using the HDP-HMM, sticky HDP-HMM and the stateful HDP-HMM. We explore the relative performance between the models, by simulating data from a very noisy three-state HMM with Gaussian emissions – see fig. 4.15.

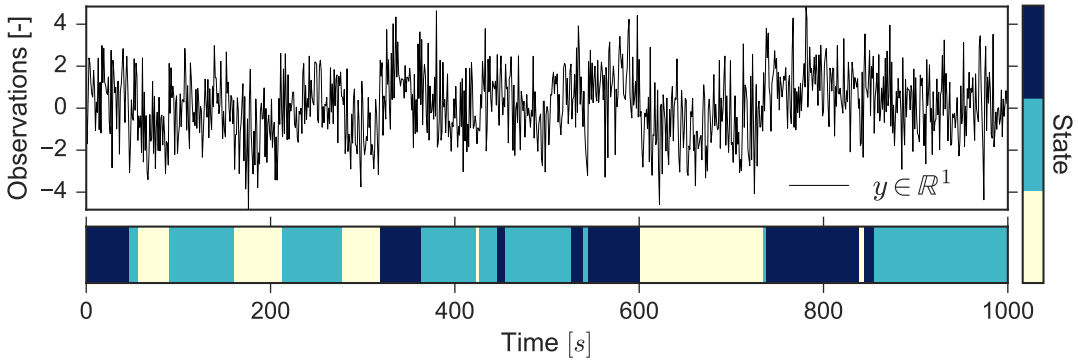


Figure 4.15: Observation sequence (top panel) and true state sequence (bottom panel) for a three-state HMM with Gaussian emissions with state persistence.

Performance is measured with the maximum likelihood estimate of the normalised mutual information (NMI) — a clustering metric (MacKay, 2003). Formally the mutual information

(MacKay, 2003) of two discrete random variables X and Y is defined as

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} \mathbb{P}(x,y)\log\left(\frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\ \mathbb{P}(y)}\right). \tag{4.37}$$

NMI is an normalisation of the mutual information score to scale the results between 0 (no mutual information) and 1 (perfect correlation), and is given by

$$\mathrm{NMI}(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}, \tag{4.38}$$

where $H(X)$ and $H(Y)$ are the marginal entropies of the random variables (Cahill, 2010). Marginal entropy is also known as the Shannon entropy (Shannon, 2001) and is defined as

$$H(X) = -\sum_{x\in X} f(x)\log_2 f(x) \tag{4.39}$$

and is measured in bits, and intuitively measures the amount of information in a random variable. We receive the Shannon or marginal entropy when $f(\cdot)$ is defined as the probability density (or mass for discrete variables) function, and by definition $H(X) \geq 0$ (Crooks, 2017). Now then, how do we relate the Hamming distance to the Shannon entropy? The Hamming distance is defined on an alphabet of size two, i.e. ones and zeros. The Shannon entropy generalises this to an alphabet of size m. Consequently, the amount of information gained by seeing the next symbol k in alphabet m of a bit string is

$$I(k) = -\log_2 \mathbb{P}(k). \tag{4.40}$$

Thus the average amount of information $I(k)$ per character, contained in a string, is given by a Shannon entropy

$$H(X) = -\sum_{x\in X} \mathbb{P}(x)\log_2 \mathbb{P}(x) \tag{4.41}$$

where X is defined on an alphabet m and $\log_2 m \geq H(X) \geq 0$. Hence, although earlier we used the Hamming distance, to judge model utility and inference capacity, it is not paramount which information theoretic measure we adopt. Ultimately, the ones used herein measure information overlap between the ground truth and inferred sequences. Of course, we have defined the Hamming distance as measuring the distance between two strings, but it can also be defined as the distance between a given string and the origin $\{0\}^n$, for some string of length n. Then we measure the distance from the origin to the string, and the same follows for the Shannon entropy. In this sense they are similar metrics, but the Shannon entropy is defined on a larger alphabet (state-space) $m > 2$.

⋆ ⋆ ⋆

The parameters of the conjugate prior are set as follows: $\mathcal{NIW}(\overline{\mathbf{Y}}, D+2, 0.01, S \times \mathrm{Cov}(\mathbf{Y}))$. Here $\overline{\mathbf{Y}}$ is the empirical mean and S is a scaling factor for the scaling matrix (Murphy, 2012). For all models we place a prior of $\Gamma(1, 0.01)$ on the concentration parameter γ, of the base measure G_0 and use the same discrete measure H for all models. We place non-informative hyperpriors on the space of α and κ and perform inference using sequential Monte Carlo (SMC) and particle Gibbs (Wood et al., 2014). We condition the models on the data and sample state trajectories. *Without Bayesian optimisation* we set the emission prior to $\mathcal{NIW}(\overline{\mathbf{Y}}, 3, 0.01, 0.75 \times \mathrm{Cov}(\mathbf{Y}))$.

For experiments *with* BO we seek to maximise $\log \mathbb{P}(y_{1:T})$, as discussed in §2.6.5, by optimising the model hyperparameters used to sample state trajectories s.t. $\theta \triangleq \{\alpha_\alpha, \alpha_\beta, \gamma, \lambda, C, \nu\}$, where α_α and β_α are the shape and rate parameters of the gamma distribution respectively, which is the prior on the concentration parameter α. For the sticky and stateful HDP-HMM we extend this set to $\{\alpha_\alpha, \beta_\alpha, \alpha_\kappa, \beta_\kappa, \gamma, \lambda, C, \nu\}$. We use the expected improvement as our acquisition function a_{EI} (for details see §2.3). We tested many acquisition functions during our investigation of using BO in this set of experiments, including the most common ones:

- Probability of improvement — see (Shahriari et al., 2016, eq. (42))
- Expected improvement — explained in §2.3
- Entropy search — see (Shahriari et al., 2016, eq. (47))
- Upper confidence bound — see (Shahriari et al., 2016, eq. (45))

After extensive evaluation, we settled on the expected improvement because:

1. Intuitively, the expected improvement has easily interpretable properties: if we maximise a_{EI}, we will either sample from points for which we expect a higher value of f (we will use f as a proxy for our objective function), or points in a region of f we have not yet explored.
2. From the above point, we can conclude that a_{EI} is high when the uncertainty around a point θ is high.

3. The expected improvement elegantly trades off between exploitation (evaluating at points with low mean) and exploration (evaluating at points with high uncertainty), in a way that, for example, the probability of improvement does not.

4. Finally, the properties of the expected improvement, means that a_{EI} is high when the posterior expected value of the loss s higher than the *current* best value $\theta^*_{\mathsf{current}}$.

For all BO experiments we use 1000 particles for inference and use the samples to optimise a_{EI}. Once θ^* is recovered, these hyperparameters are used for the respective model, and experiments rerun for the full particle set $\{2^k \times 10^3 \mid k = 0, \ldots, 4\}$.

Further we make use of two kernels; the radial basis function K_{RBF} and the Matérn 3/2 –kernel K_{M32} (Rasmussen & Williams, 2006). These are popular choices in the literature, we explore both because K_{RBF} is infinitely differentiable, which means that the GP with this covariance function has mean square derivatives of all orders, and is thus very smooth (Rasmussen & Williams, 2006). Smoothness this strong is typically unrealistic for many physical processes, hence K_{M32} is explored as well. It is only once differentiable and, therefore, makes strong assumptions about the smoothness of f (in the Bayesian optimisation sense). Note that continuity and differentiability are both ways of quantifying the smoothness of a function, with differentiability implying greater smoothness than continuity. Which is to say that a function f which is differentiable at all points is fully continuous. To elucidate this point, consider example 4.1.

EXAMPLE 4.1: CONTINUOUS BUT NOT DIFFERENTIABLE

Consider the following function

$$f(x) = x^{\frac{1}{2}} \tag{4.42}$$

and its first three derivates

$$f'(x) = \frac{1}{2\sqrt{x}} \tag{4.43}$$

$$f''(x) = -\frac{1}{4x^{\frac{3}{2}}} \tag{4.44}$$

$$f'''(x) = \frac{3}{8x^{\frac{5}{2}}}. \tag{4.45}$$

Whilst we can easily write down these derivatives for eq. (4.42), this function is still *not* differentiable because $f'(x)$ does not exist when $x = 0$ as demonstrated:

$$f'(x = 0) = \frac{1}{2\sqrt{0}} \to \nexists.$$

Consequently, a continuous function is *smoother* than a discontinuous function like eq. (4.43). Similarly, a differentiable function is smoother than a continuous non-differentiable function.

Performing full Bayesian inference on the model parameters, for all three models, yields the results in the *top row* of fig. 4.16. It is clear from this instance that NMI increases with particle count, but so too does computational cost. Recall that particle MCMC uses SMC algorithms to design efficient high dimensional proposal distributions for MCMC algorithms (Andrieu et al., 2010). That is where their inherent strength is derived. In particular, PMCMC relies on SMC to generate samples of the highly correlated state trajectory within an MCMC sampler (Andrieu et al., 2010). This is to tackle two of the fundamental properties of SMC, as outlined by Frigola et al. (2013):

1. The standard SMC algorithm cannot handle inference over the model hyper-parameters.
2. In general it does not provide an accurate approximation of the full joint distribution, because of path degeneracy (discussed in the preliminaries).

Consequently, we maintain a posterior on our hyperparameters using PMCMC, and leverage the mechanics of PMCMC to receive both a posterior over the model parameters and the latent state-sequence.

Further, model structure plays a large role where the stateful HDP-HMM demonstrates better clustering ability than the other two models. At the same time neither model,

for this low number of particles, performs well, and performs best under SMC inference. Instead optimising the hyperparameters demonstrates a clear gain – even for a signal as noisy as the test case (figure fig. 4.15), for all models. *Why is this the case?*

First, and the likeliest cause of the poor results with full Bayesian inference, are very poorly specified model priors coupled with noisy data. Considering fig. 4.15 purely from a visual analysis, it is especially difficult to separate the dark and light blue states. But even so, the yellow state sometimes also enters the state-domain of the blue states – this is down to the noise model. Considerable amounts of white Gaussian noise were added to the model, this, combined with 'sparse' inference power (i.e. we used a small number of particles) yielded the results seen in fig. 4.16. This was not the case for the same inference task using Bayesian optimisation.

The Bayesian optimisation model that we employed used the expected improvement as the acquisition function. This acquisition function, as noted, explicitly incorporates a tradeoff between exploitation and exploration. Consider this, let f_{min} be the smallest value of f observed thus far. Expected improvement evaluates f at the point that, in expectation, improves upon our value of f_{min} the most. Explicitly this leads for the utility function

$$u(\theta) = \max(0, f_{\mathsf{min}} - f(\theta)) \tag{4.46}$$

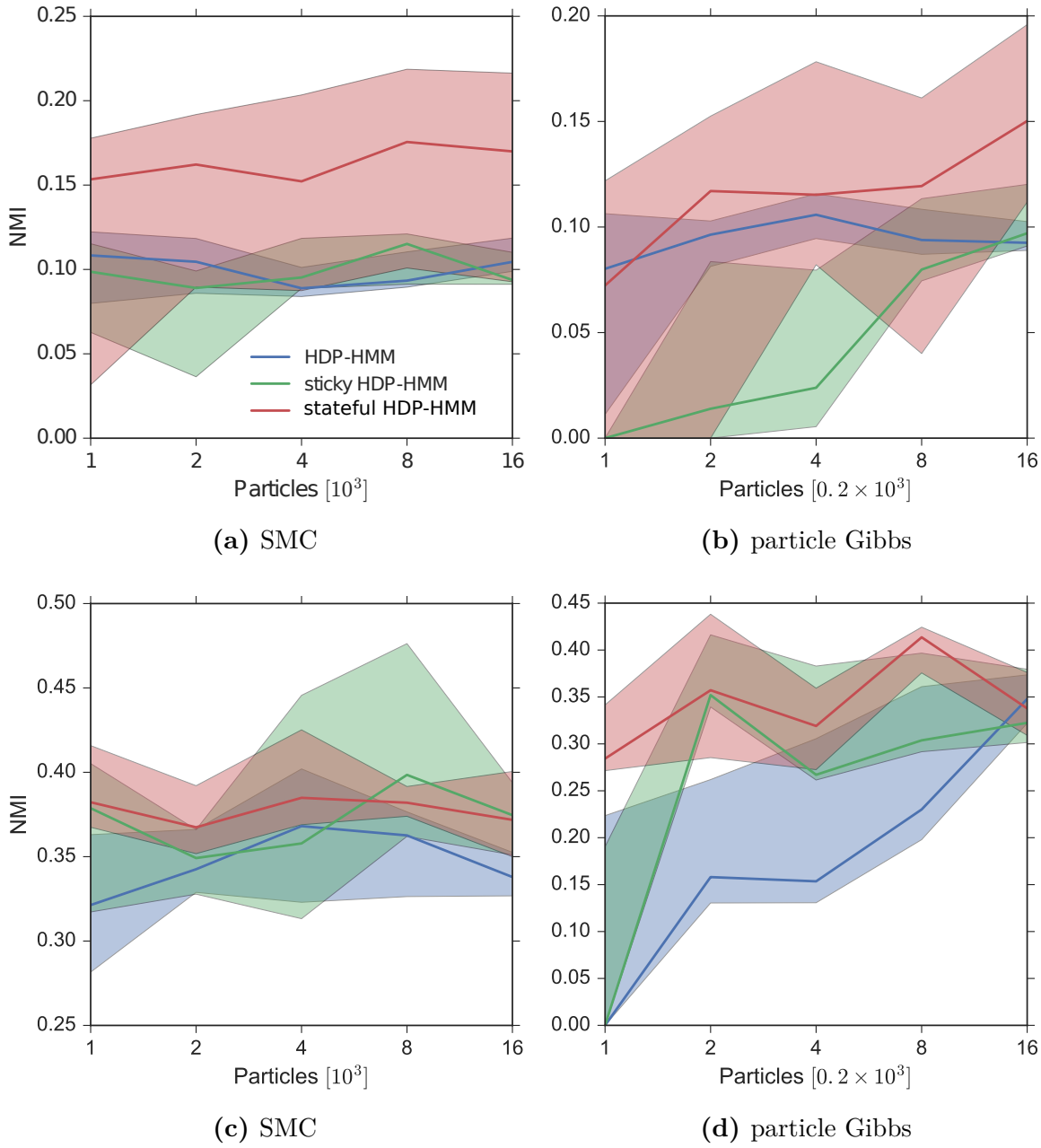
whereupon the acquisition function takes on the form

$$a_{\mathsf{EI}} = \mathbb{E}[u(\theta) \mid \theta, y_{1:T}]. \tag{4.47}$$

Consequently then, under this regime, Bayesian optimisation will select the point with the highest probability of improvement (the one with maximal expected utility). This explains why 'BO + full Bayesian inference' works better than simply 'Bayesian inference' – inference is incentivised to back out of low-utility regions of the state-space, and instead explore those with maximal expected utility. Ultimately this means that better hyperparameters are found faster, for this low particle count. If the particle count were higher, and the experiments allow to run for longer, in the limit, the posterior estimates would converge on the same estimates found by Bayesian optimisation.

However, it should still be emphasised that these are very average results, yielded from running an inference model on a very noisy signal. But in that sense it is also realistic,

when compared to many real-life problems that we face (see real data experiments in §4.4.3 and §4.4.6).

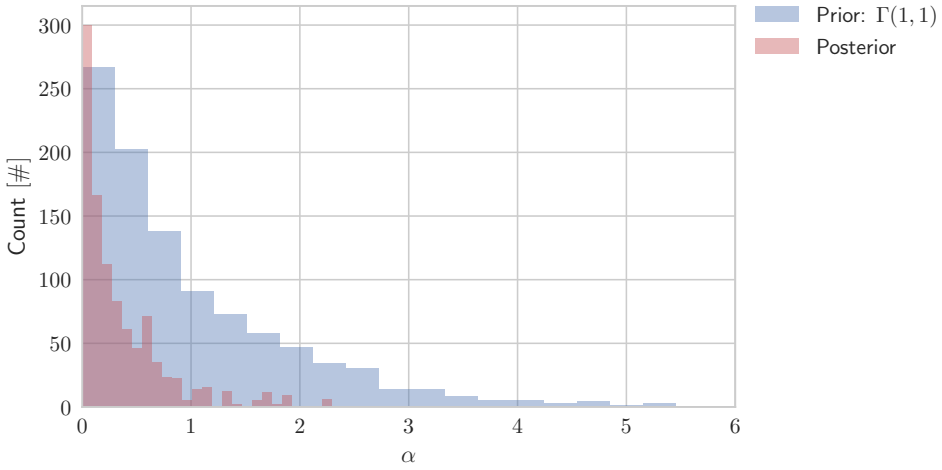


(a) SMC **(b)** particle Gibbs

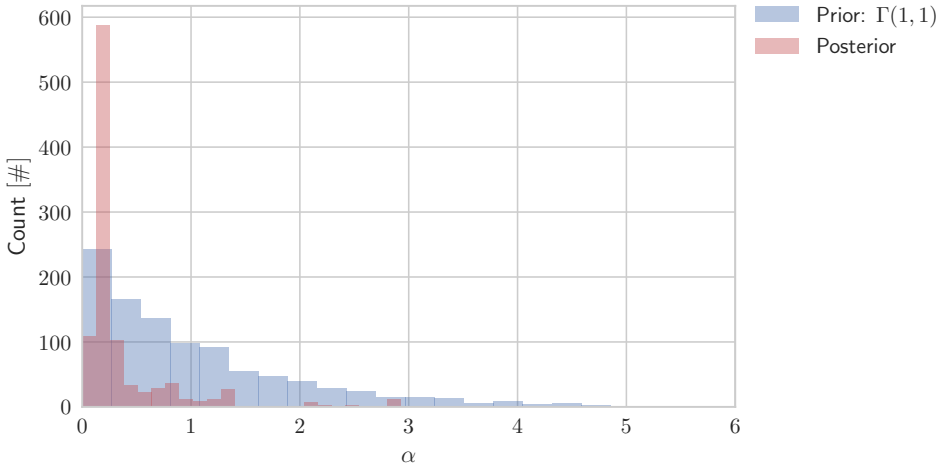
(c) SMC **(d)** particle Gibbs

Figure 4.16: The median (thick line), the 25^{th} and 75^{th} NMI quantile comparison of inferred latent state sequences, and the ground truth, using SMC and particle Gibbs. The first row depicts baseline results without optimised hyperparameters, and the second row shows results with optimised parameters.

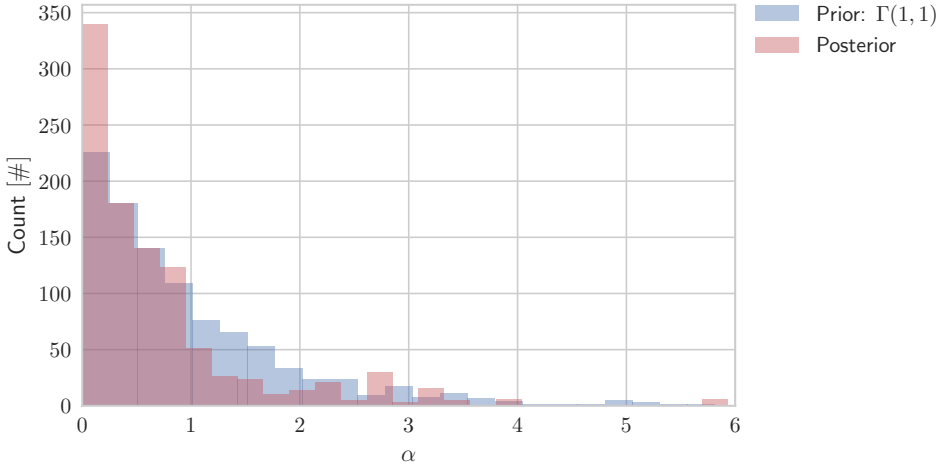
As can be seen from fig. 4.16 it is clear that neither model performs well in the difficult inference regime, in which they were tested. Moreover the small particle count used in these experiments demonstrate that for these methods to be efficiently applied, more particles are necessary to obtain the latent state trajectory.



(a) HDP-HMM



(b) Sticky HDP-HMM



(c) Stateful HDP-HMM

Figure 4.17: Posterior estimates of the concentration parameter for the three models, using 500 particles, 1000 samples and SMC inference.

Further, we can consider the posterior estimate of α parameter of the three models, this is shown in fig. 4.17. In this figure we see the posterior univariate distributions, of the concentration parameter for all models. From this we can clean that the HDP-HMM settles on a posterior which is tightly peaked around small values of α. The sticky HDP-HMM also settles on a posterior estimate which too is highly peaked, the stateful HDP-HMM by comparison does not. From fig. 4.16 it is not clear which posterior estimate is better in this instance, since neither model performs particularly well. Whilst the sticky HDP-HMM centres its mass around a tight posterior, the stateful HDP-HMM spreads it more widely. Benefits can be found with both estimates; first, having a tightly peaked posterior means that the particles *may* have settled on the correct latent density, as particles around this density get consistently high weights during inference. But at the same time, because we made the concentration parameters stateful with the stateful HDP-HMM, we are able to explore this posterior more by virtue of tying an α to each state, and in so doing, make the space of α more heterogeneous.

4.4.3 Human activity modelling

In this section we apply our methodology and models (where we will place extra emphasis on the stateful HDP-HMM, for demonstration purposes), to two challenging labelled human locomotion datasets. The labelling in these datasets is worth pursuing, before going further.

Labelling data is subjective and tedious work. Sometimes subjective annotators make mistakes which add noise into the labels. Treating these noisy labels as the ground truth is typically harmful for most learning methods (Hu et al., 2015). Steps are sometimes taken to alleviate this labelling bias e.g; Hu et al. (2014) suggested a method that models each label as a multinomial distribution rather than deterministic. In (Hu et al., 2015), the authors treat all of the labels as noisy data, and add minor probability mass to incorrect labels enabling the model to converge to a better representation of the actions. Hence, with this in mind we use the labels with caution, in inferring statistical properties of the observations. Hence we use the labels merely as a guide, rather than absolute truth, providing a useful measure of our method utility in segmenting locomotion.

4.4.4 PAMAP2 physical activity modelling

The PAMAP2 dataset (Reiss & Stricker, 2012) contains observations of 18 different physical activities such as running, cycling and walking (each subject performs a smaller subset of these) performed by nine subjects wearing three inertial measurement units (IMU) and a heart rate monitor. Where each time-slice is indexed by i i.e. $y_i \in \mathbb{R}^{3\times 17}$. Where each IMU recorded large and small scale 3D acceleration, 3D gyroscopic readings and 3D magnetometer measurements. We select the gyroscopic observations s.t. $y_i \in \mathbb{R}^9$ and $n = 2528$ (total of 22,752 data points) as they are perceptively the more complicated segmentation case. Lets consider this further.

Gyroscopes measures orientation and angular velocity by operating a freely spinning wheel in which the axis of rotation self-selects the orientation. Compared with acceleration and magnetometer signals, gyroscope measurements provide a more complex analysis regiment because we are consider the angular velocity, and not absolute velocity. The last part is important because our accelerations and magnetometer results are absolute w.r.t. to their measurement axis. Angular velocity on the other hand provides the temporal rate of change in angular displacement relative to the starting position, consequently is set on a bounded domain $\omega \in [0, 2\pi]$. Acceleration and magnetism are unbounded measures and hence are allowed to take on possibly more distinct and outlying values for certain classes, thus providing requiring more specific instances of the hyperparameters, the setting of which we use for the segment identification in our model formalism. Angular velocity, having a fixed domain, will necessarily require instances of hyperparameter settings which are similar, thus making it harder to draw inferences on what segment label it should be afforded any particular setting.

⋆ ⋆ ⋆

Continuing, note that the authors have provided a special class for "transient motion" such as switching between the performance of different activities. These regions are of particular interest since the switching behaviour of the observations serves to inform the segmentation of atomic motions (if any) which may or may not be present in the observations. For each set of experiments we used 1000 particles and 100 samples. We also explored multiple inference methods including: particle Gibbs, SMC and particle independent Metropolis-Hastings (Andrieu et al., 2009, §3). The results for the latter two

methods can be found in appendix B. Hence, results are shown in fig. 4.18 for particle Gibbs, for our entire model set introduced up to this point. Common model parameters are shown in table 4.2.

Table 4.2: Common model and emission priors used for PAMAP2 experiments.

Model parameter	**Prior**
α, γ, α', γ'	$\Gamma(0.5, 1)$
κ	$\Gamma(0.5, 1)$
H_θ	$\mathcal{N}(0, 1)$
$H_\mathcal{D}$	$\text{Pois}(\lambda)^\dagger$
μ_0	$\overline{\mathbf{Y}}$ (empirical average)
λ_0	$D + 2$
Ψ	$0.75 \times \text{Cov}(\mathbf{Y})$
ν	0.1

† Used as part of parametric mixture duration distribution, where $\lambda = \{100, 300, 600\}$ with weights $w = \{0.25, 0.25, 0.5\}$.

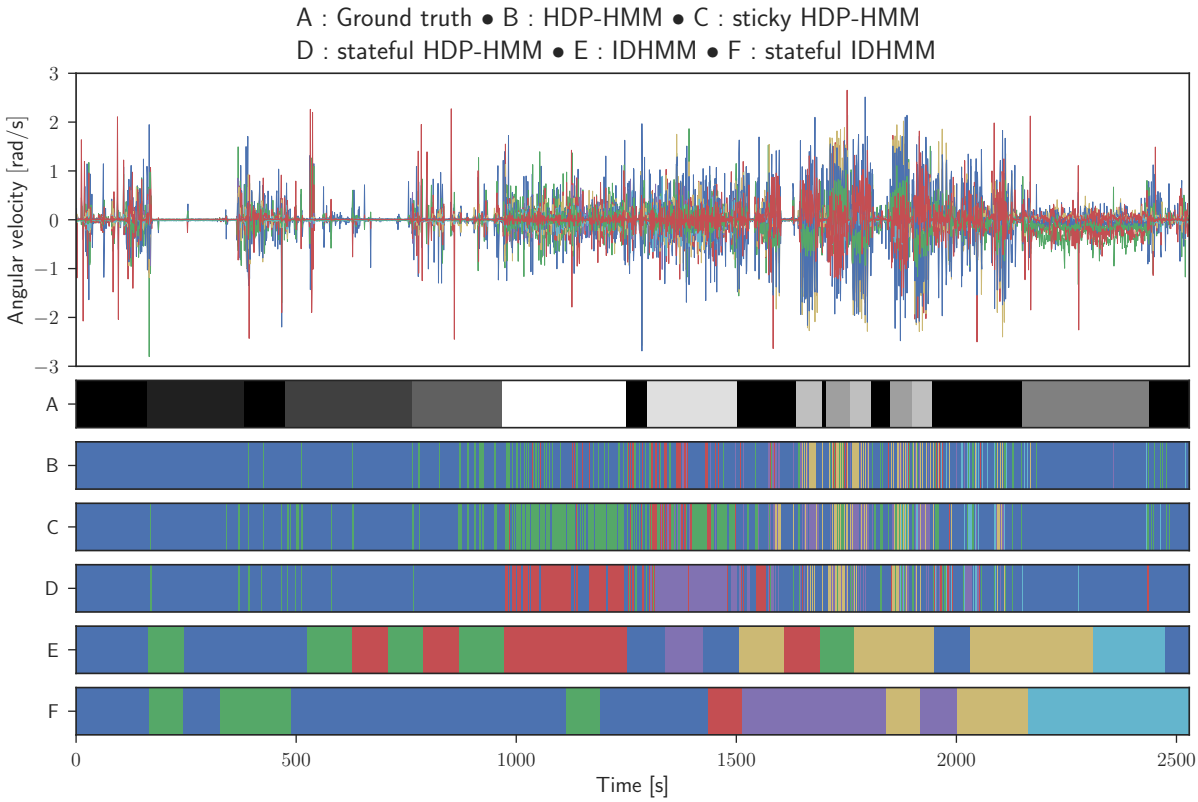


Figure 4.18: From the top, the **first panel** shows the observations used for segmentation. The **second panel** shows the manual segmentation. The **following panels** depict the model segmented sequences, for the highest log-likelihoods. Inference was provided by particle Gibbs.

The results in fig. 4.18 show different segmentation performance for our given models.

Though we have noted that this is a complex segmentation task (with segmentation that is also not absolute), some models do perform better than others. Notably the HDP-HMM (A) and the sticky HDP-HMM (C), introduce too many states and switch too fast in between them. The stateful HDP-HMM (D) does better, but it too introduces too many states, but is nonetheless persistent. The IDHMM (E and F) models do a better job of segmentation but with a comparatively strong prior imposed. Even then though, the models hone in on regions where movement is not supposed to be found, and miss other regions where it is. The question that now arises is: *can we improve the models without increasing the particle count?* To investigate this question we will employ the stateful HDP-HMM because its performance in fig. 4.18 is at best average, but better than models B and C. Hence consider the transition probabilities for this model, under the data, calculated from multiple inferred state trajectories. This is shown in the bottom panel of fig. 4.19.

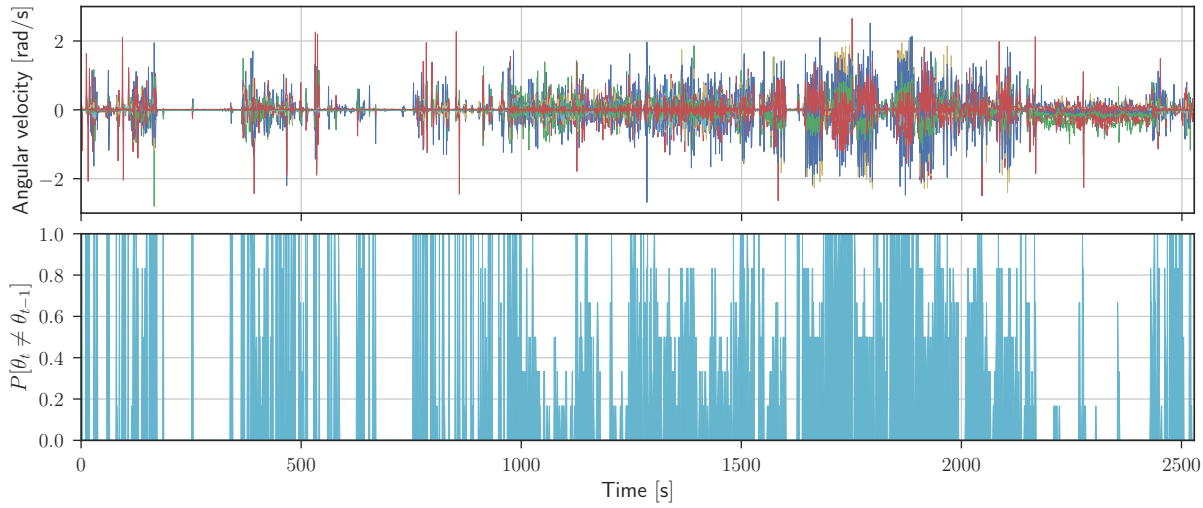


Figure 4.19: State transition probabilities shown in the **bottom panel** calculated from multiple runs of the stateful HDP-HMM. The **top panel** shows the observations sequence.

What fig. 4.19 demonstrates is a propensity for state switching. Moreover it also demonstrates an incorrect state-cardinality, being far too high. This is derived from a misspecified prior which cannot correct for the observations, since there are not enough of them, nor enough particles. One solution is to tune the prior using Bayesian optimisation.

4.4.4.1 PAMAP2 with Bayesian optimisation

We perform inference on the stateful HDP-HMM, using SMC with 1000 particles, where the size of the model now necessitates Bayesian optimisation. BO is performed over the space of model hyperpriors and some of the parameters of the NIW prior, such that $\theta = \{\alpha_\alpha, \beta_\alpha, \alpha_\kappa, \beta_\kappa, \gamma, \lambda, \nu, S\}$. Where α_- and β_- are the shape and rate parameters for the Gamma prior on the concentration parameter α for the DP, and the self-transition parameter κ. We explore kernels K_{M32} and K_{RBF} using the same acquisition function a_{EI}.

The PAMAP2 datasets is significantly more challenging than the TUM Kitchen (which we explore in the next section §4.4.5), particularly as we have chosen a difficult state representation. Starting with the inferred state cardinality, using K_{M32} yields seven and K_{RBF} six. Compared to true value of nine. Perhaps the biggest problem with the observation is that they are very noisy, and as such, the clustering becomes challenging, as there is little to distinguish inbetween features; consequently assigning the wrong labels to activities. It is worth noting that activities with relatively little noise such as 'walking' (start: $\sim$ 2200s) and high noise such as 'vacuum cleaning' (start: $\sim$ 1300s) are labelled with ease since these features are easy to distinguish from noise and or other similar activities (in feature space). We achieve a similar NMI score of 0.54 and 0.41 for K_{M32} and K_{RBF} respectively. This is to be expected given that the smoothness assumption the latter kernel makes, are inappropriate for this labelling task. Further, by considering the expected switching probability in the bottom two panels of fig. 4.20, we can use these expectations as a different form of labelling. The trajectories that we display in the main panel of figure fig. 4.20, is indeed the maximum log-marginal-likelihood sequence under the observations. However we use all sampled sequences to generate the expectation plots. As is shown, they do indeed demonstrate more certain segmentation as they are weighted probability functions. Thus, we take uncertainty into account; although state sequences are poor compared to the ground truth, we can estimate validity of the inferred state switches in the sequence, using the calculated expectations.

Consider further the distribution over the states, for the max log-likelihood sequence, fig. 4.21.

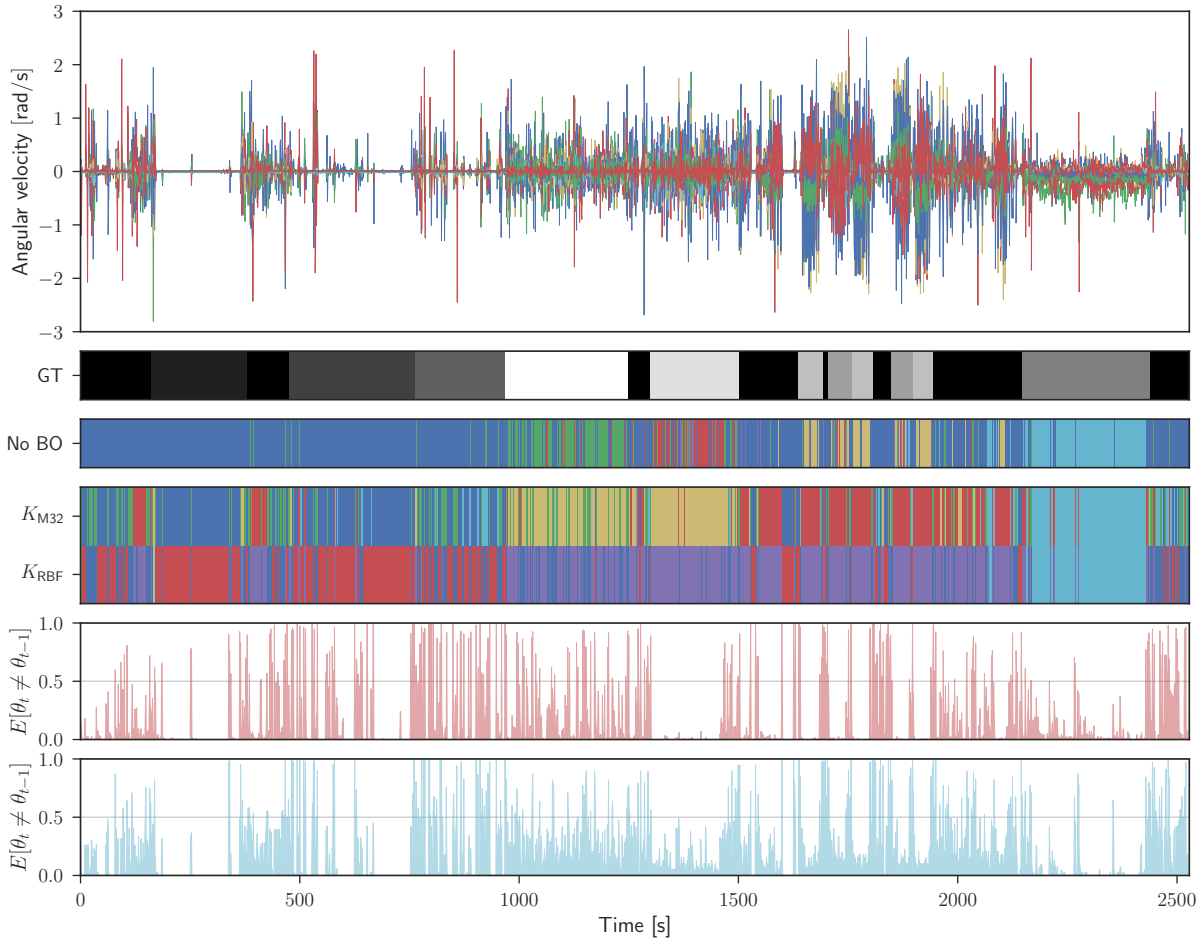
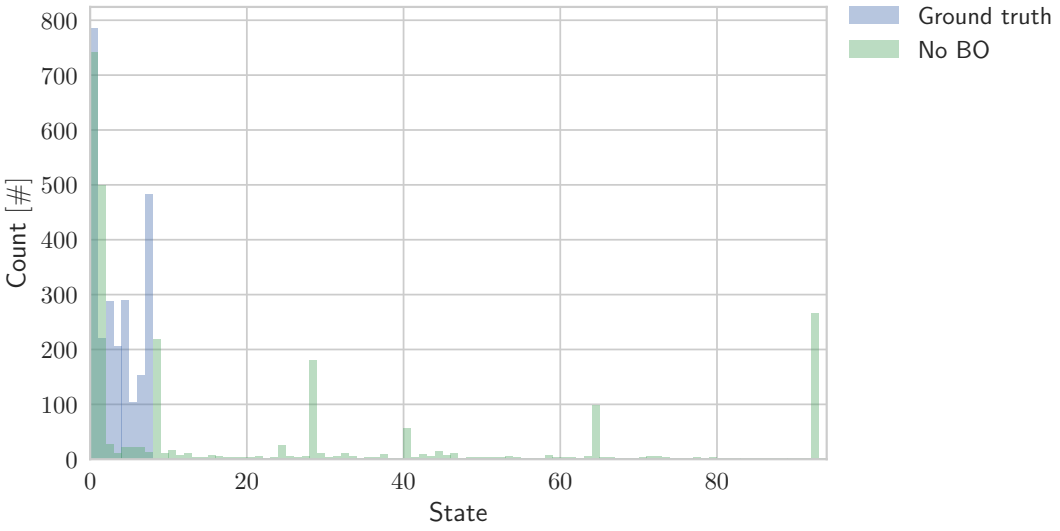


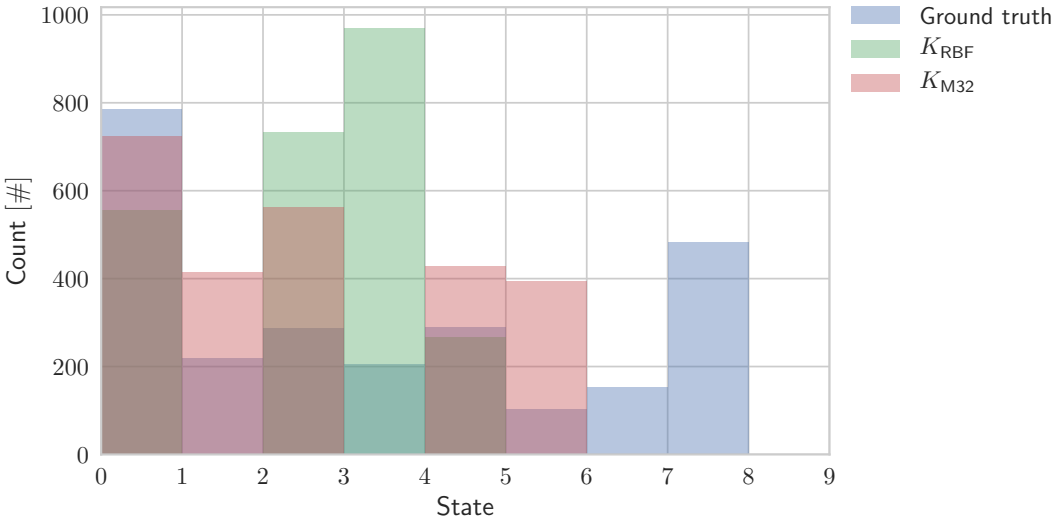
Figure 4.20: From the top, the **first panel** shows the observations used for segmentation, the **second panel** depicts the manually labelled state sequence and the **third panel** shows the state SMC inference *without* Bayesian optimisation. In the **fourth panel** the two inferred segmentation sequences with kernels K_{RBF} and K_{M32} respectively, for the highest log-likelihoods. The **bottom panels** show the expected state switching probability under the two kernels respectively. Adapted from (Dhir et al., 2016b).

The above distributions demonstrate the utility that BO can have on a complex inference task such as ours. First, we see some evidence that BO can leverage against the label-switching problem of BNP, by focusing particles on high-yield regions of the latent-space. Without this active search over the latent space, we receive the familiar plot in the top panel of fig. 4.21.

By using little prior knowledge of the state space and the data at hand, we demonstrated through the use of the stateful HDP-HMM and Bayesian optimisation, a methodology for sampling complete activity sequences. We have shown that quality of the feature set greatly influences the utility and accuracy of the recognition. It is also clear that there is a marked, positive difference, between the state-sequence without BO in fig. 4.20 and those with. By using probabilistic programming, it is possible to leverage powerful inference



(a) No Bayesian optimisation.



(b) Bayesian optimisation, with two different kernels.

Figure 4.21: State cardinality distributions. The top panel shows the max log-likelihood sequence, state-cardinality histogram. The bottom panel shows the max log-likelihood sequence, state-cardinality histogram.

methodologies, in a black-box manner. Married with BO, these methods define a powerful way in which activity recognition can be induced in an almost automatic manner. Since we do not need to preselect the state space cardinality nor model hyperparameters (beyond a good guess).

There are number of different ways in which results can be improved. The most obvious, to start, is to pick a feature set that captures the full modalities of human locomotion in some setting. Secondly inference algorithmic development is ever ongoing, and will

become more adept at performing inference in high-dimensional state spaces. Finally, BO has been used throughout this work, but usually only with default settings and standard kernels. Kernels are the most important item in BO, and should be chosen with care. Or better yet, their structured learned from observations as well (see the work by Duvenaud et al. (2013)).

Finally, it is not difficult to envision how these results could be utilised. In real operating environments robots typically do not have access to ground truth. The information might not be available, too expensive, time consuming or difficult to collect, thus one has to rely on unsupervised approaches. One such principled approach has been demonstrated in this section.

4.4.5 TUM everyday manipulation modelling

Having demonstrated the utility of using Bayesian nonparametric SSMs and Bayesian optimisation, in environments with high uncertainty, we apply this method set to the TUM-Kitchen dataset (Tenorth et al., 2009) – again using the stateful HDP-HMM as our base model.. It is recorded in a home-care scenario where subjects perform daily activities of living, in a kitchen. The kitchen is equipped with a set of ambient sensors and four static overhead cameras. The full body joints are tracked with a motion capture system. Labels are provided for the left and right hand, and the trunk of the subject. We use joint positions as they are a common feature set for locomotion segmentation (Sung et al., 2011). Where $\mathcal{D} = \{y_i\}_{i=1}^{n}$ and $y_i \in \mathbb{R}^{28\times 3}$ s.t. there are 28 tracked 3D joints. We select a subset consisting of the left arm (upper arm, forearm, hand and fingers) s.t. $y_i \in \mathbb{R}^{4\times 3}$ and $n = 1000$ (total of 12,000 datapoints). Results are shown in fig. 4.22.

It is clear that our unsupervised segmentation is different to the ground truth. This was expected as has been discussed. Of greater interest is that the correct number of activities has been inferred (eight) for the K_{M32} kernel, whereas using K_{RBF} inferred a state cardinality of ten. What is worth noting is that the manually labelled sequence is segmented into highly discrete activities such as “reaching” or “carrying while locomoting” (Tenorth et al., 2009). Locomotion, however, is not that discrete and instead consists of atomic motions, the combination of which serves to create larger locomotion behaviours. Tellingly, this periodic behaviour is indeed what is demonstrated in the inferred latent

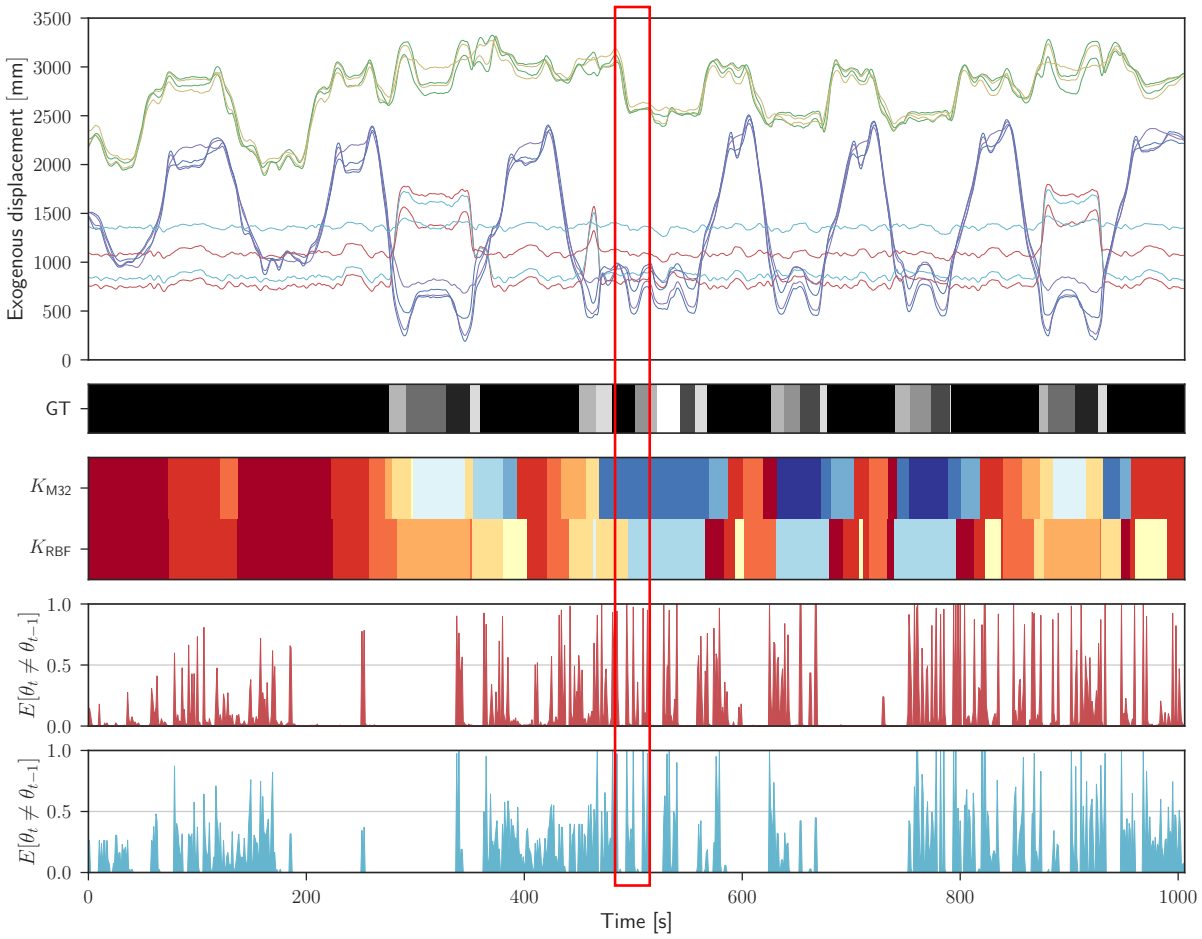


Figure 4.22: From the top, the **first panel** shows the observations used for segmentation, the **second panel** the manually labelled state sequence, the **third panel** shows the two inferred segmentation sequences with kernels K_{RBF} and K_{M32} respectively, for the highest log-likelihoods. The **bottom panels** show the expected state switching probability under the two kernels respectively. Adapted from (Dhir et al., 2016b). The red box depicts the region around the 500s mark.

state sequences for the left hand. It being indicative of the periodicity and swing, often exhibited in human locomotion. That being said, it is also clear that our segmentation demonstrably fails to register certain activities. Indeed, consider the bottom two panels of fig. 4.22, where the expected probability of a state switch is displayed; $\mathbb{E}[\theta_t \neq \theta_{t-1}]$. For example there is a clear region in the middle of dataset (around 500s) where the subject is grasping and reaching for objects, that is evidently not being registered with our methods. This is most likely due to an unrepresentative feature set. On the other hand the expectation plots demonstrate in more detail the dynamic switching behaviour of the dataset. From it, it would not be unreasonable to suggest that the labelling provided for this dataset is too coarse. This becomes evident when cross-validating with video evidence. Despite this, we achieve an NMI score of 0.54 and 0.51 for K_{M32} and K_{RBF}

respectively.

Further, note also that fig. 4.22 demonstrates some high switching probabilities in the bottom two panels, which are not reflected in the inferred state-sequence in the middle panels. This is because we calculate $\mathbb{E}[\theta_t \neq \theta_{t-1}]$ for all sequences, but the state-sequence shown has the highest log-likelihood under our model (which does not necessarily have to follow the expected switching distribution). In effect we can regard the switching distribution as having been tempered by the low log-likelihood sequences, which is why expectation mass is added in places where a switching does not occur in the final sequence (i.e. the middle panel). We are likelier to see more representative expected switching behaviour, if we only used the top quartile of results, rather than all sequences.

We now demonstrates that our methods are also applicable in domains outside prosthesis design. Indeed, they are applicable to any time-evolving phenomena.

4.4.6 Lion behavioural modelling

A generative model describes a process, usually one by which observable data is generated. Generative models represent knowledge about the causal structure of the world. These models can be used to answer many different questions, by employing conditional inference. It is possible to use deterministic generative models to describe the ways a process could unfold, but due to sparsity of observations or actual randomness there will often be many ways that our observations could have been generated. We are interested in sequential data, from which we aim to infer meaningful states, along with the defining characteristics of each state (Johnson & Willsky, 2013). More often than not, such state discovery needs to be done in an unsupervised fashion, as the application commands little or no information about the latent state cardinality nor the duration (dwell-time) of those states. This scenario is typified by animal behaviour modelling. Though different to the applications we demonstrated in §4.4.5 and §4.4.4, the fundamental problem remains the same: dynamic uncertainty in time.

Animal accelerometer data allows zoologists to identify important correlates and drivers of animal behaviour. Use of accelerometers is widespread within animal biotelemetry as they provide a means of measuring an animal's activity in a meaningful and quantitative way, where direct observation is not possible. In *sequential* acceleration data there is a natural

dependence between observations of movement or behaviour, a fact that has been largely ignored in most analyses (Leos-Barajas et al., 2017). Recordings are typically sampled at a high temporal resolution, sometimes for years at a time, using tri-axial accelerometer tags (Lush et al., 2015), which quickly results in terabytes of data that present various challenges regarding transmission, storage, processing and statistical modelling. Indeed, much of the focus in the analysis of acceleration data has been on identifying patterns in the observed waveforms that correspond to a known behaviour or movement mode. The latter can be achieved by employing statistical classification methods, and entails observing the animal, *manually assigning labels* corresponding to *known* behaviours to segments of the data, and training a model using the labelled data in order to subsequently classify remaining unlabelled data based on certain *chosen acceleration features* deemed to be pertinent by domain experts.

Consider the recent work of Pagano et al. (2017) wherein the authors use tri-axial accelerometers to identify wild polar bear behaviours. They note that identification of wild animals can be facilitated using captive counterparts, as their accelerometer signatures are generally assumed to be similar to those of their wild kin (Pagano et al., 2017). They use their captive bears as surrogates for wild ground-truth behaviour, upon which they model polar bear behaviour on sea ice and land, using random forests classification and hand-engineered features. Their results, though of good accuracy, rely on hand-engineered features, large assumptions about captive and wild behaviour and a fundamental need for labelled data to infer behaviour. Along this trail of thought, the work of McClune et al. (2014) is relevant to our discussion. Therein, the authors fitted tri-axial accelerometers to a tame and a captive Eurasian badger, whereupon it was allowed to roam free in an enclosure, whilst movements were video recorded and used as ground-truth for its behavioural states. Again, features were hand-engineered using, e.g., acceleration magnitude and principal component analysis. The $k-$nearest neighbour classifier and decision trees were used to automate classification of behaviours (McClune et al., 2014) – for a primer on these methods see §2.2. Their success ranged from 77.4% to 100% classification accuracy, though again deploying a highly laborious process, where a human is necessary for the extraction of the video ground-truth (and classification is conditioned on the ground truth existing at all). The work by Leos-Barajas et al. (2017) is faced with precisely this challenge, where they seek to measure an animal's activity in a

meaningful and quantitative way where direct observation is not possible (Leos-Barajas et al., 2017). In doing so, they investigate a marine and an aerial system (sharks and eagles). They used the classical HMM to effect supervised and unsupervised learning of animal activity. In the former case they used overall dynamic body acceleration of a shark, over one second intervals, as the feature set, then using a two-state HMM they modelled, in order to ascertain, if the shark was *more* or *less* active during various point of the day. In the unsupervised setting they do not strictly employ the HMM for classification, but rather as a simple approximate representation of the real data-generating process (which, they note, "may or may not entail that the nominal HMM states are biologically meaningful" (Leos-Barajas et al., 2017)). In the latter case, an HMM is used to segment unlabelled acceleration data into a finite set of pre-specified categories (Leos-Barajas et al., 2017). They go on to show that the metrics which they derive from the learned eagle -states provide meaningful insight into activity levels and thus can thus lead to biologically interpretable states. Their study is similar to that of Phillips et al. (2015), in which the authors applied HMMs in an unsupervised context to model the behaviour of free swimming tuna from vertical movement data collected by data-storage tags (Leos-Barajas et al., 2017, §3.3).

As we have demonstrated some studies *do* employ models with temporal dependency (e.g., the Markov assumption) but the far more popular method is to deploy classification algorithms that do not, e.g., support vector machines (SVM) or random forests see, e.g., (Carroll et al., 2014). We, like Leos-Barajas et al. (2017), note that disregarding the serial dependence in the acceleration data usually is not a realistic assumption. Moreover, independent and identically distributed statistics (i.i.d.) pose a particular risk if "inferential statistics are applied to the output of say a machine learning algorithm" (Leos-Barajas et al., 2017).

⋆⋆⋆

To ensure the practical relevance of our analyses, this project was undertaken as a joint venture with the zoologists at our institution who study the particular pride of lions whence our biologger observations originate. We demonstrate our methods by applying them to lion behaviour segmentation, to better understand their ecology.

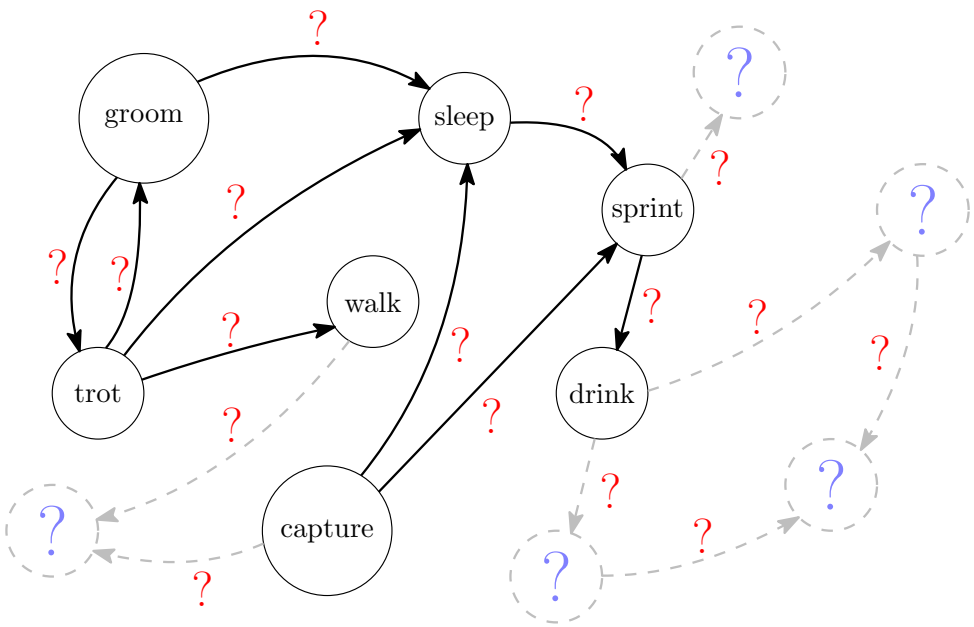


Figure 4.23: The diagram graphically illustrates the problem a Bayesian nonparametric is faced with; the transition dynamics, in red, have to be learned supposing we have a good idea of what some baseline activities look like – these are marked with solid circles. But given that our understanding of the behavioural dynamics of African lion is small, there will be a number of states, marked with dashed circles, that will need to inferred from the time-series observation. Thereupon the transition dynamics from these inferred states, will also have to be learned.

Biologger observations are becoming increasingly popular tools for animal behaviour research. The number of studies using accelerometers in particular, has increased rapidly over the last 15 years due to the advantages offered over methods relying solely on direct observation (Brown et al., 2013). While direct observation may be the only viable means of studying animal behaviour in certain cases, it can pose several difficulties which may include biases suffered as a result of observer presence (Gutzwiller et al., 1994) or the inability to continuously observe the focal animal if it is an elusive species, or a species that occurs in inaccessible habitats. The African lion is an example of a species for which behavioural research can benefit from accelerometer data-loggers due the challenges associated with keeping study individuals in sight continuously while avoiding influencing their behaviour. However, with the ability to record continuously at sampling frequencies as high as 10,000Hz (Brown et al., 2013), accelerometers generate extremely large datasets which are impossible to classify manually, which is why unsupervised learning could help.

The majority of studies which make use of machine learning to classify large accelerometry datasets, tend to rely on supervised learning techniques (Brown et al., 2013) or very

coarse quantification of activity (Noonan et al., 2014). While such techniques have proven effective for many studies focussing on select, broad behavioural states such as 'stationary', 'mobile' and 'feeding' as shown by (Grünewälder et al., 2012) on the cheetah, they are potentially limiting for those aimed at developing detailed activity budgets, as detailed manual labelling can be labour intensive and difficult. Hence, in this study we seek to dwell deeper by investigating, on a per-second basis, tri-axial accelerometry $(\ddot{x}, \ddot{y}, \ddot{z})$ and magnetometer $(\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z)$ observations, collected from a male lion using a 32Hz sampling rate. *Unlike* previous studies, we shall employ a novel form of feature engineering, for our time-series observations, using the recent development of the variational autoencoder (Kingma & Welling, 2013) – see §2.1 for background notes on this model.

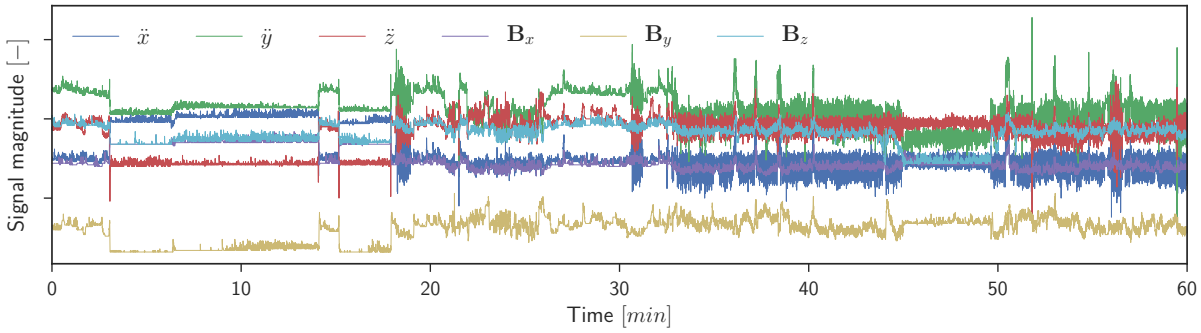
4.4.6.1 Unsupervised feature learning

The study by Rahman et al. (2016) presented an application of autoencoders (AE) to temporal tri-axial accelerometry observations. They used it to effect unsupervised feature learning, later used for supervised classification of cattle behaviour. This ties in with chapter 3 where our feature learning was supervised instead. But the AE data-driven approach is one which we shall espouse with the difference that we instead use *variational* AEs (VAE), whose generative nature has the advantage that the learned features are easier to interpret. In this study, we used a VAE to perform unsupervised feature learning – see fig. 4.24c. We use the VAE to receive a low-dimensional representation of our data, but also to learn a maximally representative feature. It is on these learned features that we perform our further analyses using state space models implemented in Anglican.

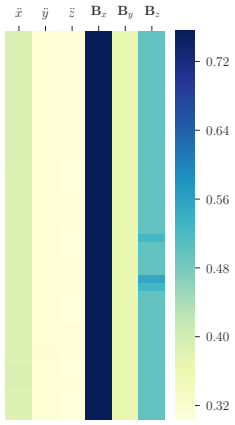
Note that we have used windows of 3s, this size was empirically determined to work best. Larger window sizes did not add much by way of information, whereas smaller ones were not informative enough to be useful. Hence, the size was set to 3s to best tackle this trade-off.

4.4.6.2 Fuzzy ground truth

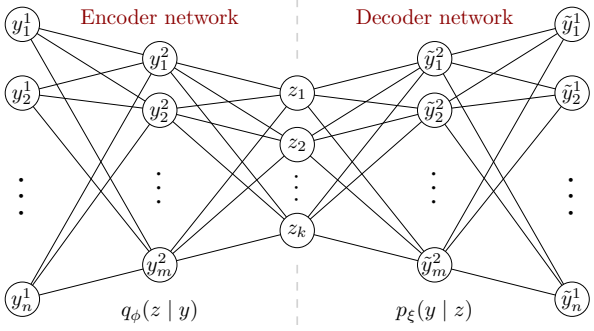
Ground truth (GT), as we have already alluded to, is an elusive property in the zoological domain. In §4.4.6 we noted how e.g. video was used as a form of GT. In our case, GT, or labelled observations, are received via sound (Trethowan et al., 2017). The collar of each lion, apart from being equipped with sensors that log physical variations (e.g.,



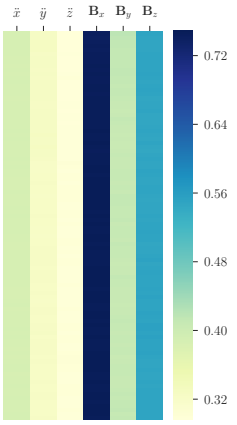
(a) One hour of raw data captured at 32Hz, containing a total of 115,200 multivariate observations.



(b) Example of original $3s$ feature (96×6).



(c) General structure of variational autoencoder.



(d) Reconstructed example feature of size (96×6).

Figure 4.24: The top panel shows the raw data used for this study, consisting of tri-axial accelerometer and magnetometer readings taken from the lion's collar. The bottom figures display the variational autoencoder framework, with original (b) and reconstructed (d) features shown. Adapted from (Dhir et al., 2017c). Note that the way information flows is from left to right in fig. 4.24c, where a window of measurements (3s) are mapped into a latent representation and subsequently mapped out of this latent space, and back into the observed (3s) space again.

acceleration), also log the audio of each animal. This enables the zoologist to get a measure of the animal's activity at time t. It also means that in order to ascertain a dataset that can be used for statistical learning, an exceptionally expensive process takes place where a human listens to an audio recording. For it to be of any use though, that recording has to span not hours, but days. Consequently this type of labelling is prohibitive due to its huge cost in man hours, but it also needs to be performed by the same person, as to remove as much bias as possible (e.g., our dataset contains 'trot' and 'walking' - two activities that are perceptively similar). This is another reason why unsupervised learning could prove preferable, being solely observation-driven. Furthermore, whilst it is true that a human does listen to the lion for a significant portion of time, she does *not* listen to the

whole recording (which again can span days). Instead, recordings are sub-sampled, where, e.g., every other minute (or five minutes in some cases) are monitored and the inferred label (based on the zoologists' interpretation of the lion's *audible* activity at that point in time) is interpolated until the next sampling point. All of which leads to a form of semi-GT.

Consequently, the labour intensity involved in extracting this semi-GT demonstrates why unsupervised methods, such as those demonstrated within, are useful, as it is not practical for humans to be involved in large-scale (years) behavioural segmentation. Moreover, we say *semi* because obtaining perfect information regarding the lion's behaviour over one day, would require a second-by-second log, as well as video surveillance, to be sure as to the validity of the labelling. That being said, the semi-GT is still very valuable as it can guide inference towards appropriate model selection.

4.4.6.3 Model evaluation

Before describing the minutia of our experiments, it is important to re-iterate the purpose of this exercise, and the potential value it could have for zoological studies. Whilst a human will need to be in the semi-GT extraction-loop, we propose that the methods within can function as a conduit for *behaviour discovery*. Differently put, we posit that our methodology can function as a useful tool for zoologist, as the unsupervised segmentation will allow them to hone in on regions of interest, and consequently will allow them to more intelligently choose regions of interest for their work. Upon which their semi-GT can be used in any of the standard supervised classification methods which hitherto have proved their worth in this domain (when good GT is available).

For our experiments, we used 10 hours of labelled[2] observations for one lion, part of a pride currently being studied by our institution. After experimenting with several window sizes, we settled on 3s as providing a useful level of granularity. As such, each observation $y \in \mathbb{R}^{96\times 6}$ (see fig. 4.24b), was normalised, passed to a VAE, where a low-dimensional latent representation $z \in \mathbb{R}^3$ was extracted. The sequence of latent representations is what we used as input for our models, all of which were written as probabilistic programs, to which we applied black-box SMC inference. We used the same conjugate prior as in §4.4.1. For details see table 4.3. In summary, the purpose of these methods is not to

[2]In the sense in which this exposition is framed.

Table 4.3: Experimental model and emission priors, used for inter-model comparison.

Model parameter	Prior
α, γ	$\Gamma(0.5, 1)$
α', γ'	$\Gamma(1, 1)$
κ	$\Gamma(5, 1)$
H_θ	$\mathcal{N}(0, 1)$
$H_\mathcal{D}$	$\mathcal{U}(1000)$
μ_0	$\overline{\mathbf{Y}}$
λ_0	$D + 2$
Ψ	$C \times \mathrm{Cov}(\mathbf{Y})$
C	$\mathcal{U}(0.5, 2)$
ν	$\mathcal{U}(0.1, 2.0)$

segment the signal into 'correct' features (given that no true form of the ground truth exists). Rather, the purpose, given limited and noisy information, is to detect regions of interest (as opposed to, e.g., large regions of a resting behaviour). Results are shown in fig. 4.25.

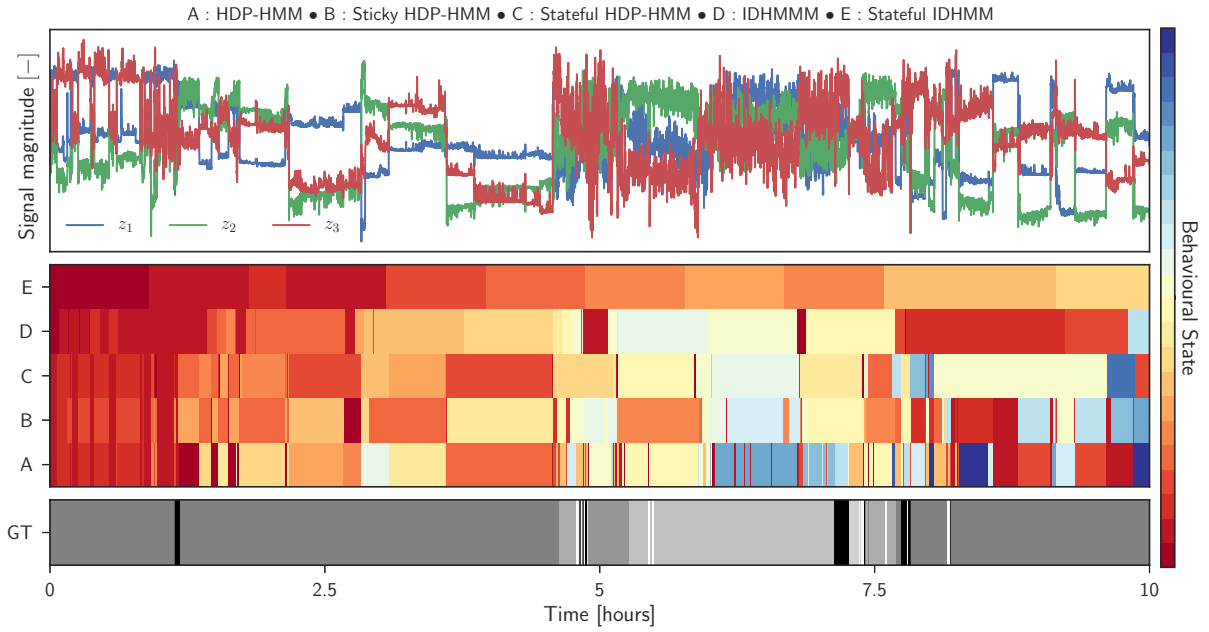


Figure 4.25: The **top panel** displays the feature-set over which inference was performed. The **middle panel** shows the inferred state trajectories, with the highest log-marginal likelihood $\log \mathbb{P}(y_{1:T})$ for all models. The **bottom panel** displays the manually labelled ground-truth which serves as a comparison to our unsupervised labelling. The colorbar maps the numbers of inferred states to each model heatmap in the middle panel. Adapted from (Dhir et al., 2017c).

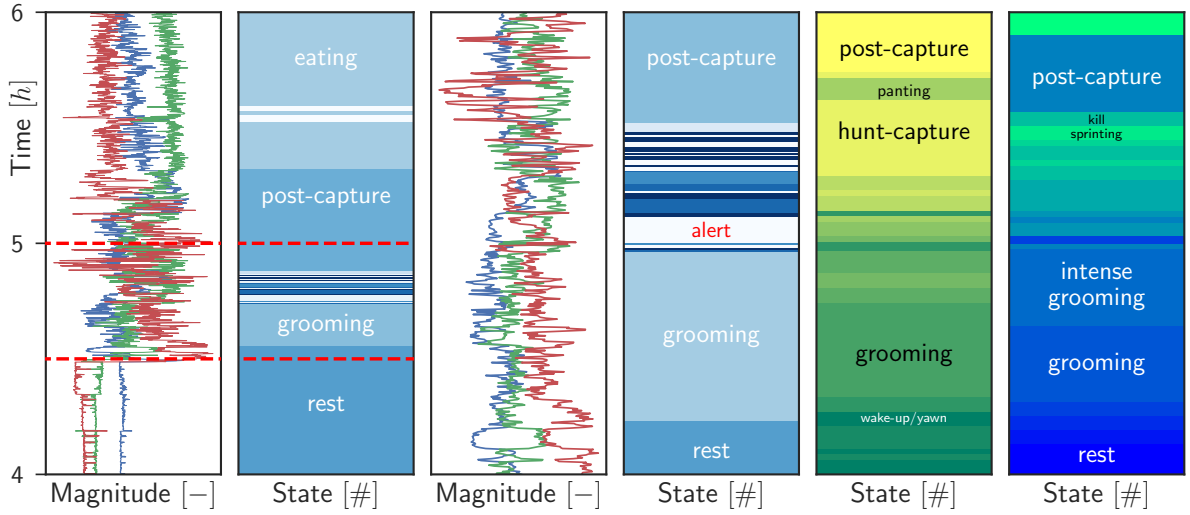


Figure 4.26: Detailed depiction of a two hour segment which features hunting behaviour, using the IDHMM and the stateful IDHMM. The **first two panels** (from the left) show a two hour segment with 'ground truth', the **second pair of panels** show a zoomed in 30min period in which multiple behaviours exist (some of which have been annotated). The **final two panels** show the inferred behaviour sequences using the IDHMM and the stateful IDHMM (last panel). Adapted from (Dhir et al., 2017c).

4.4.6.4 Detailed analysis of hunting segment

In fig. 4.26, we demonstrate the utility of the IDHMM and the stateful IDHMM, on a smaller segment of fig. 4.25. In this experiment, we sought to understand if our methods could accurately segment fast-changing animal behaviour, specifically that related to hunting. Panel four, from the left, in fig. 4.26 contains the manual labelling of this segment, followed by the IDHMM and stateful IDHMM inferred state trajectories. For model and conjugate prior details, see table 4.4. We experimented again with SMC using 1000 particles and 500 samples, as a well as a mixture of Poissons for a duration prior.

Post-analysis showed that the models are reasonably successful in segmenting the time-series from a zoological point of view; subtle and short behaviours can be picked out from the audio, which the models 'accurately' recognise from the input sequences. Granted, post-analysis is subject to human error and bias, but is still valuable, if only to validate that *something* has been found and should be studied in more detail. As an example, consider panel four again (from the left), the large top segment labelled 'post-capture' is immediately preceded by a long (minutes) sequence of alert and walking behaviours, all of which eventually lead to a chase and kill. The human labelling of this segment is good, but could be better at picking out subtle behaviours such as the sprint that leads to the

Table 4.4: Experimental model and emission priors, used for detailed analysis of hunt segment.

Model parameter	Prior
α, γ	$\Gamma(1, 5)$
α', γ'	$\Gamma(2, 1)$
H_θ	$\mathcal{N}(0, 1)$
$H_\mathcal{D}$	$\text{Pois}(\lambda)^\dagger$
μ_0	$\overline{\mathbf{Y}}$
λ_0	$D + 2$
Ψ	$C \times \text{Cov}(\mathbf{Y})$
C	$\Gamma(5, 5)$
ν	$\Gamma(1, 0.75)$

† Used as part of parametric mixture duration distribution

kill (behaviours currently folded into the 'post-capture' and 'capture' labels, the latter of which is not shown on the plot). The stateful IDHMM successfully captures this, while the IDHMM does not.

4.4.6.5 Discussion

From the middle panel of fig. 4.25, a clear trend emerges regarding the nature and behaviour of the models w.r.t. to the observations. The IDHMMs variants (models E and D in fig. 4.25) allow the practitioner to employ specific domain knowledge regarding the duration distribution of the phenomena being studied. Hence as shown, the model samples from a bespoke duration distribution, where, in this instance, we have employed a simple mixture of discrete-uniform distributions that reflect the duration content as seen in the feature space. In fig. 4.25, the light grey area preceding the five-hour mark, constitutes an area of less frequent behaviour (as labelled by the zoologists); a hunt (labelled as 'capture'), followed by a kill, followed by post-kill behaviour such as eating and drinking. It is clear that all models segment the onset of this activity sequence, but then differ in the duration properties and number of activities present in this event segment. The HDP-HMM and the sticky HDP-HMM both capture the fast switching dynamics. The other models do not, the IDHMMs do not by design, as they are primed to find *large* regions of interest, with statistical observation similarity. This points towards a scenario where both types of models are used jointly, as the strength of their sum is greater than their individual parts.

Viewed this way, we can apply the models of fig. 4.25 top to bottom. State-space models that better deal with coarse state-space granularity, and observations with non-geometric duration distributions, are labelled top to bottom in the middle panel, according to how much granularity they offer the user for this task. Conversely the IDHMM can be tuned to model bursts of activity as demonstrated in fig. 4.26. Since the IDHMMs can be tuned to segment specific duration-lengths, they are most useful for segmenting regions of the observation space where the user has some certainty of specific activities having taken place (e.g., it is reasonable to assume that a large section of the day will demonstrate very small accelerometry readings, owing to the lion sleeping).

Having ascertained where large regions of interest are located, we can turn to models A-C of the middle panel in fig. 4.25. The sticky HDP-HMM, not being as state-persistent as the stateful version, does not smooth out the activity labelling as much, but still quickly introduces new labels for surprising features. The inferred state cardinalities for the HDP-HMM, sticky HDP-HMM and the stateful HDP-HMM were 23, 18 and 20 respectively. The activity set, as labelled by the zoologists consisted of 14. That should not be taken as evidence that these models are converging to the right number. Critical analysis must still be maintained as there are many minor activities, which should be differentiated, such as 'trot' and 'walk' which, from a feature point of view are almost identical. There are also large regions where the label 'unknown' has been ascribed. Hence, by allowing the inference to sample group-specific hyperparameters, it can be seen that this has the overall effect is of favouring models not with fewer states, but models that are state-persistent with their available states.

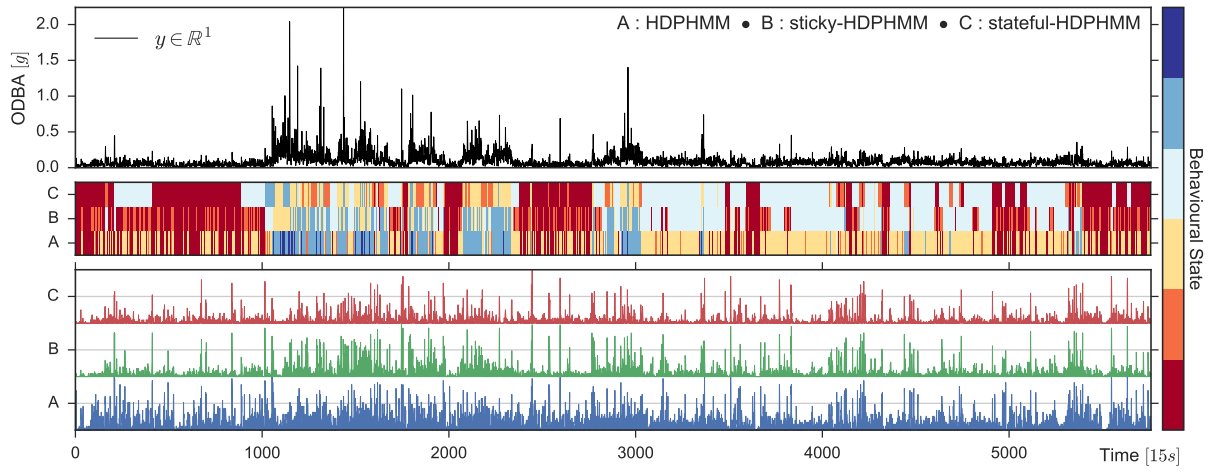
The unsupervised learning methodology demonstrated in this section, holds promise when used in conjunction with supervised methods as no prior behavioural states need to be specified, thereby allowing for the recognition of less obvious or unknown behavioural states that may be missed through limited observation. Prior classification of states used in supervised learning may be subject to confirmation bias where an observer may oversimplify a chosen state based upon their expectations and thus exclude a separate, and perhaps more subtle, behavioural class (van Wilgenburg & Elgar, 2013). Moreover, there are many ventures for further exploration from the modelling side, such as training the models in a semi-supervised fashion and then using those models, to segment other regions. We, furthermore, suggest that the methods demonstrated within can be particularly valuable

for lion behavioural ecology as the last detailed activity budget for the species was compiled more than four decades ago by (Schaller, 1974), where unobservable behaviours may not have been recognised. This method also has cross species potential, making it applicable to the wider sphere of animal behavioural ecology.

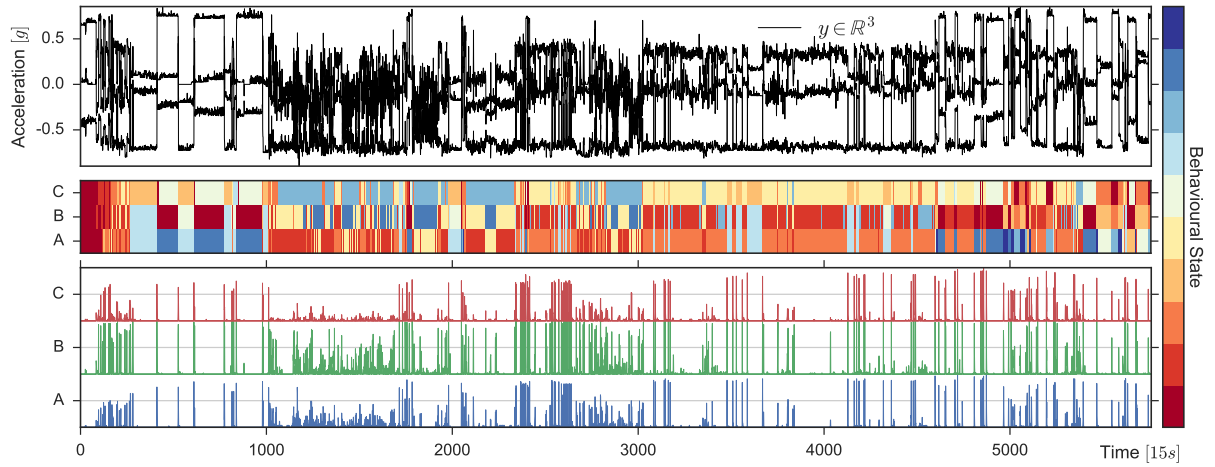
4.4.6.6 Lion modelling with Bayesian optimisation

The same methodology was used as in §4.4.2; interleaving sampling with BO, using again $\log \mathbb{P}(y_{1:T})$ as the objective function. The first set of results are shown in fig. 4.27a. Beginning with the middle panel, it is clear that the stateful HDP-HMM favours a state-sequence with fewer transitions, as is evident from the bottom panel in the same figure. The stateful HDP-HMM consistently infers a lower probability of switching state than the other two models. Moreover, it better segments similar sections of the overall dynamic body acceleration (ODBA) signal into correct assignments. Where the original HDP-HMM and the sticky HDP-HMM are more prone to introduce state switching behaviour. The HDP-HMM, without a self–transition bias, rapidly transitions among states. Lions are known to spend most of their day resting, hence segments of continued state-persistence are expected. Moreover the posterior state cardinality is shown in fig. 4.28.

As the name suggests, ODBA is good for determining, the overall state of the lion. However, to gain greater insight into the lion's behaviour ecology as part of the more global state found using ODBA, we consider the individual mean accelerations in fig. 4.27b. Perceptively this is a far harder feature set to segment. The models are broadly in agreement w.r.t. to the switching dynamics of the observations, as the bottom panel of figure fig. 4.27b shows. Indeed, for this feature set the HDP-HMM is introducing less state switches than the sticky HDP-HMM, but still more than the stateful version. W.r.t. to the state cardinality in this case too the original and the stateful models favour fewer states.



(a) Results using overall dynamic body acceleration as the feature set.



(b) Results using tri-axial accelerometry as the feature set.

Figure 4.27: Top panel: feature set over which inference and BO was performed. **Middle panel**: inferred state trajectories, with the highest log-marginal likelihood $\log \mathbb{P}(y_{1:T})$ using models A, B and C, corresponding to the HDP-HMM, sticky HDP-HMM and the stateful HDP-HMM respectively. The **bottom panel** shows the expected transition probability $\mathbb{E}[x_t \neq x_{t-1}]$ for all three models, calculated using all samples from the respective model.

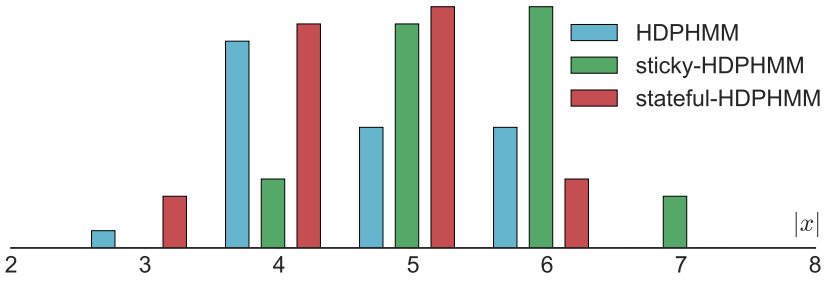


Figure 4.28: Posterior estimate of the latent state cardinality for all models.

CHAPTER 5

Kriging for prosthesis control

Contents

The final body of work which we present, pertains to the use of Bayesian nonparametric methods in the realm of prostheses control. More specifically we are interested in *change*; change in the driving functions of human locomotion behaviour in response to endogenous and exogenous influences – see fig. 5.1 for a high-level understanding of the influences involved.

Figure 5.1 succinctly summarises our approach in this thesis. In chapter 3, we investigate

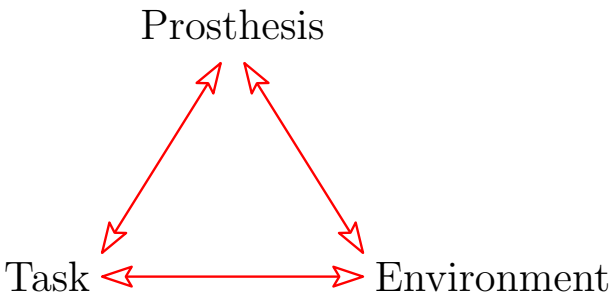


Figure 5.1: The interaction triangle between the device (prosthesis), its operating environment and the task it has been set to perform. Together they form the operating interaction triangle.

basic incidents or *tasks* that in some way our form need to be extracted from a set of observations. Those observations originate in the operational environment of the prosthesis, and can be measured by way of time-series observations of the environment (which includes the dynamics of the prosthesis as well). Performing inference on those observations, to extract the underlying behaviour inherent from user-environment interaction, was investigated in chapter 4. This chapter will investigate how we can regress over similar incidents or tasks. We take 'similar' to mean various forms of bipedal locomotion; walking, jogging or running, for example.

⋆⋆⋆

The study of powered prostheses sits within the larger emerging domain of rehabilitation robotics, where automation assistive machines (AAM), such as powered wheelchairs, neural prosthetics and exoskeletons, play an ever increasing role in re-establishing locomotion in people who have lost it due to disease, accident or war (where, e.g. neural prostheses are widely used for veterans of armed conflict) (Argall, 2013; Morimoto et al., 2012; Cheng et al., 2013; Aertbeliën & De Schutter, 2014). Beyond the replacement of limbs there are also many neurological and orthopaedic disorders (e.g. Multiple-Sclerosis, Stroke, Guillan-Barre Syndrome and Cerebral Palsy), which also reduce or eliminate voluntary recruitment of muscles. Such loss diminishes or renders impossible the performance of motor tasks or maintenance of muscles, connective tissue, and metabolic systems, that depend on muscle activity for their function and integrity.

We seek to ease voluntary muscle recruitment by also sharing control of AAMs with the motor-impaired individual and an auxiliary system – see fig. 5.2 for an illustration of this. Where we augment current control approaches with that of a learned model (see fig. 5.3 for deeper understanding about what sort of data we are considering) acting in

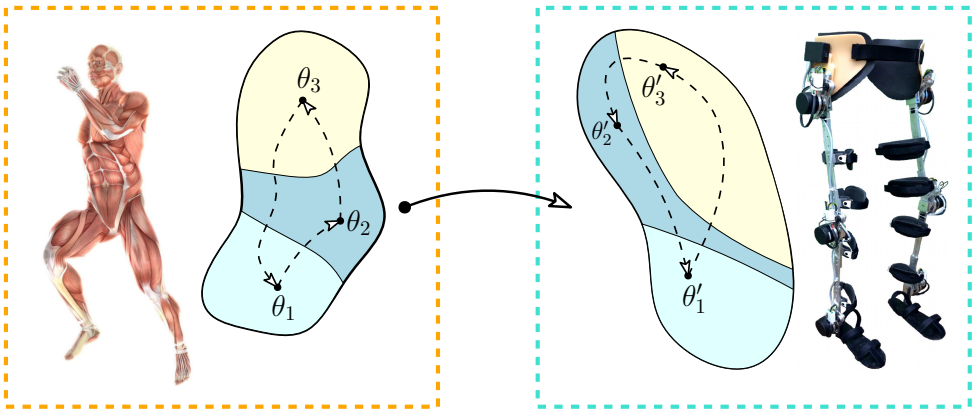


Figure 5.2: A simplified illustration of a human locomotion control manifold on the left, and its correspondent manifold, for the AAM on the right. A control path $\theta_{1\to3} \triangleq \theta_1 \to \theta_2 \to \theta_3$ is shown which corresponds to some sequence of activities (e.g. running $\to$ walking $\to$ jogging) where each parameter set θ lives on some sector of the activity-manifold (indicated by its colour). The task at hands seeks to transfer the same control behaviour to the AAM, so that $\theta'_1 \to \theta'_2 \to \theta'_3$ as closely as possible gives rise to the same kinematic behaviour as $\theta_{1\to3}$.

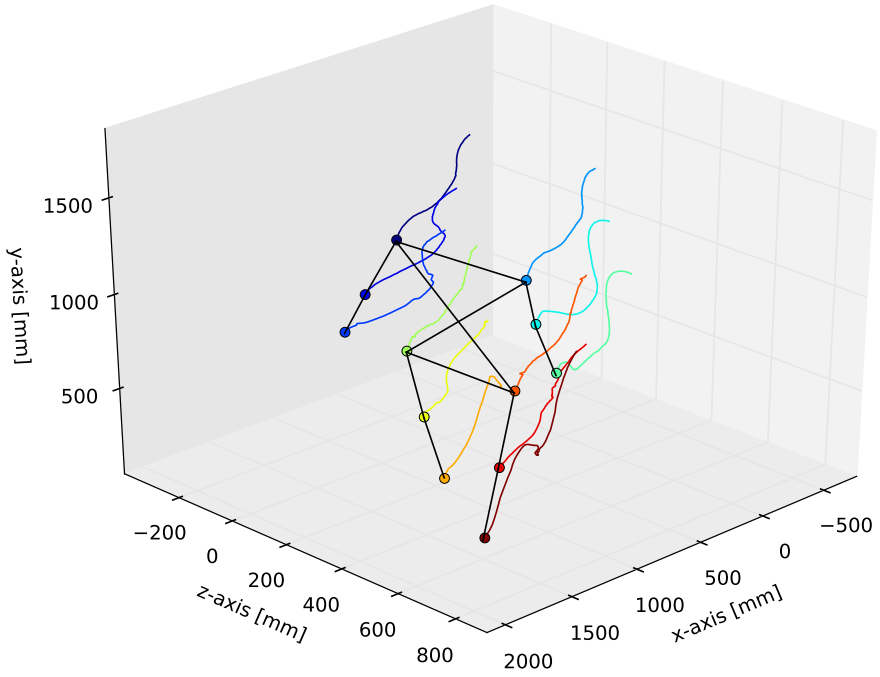


Figure 5.3: Motion capture (MOCAP) observation can be used to learn the temporal evolution of human dynamics. Shown is a subject performing a walking motion, in which the exogenous marker coordinates have been plotted over time as the activity evolves. Generated from observations used in (Dhir & Wood, 2014).

parallel or in some cases; instead of the individual (i.e. a trade-off between complete AAM control autonomy or control of the AAM shared with the user), which would relieve her of significant control load, while still being in overall command of her locomotion. Within the clinical context of AAMs, machine learning (and in particular Bayesian nonparametrics)

currently plays a limited, scarcely visible, role (Argall, 2013). Though AAMs are crucial in facilitating the independence of those with severe motor impairments, patients exist for whom the control of these devices remains an insurmountable hurdle.

We have drawn upon one domain of mathematics and statistics in particular, in respect to tools which we perceive will become useful (computational complexity allowing) in future AAM control systems: Bayesian nonparametrics. The quantitative reasons for this will become clear as the chapter progresses, but we offer a brief discussion as to the qualitative notion of using BNP as part of the AAM control echo system. Nonparametric models constitute an approach to model selection and adaptation, where the sizes of models are allowed to grow with data size. Comparatively, parametric models only use a fixed number of parameters. *What if our parametrisation is not enough?*

Any model selection will always be subject to model structure constraints, which is to say that even if we found the most optimal model $\mathcal{M}^*$ with an inappropriate structure, it will *never* be able to accurately model the phenomena under study. Concretely, consider e.g. that we want to model some second-order Markov process, but we are careless in our model structure selection and only specify a model with first-order Markovian dynamics. Then, irrespective of how good our inference is, we *cannot* capture those second-order dynamics with our models. And neither could a BNP model. But compared to a parametric model, under an appropriate model structure, the BNP model would be able to adapt the parameter space as more information is passed to the model, and thus appropriately grow (and shrink if so necessary). This a parametric model cannot do, so if some complex set of samples were received in the future, it would not be able to capture those dynamics faithfully. Therein lies our justification for supposing that BNP has a role to play in AAM control design.

Any subject fortunate enough to be fitted with an active prosthesis or orthosis, will effectively be a giant sensor, continuously passing new information to the control system. We suggest that it would be a poor design indeed, that does not take advantage of this rich temporal and high-dimensional information, in a way that allows the AAM not just to adapt, but to adapt to the individual user, and so offer truly personalised rehabilitation.

This chapter therefore explores the interface between machine learning, biomedical engineering and rehabilitation robotics, with the hope of enabling prostheses the freedom to automatically move and *feel* like a real leg for the user. Our interest in this domain is formulated through the following problem statement, which we qualify with a method motivation followed by our proposed solutions.

Problem statement Making a prosthesis adapt to the exogenous and endogenous influences, is currently an open problem.

1. Using Bayesian nonparametric methods, can we design a control system that is able to adapt to each subject, such that the AAM mimics the dynamics of each individual user?
2. Whilst there is no shortage of effective control schemes for AAMs, few rely on BNP methods, and fewer still make use of them for velocity transition – is it possible to employ Gaussian processes regression to find velocity transition vector fields?
3. Currently optimal control methods provide the great majority of control implementations for active prostheses, is it possible to combine impedance controllers with BNP methods, to yield a robust alternative?

Motivation We combine Gaussian process regression and impedance control, to elicit robust, anthropomorphic, adaptive control of a powered ankle prosthesis. We learn the nonlinear manifolds which guide how locomotion variables temporally evolve, and regress that surface over a velocity range to create a manifold. There are many reasons for this approach but some are that GPs give us:

1. The posterior predictive distribution can be found in closed form when we use a Gaussian likelihood function[1]. This enables us to place uncertainty bounds on the predictions themselves – a particularly useful property in healthcare applications.
2. We can incorporate our modelling assumptions into the kernel design and the mean function. A second possibility of prior information incorporation is to change the input/output data (which we use to train the GP) in which the behaviour of the unknown system is contained in explicit form. We could pass derivatives for example.

[1]Closed form solutions do not exist when use other likelihood functions, and we must then resort to approximate inference methods such as variational inference (Rasmussen & Williams, 2006).

3. Gaussian processes allow us to balance the capacity and suitability of the model. This is to say that, by limiting the model capacity we can prevent overfitting (e.g. by specifying a suitable kernel) (MacKay, 2003). This property flows from the marginal likelihood which enables one to compare models.

Contributions The joint set of manifolds, as well as the temporal evolution of the gait-cycle duration is what we term a *locomotion envelope* (Dhir et al., 2018, 2017a). This control construction is our primary contribution, it is broken down as follows:

1. We are able to combine GP regression with impedance control, thereby rendering a system capable of smoothly transitioning between different locomotion velocities.
2. By using BNP methods, we show that the system is able to adapt to the individual gait cycles, of each subject used for the experiment.
3. We show that by using impedance control, we demonstrate a robust method for trajectory reference matching (where the test trajectories are generated from GP regression).

In sum, locomotion envelopes combined with impedance control provides a powerful method for adaptive control, as well as a mechanism for smoothly transitioning between self-selected velocities.

⋆⋆⋆

The material presented within is conditional on prior work, which can be found in the preliminary material. The following subsections have relevant dependencies:

➢ Though we give a robust treatment of Gaussian processes regression in this chapter, section §5.4 is well informed by §2.7.1.

5.1 Powered prostheses

A prosthesis is any device that replaces the biomechanical functionality, of a healthy limb. Though they have traditionally been "energetically passive" tools (Lawson et al., 2014) (meaning that they are endowed with the ability to store energy, but cannot produce it), this is an unsatisfactory solution to a problem which requires a like-for-like replacement, i.e. a device capable of *generating* power – like our own muscles and limbs. Conversely,

passive devices cannot produce the required joint-torque required to replicate that of a healthy limb.

The joint of a healthy limb can produce a large and varied range of mechanical behaviours (Lawson et al., 2014), which, as noted by Lawson et al. (2014) "are, in general, characterised by power dissipation, storage and generation". Consequently, by replicating these functionalities in an artificial limb (i.e. a prosthesis), it is reasonable to propose that a better solution to the aforementioned problem can be found. That being said, consider further that users of passive prostheses, in general:

- walk slower (Genin et al., 2008);
- use significantly more energy whilst developing locomotion (Genin et al., 2008; Waters et al., 1976);
- are more limited in the terrain, activities and combinations thereof, that they can reasonably traverse and undertake respectively (Vrieling et al., 2008);
- fall more frequently when compared to healthy subjects (Miller et al., 2002; Lawson et al., 2014).

Hence, purely from a mobility and stability perspective, it is clear that current prostheses demand some power-generating capability. However, it would be unfair to say that passive prostheses have no advantages, indeed there are plenty.

First and foremost, and perhaps the most obvious, being 'passive' means that a whole host of systems do not need to be accounted for such as actuators, controllers for the actuators, the extra weight that comes with having power-generating capability (e.g. the typically high-torque servo motors), nor the complex human-robot interface which enables the prosthesis to take appropriate action on command from the human user. Indeed, mechanically passive prostheses, are still the most popular means by which to restore some locomotion to users (though this has more to do with cost than functionality).

Moving on to the powered domain, work on powered prostheses has been forthcoming for several decades now (see the early work by Heger et al. (1985) and the more recent, and very thorough, thesis by Grimmer (2015)), and an impressive body of work exists on the topic. Our work however, is not concerned with *how* power is generated in the device, but

rather how it is controlled in manner befitting the complexities and large inter-subject variability of users (no two humans have the same biomechanics) – see fig. 5.5.

5.2 Powered prosthesis control

Like the powered prosthesis work, there also exists a significant body of work on their control – see e.g. the excellent review papers by Tucker et al. (2015) (fig. 1, of that paper, gives an excellent synopsis) and Ferreira et al. (2016). We are not particularly interested in these controllers per se, but rather in an aspect and functionality which is missing from the great majority of these protheses, namely *velocity adaptation* and *regulation*.

To understand how velocity adaptation and regulation can be helpful in regaining mobility, it is worthwhile considering how healthy lower-body kinetics and kinematics work. For this, we shall make ample reference to one full gait-cycle, shown in fig. 5.4. To put it simply, normal and healthy human variable velocity locomotion, necessitates a change of joint torque at the body joints, as overall velocity is increased or decreased (Winter, 1983). For example, the overall joint torque load is different when walking, compared with running. This is rather obvious, but for the sake of completeness we will continue the discussion in full.

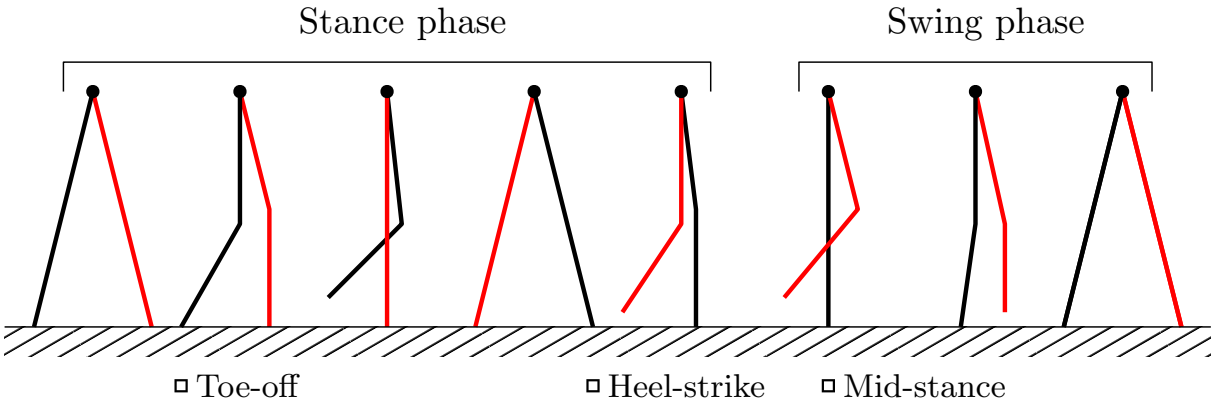


Figure 5.4: The human bipedal gait cycle consists of a stance phase and a swing phase, and is the sequence of movements during locomotion, in between when one foots touches the ground, and when that foot again touches the ground. The sequence of movements propels the centre of gravity forwards. Depicted is a simplified version, where only the legs are depicted but not the feet or the upper body. The right leg is shown in red. Nominal gait-cycle events are emphasised with squared boxes and their place in the cycle.

Following the excellent discussion by Lenzi et al. (2014b), a brief synopsis of the torque profile during walking can be stated. During the stance phase of walking, the torque

profile follows a highly nonlinear trajectory (see fig. 5.3 for raw MOCAP observations that demonstrate his phenomenon). This is commensurate with the complex loading profile of the actuators (muscles) (fig. 3 of Thelen & Anderson (2006)'s work demonstrates this very well), as well the involved actuator dynamics required to produce said profile (i.e. multiple muscles firing at different times, with different intensities etc.). This is just whilst walking on flat and straight surfaces. Where the overall 'goal' is to support the body weight against gravity, as well as to propel it forward in the desired direction (Lenzi et al., 2014b). Continuing with the swing phase; Lenzi et al. (2014b) note that a "progressively faster movement" is required at increased velocities to ensure that the placement of the foot, during swing, is well and timely placed to prepare for the forthcoming heel strike (see fig. 5.4). For an example of this profile see fig. 5.5 and fig. 5.6. Consequently, as the transition between profiles is highly nonlinear, this behaviour needs to be replicated by current controllers. That is a tall order to say the least (we will discuss details of state-of-the-art methods in §5.3), but also a good justification for why controllers should be adaptive – something which they, generally, are not.

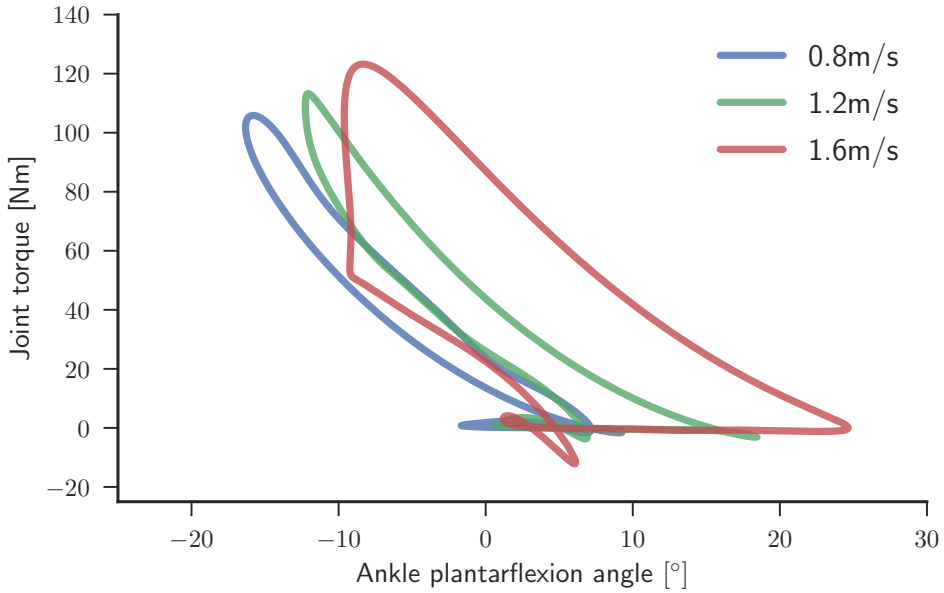


Figure 5.5: Ankle-angle torque profile for a male subject from the Moore dataset (Moore et al., 2015), for various velocities. The profiles noticeably shift with an increase or decrease in velocity.

⋆⋆⋆

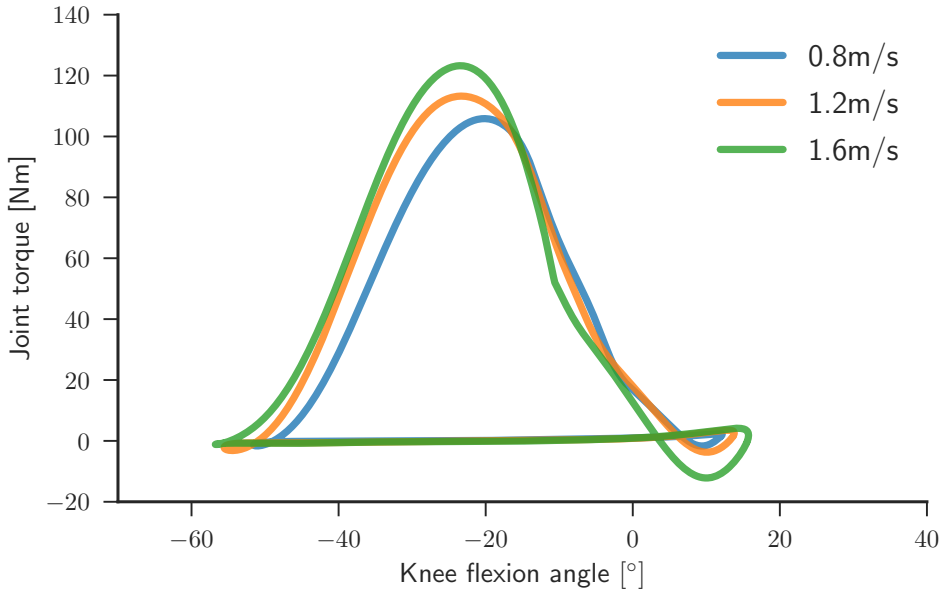


Figure 5.6: Knee-angle torque profile for a male subject from the Moore dataset (Moore et al., 2015), for various velocities. The profiles noticeably shift with an increase or decrease in velocity. We do not use an actuated knee-joint throughout this study, but we do include it for reference, and to demonstrate how multivariate GP regression behaves on other joint, beyond the primary power-injecting degree of freedom: the joint angle.

This chapter will discuss a control strategy for a powered ankle prosthesis for transfemoral (above the knee) and transtibial (below the knee) amputees. We will also be discussing other further applications where this body of work could be used. Indeed, as noted by Song & Geyer (2015):

> Human locomotion control models have the potential to elicit new controllers for legged robots and to provide simulation platforms for testing walking assistive devices.

First, we begin with a review of current state-of-the-art models, methods and controllers for velocity regulation *and* adaptation, in powered ankle-prostheses.

5.3 Related work

Lots of people could benefit from a powered ankle. To this end, active prostheses[2] produce positive work during walking and inject energy to the gait during push off and propulsion. Powered ankle-foot prostheses can accommodate fast walking beyond the capability of their passive counterparts, and recent innovations and advancements have led to the

[2]These results are applicable to orthoses as well, but that is outside the scope of this thesis.

development of various devices for physical assistance and human locomotion restoration (Tucker et al., 2015). The design of a prosthesis is twofold; one part is mechanical, the second is the control of the mechanical. In this review we focus on the latter.

Curiosity about control schemas date back several decades. Indeed, early work on powered transfemoral (above-knee) prostheses was undertaken by Grimes et al. (1977); Flowers & Mann (1977); Stein & Flowers (1987). But, as noted earlier, control of these devices have two sources of origin (broadly speaking); those that explicitly seek to control prostheses for human ambulation, and those that seek a biomimetic solution for bipedal robots. Virtues and vices can be found in both domains, and we shall discuss both in turn, starting with the former. For emphasis, recall that we are *only* interested in strategies that relate to velocity adaptation and regulation, *not* the general field of prosthetics control strategies as reviewed by Ferreira et al. (2016) and Tucker et al. (2015). Indeed, as noted by Lenzi et al. (2014b) "available controllers for powered transfemoral prostheses cannot generalise across different walking speeds".

The most common solution used for control of active prostheses is to match the torque-angle profile of a healthy human ankle at the powered prosthetic joint (Hitt et al., 2010; Sup et al., 2008, 2011). Sup et al. (2009) describe a finite-state impedance control approach to control a prosthesis during walking and standing. This control method often leads to a few fixed walking speeds but lacks the adaptability required to seamlessly change the velocity or step size following the user's intention. In addition, gathering human experimental data at different speeds is limited to a few discrete values and it is difficult to perform numerous experiments to derive walking data at intermediate walking speeds or step sizes (Liu et al., 2008; Arnold et al., 2013; Moore et al., 2015).

In order to address the adaptation problem of this control method Au et al. (2008) proposed a reflexive neuromuscular model with positive force feedback. They showed that the proposed method can adapt to changes in the walking speed and floor inclination. Further, Markowitz et al. (2011) proposed a neuromuscular reflexive model with speed adaptation for a powered ankle-foot prosthesis and tested it at three walking speeds (0.75m/s, 1.0m/s, and 1.25m/s). Hargrove et al. (2013) extracted electromyography (EMG) signals from a patient, who had undergone targeted muscle reinnervation surgery, and then used these to provide a robust control mechanism for an ankle and knee prosthesis to walk with a fixed speed on level ground, stairs, and ramps with a 10° slope. Lenzi et al. (2014a) demonstrate

a control approach for a transfemoral prosthesis, which regulates the ankle and knee joint torque by estimating the walking phase and speed. A two-dimensional lookup table with a low-pass filter was used to encode torque-angle curves for two walking speeds. Quintero et al. (2016) use a proportional-derivative controller with virtual constraints, which used a human-inspired phase variable to adapt to speed variations. The authors themselves note that their method could not quite compensate for strong nonlinearities in the ankle dorsiflexion and consequently produced control errors when this happened.

Now, Ferreira et al. (2016) suggests that active powered prosthesis control falls into four categories. Including echo control, finite state impedance control, EMG based control and central pattern generator (CPG) based control. Conversely, Lenzi et al. (2014b) suggest that effective speed adaptation has been successful using two approaches. The first one, by Herr & Grabowski (2012), proposes a method that mimics human muscle reflexes. This allows the prosthesis to adopt velocity adaptation by virtue of changing the torque output without actually "measuring the walking or cadence" (Lenzi et al., 2014b). Though demonstrating very impressive results, there are some drawbacks. Specifically, because their guiding metric is the metabolic cost of transport, for five different velocities, across which they regress, it is difficult to ascertain how well their control schema transitions between velocities. Secondly, using the same taxonomy (Lenzi et al., 2014b), concerns the usage of pre-programmed (alternatively pre-specified) ankle-torque profiles. In order to cope with velocity variations Holgate et al. (2008) modulated the ankle trajectory in time and amplitude, allowing their subject to transition between velocities. But this is a parametric method, relying on look-up tables for fitting without uncertainty bounds. Finally, Lenzi et al. (2014b) propose their own method which imitates the basic velocity adaptation mechanism used by healthy (i.e. intact) legs. They employ quasistiffness profiles (we also show diagrams of these in our result section) of an intact leg, which they directly encode into their controller, and then interpolate between them based on their intention estimation. Theirs is most likely the work that comes closes to ours, but whilst they use a PD controller, we use an impedance controller, and probabilistic interpolation and extrapolation, which also gives us uncertainty bounds on our predictions. Further reviews of control strategies for lower extremity prostheses can be found in (Jiménez-Fabián & Verlinden, 2012).

Like us, GPs have been used before for similar purposes. Hong et al. (2015) use the GP dynamics model (GPDM) (Wang, 2005). The GPDM is a dimensionality reduction method which comprises a low-dimensional latent space with associated dynamics, and a map from the latent space to the observation space, it is an elegant model of dynamics that accounts for uncertainty in the model. Hong et al. (2015) use it to create a low-dimensional representation of walking motion, extracted from 50 subjects. They do this for three different speeds. This is not a control scheme, but a rehabilitation method, which can generalise between the training speeds (however their reconstruction errors are high >10%) and is used for gait training of subjects with hemiplegia. Lizotte et al. (2007) instead make use of standard GP regression (GPR) to optimise gait for quadruped and biped robots. Though they do not deal specifically with velocity adaptation and regulation (rather environmentl adaptation) their ideas are relevant to our discussion, as environmental adaptation is the next logical step for our method. Similarly Yun et al. (2014) used GPR to generate a model for gait pattern prediction for one speed (3km/h). Though different to what we propose, it does suggest a validation of our method, since the paradigm remains the same (and they demonstrate an impressive array of results); theirs is a prediction in space, ours in time.

A basic requirement of any human locomotion controller, is the ability to change speed and gait. We have shown some example studies where speed variation control is studied, for active prostheses. For completeness we also review recent advances within the field of humanoid bipedal robotics, whose aim and scope is similar to ours, and where, potentially, our methods are also applicable. These methods too have the potential to produce new controllers for bipedal robots and to provide simulation platforms for lower-limb prostheses (Song & Geyer, 2015). Hence, their inclusion is relevant to our contribution.

CPGs have been successful in human locomotion controllers (Ijspeert, 2008), where they are responsible for producing the basic muscle activation rhythms and local reflexes that modulate the muscle activations. CPGs can also be used to adapt to the environment. This locomotion paradigm has been successfully applied to prosthetics, see e.g. (Thatte & Geyer, 2014; Eilenberg et al., 2010) (however, neither is capable of modulating speed). CPGs and reflex-only models (Geyer & Herr, 2010) have in common that they generate locomotion, and switch locomotion regimes and behaviours by transitioning between different sets of control parameters (Song & Geyer, 2015). But as Song & Geyer (2015) note, it is

unconvincing "that humans store look-up tables of hundreds or thousands of low-level control parameters for all different environments and behaviours". Some CPG-based studies have overcome this by using high-level policies that modulate low-level control parameters.

For example, Van der Noot et al. (2015) consider a similar problem as this contribution. They demonstrate a energy-efficient neuromuscular model, which mimics human walking. They do this by combining reflexes and a CPG able to generate gaits across a large range of speeds. They demonstrate their approach on a simulation model of the 95cm tall COmpliant HuMANoid platform (COMAN) robot. Their results show that they were able to simulate energy-efficient gaits ranging from 0.4m/s to 0.9m/s. One drawback of their method is that it requires optimisation of the open parameters of their model, and what is more: for a fixed-time simulation (they use 60s). Further, the most common control method used in humanoid robots is based on the notion of zero moment point (ZMP). In ZMP walking the feet are kept flat on the ground while the knees are bent – this is not suitable for human walking (Vukobratović & Borovac, 2004). Further, related to this discussion, is dynamic walking:

> A theoretical approach to legged locomotion which emphasizes the use of simple dynamical models and focuses on behavior over the course of many steps, rather than within a single step, typically in an attempt to understand or promote stability and energy economy.

As Collins (2008) goes on to explain that: dynamic walking builds on the *passive* dynamic approach by adding simple forms of actuation and control (Collins, 2008, §1.2.2). Here, when we refer to *passive*, we refer to a form of locomotion which is designed such that the natural oscillation of the system (i.e. the robot) results in a gait – where the analogy is drawn to the pendulum wherein the natural oscillation of that system is expressed through the swinging of its pendulum. Fully dynamic walking (i.e. not passive) adds actuation and control to the system without overwhelming the natural dynamics of the design. In the pendulum analogy this could be e.g. adding a small amount of torque at the apex of the pendulum, to increase the angular velocity, and then allowing gravity to pull it down again, once it has reached its peak (and then continuing to add small amounts of energy to the system to maintain this new state). Dynamic walking remains an elegant approach to biped control and research. Consider for example the recent work

by Gritli et al. (2015) where the authors aim to control the chaotic dynamics, exhibited in the semi-passive dynamic walking of a torso-driven biped robot as it goes down an inclined surface. Though elegant and important in applications were energy efficiency is paramount, it does require a gravitational differential to move down an inclined surface – for *passive* dynamic systems. However, when such requirements are relaxed, some studies explicitly use the knowledge obtained in passive dynamic walking for control of actuatd dynamic walking (Iida & Tedrake, 2010), with the overall aim of promoting energy efficient control.

⋆ ⋆ ⋆

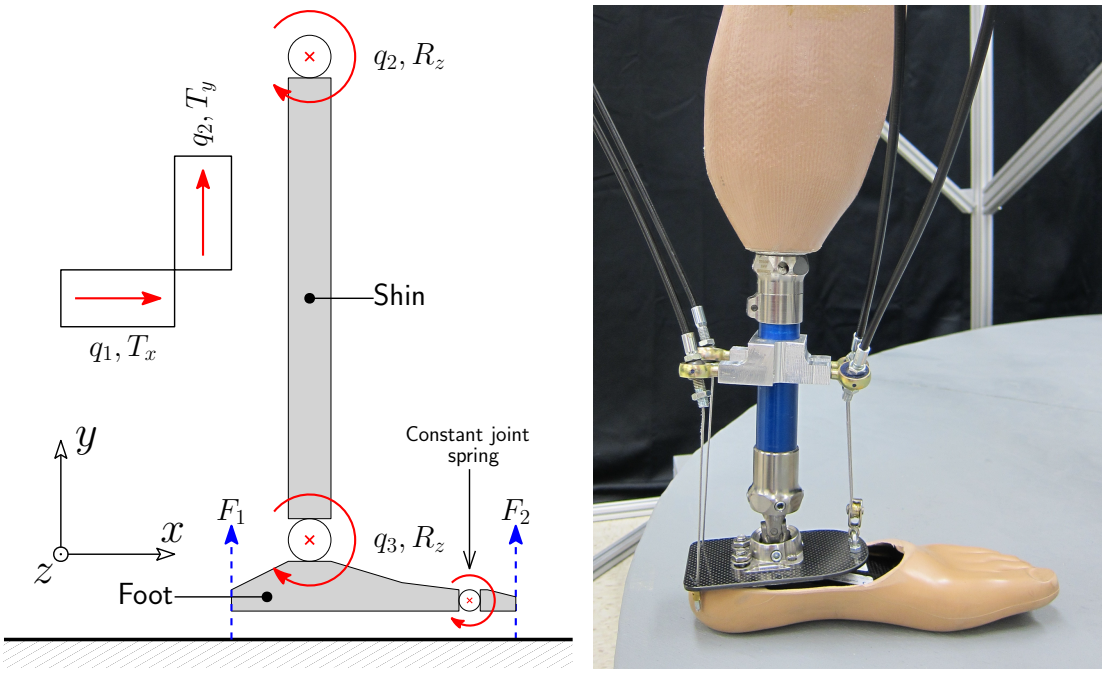
In this study, we introduce an adaptive walking control method for a simulated powered ankle-foot prosthesis (see fig. 5.7) using universal function approximation methods, combined with impedance control. The adaptive property helps the user to change the walking speed, and the nonparametric property enables the control scheme a higher level of precision as more observation are added. The proposed approach is general and can be applied to any ankle-foot prosthesis provided certain sensory information is available. Our contributions are threefold:

- We use Gaussian process regression to learn a multivariate function describing the power required at the ankle joint as a function of speed and position in the gait-cycle.
- We combine impedance control with the multivariate functions into what we term a *locomotion envelope*.
- We demonstrate our methods both on simulation and hardware experiments.

Initially we consider the sagittal degree of freedom (DoF). The shin is modelled with a rigid body connected to a planar floating base to simulate walking using recorded human walking data. The ankle-foot prosthesis and its interactions with the environment are modelled in the Robotran simulator (Fisette & Samin, 1993) as shown in §5.6.

5.4 Locomotion envelopes

We present a novel control strategy for powered ankle-foot prostheses, using a data-driven approach, which employs a combination of Gaussian processes (GP) regression and impedance control. We learn the non-linear functions, which dictate how locomotion



(a) Prosthesis free body diagram **(b)** Prosthesis

Figure 5.7: Diagrams of prosthesis used in this chapter. In fig. 5.7a a schematic diagram is shown of the ankle-foot prosthesis with five degrees of freedom. In fig. 5.7a T_x and T_y are translations along the x and the y axis. R_z is rotation about the z axis. F_1 and F_2 are ground forces. In fig. 5.7b the physical counterpart is shown, originally devised, outlined and analysed in (Ficanha et al., 2016). Adapted from Dhir et al. (2018).

variables temporally evolve using the aforementioned nonparametric method, and regress that surface over several speeds to create a manifold, per variable. The joint set of manifolds, as well as the temporal evolution of the gait-cycle duration, is what we term a *locomotion envelope*.

Current powered prostheses generalise poorly across speeds. Others that do have the capacity to exhibit several speeds, typically divide the gait cycle into several sequential periods. Each has independent controllers, resulting in many control parameters and switching rules that must be tuned for a specific walking speed and subject. It is not convincing that control strategies should rely on stored look-up tables of hundreds or thousands of low-level control parameters for different speeds. It is also unlikely that humans rely upon this approach, where a hierarchical control structure is likelier (Wolpert & Ghahramani, 2000; Dounskaia, 2010).

A more compelling strategy is adaptable, and becomes more robust and accurate with more data and provides a nonparametric approach, which allows the strategy to grow with

the number of observations. We introduce such a strategy in this chapter and successfully simulate locomotion beyond our training data.

⋆⋆⋆

The term *locomotion envelope*[3] refers to the joint set of multivariate regression surfaces which, appropriately applied, confers upon the user the ability to synthesise natural and robust bipedal locomotion (that same envelope also contains learned temporal regression functions, controlling the evolution of stride duration, across a sought range of gait speeds). We seek to regress physical properties such as joint angles, ground reaction force (GRF) and moments (GRM), from very sparse observational data. This can be viewed as a learning problem, where we are interested in learning the multivariate regression manifolds of the aforementioned properties, as well as others.

There are a multitude of options, which one could use to construct these envelopes, we posit however that the most suitable is Gaussian process regression. There are a number of reasons for this, some of which we expand upon here:

- First, given observations and a kernel, the posterior predictive distribution can be found exactly in closed form (Rasmussen & Williams, 2006).
- Second, by nature of its construction, expressivity is considerable, allowing us to incorporate a host of modelling assumptions and domain knowledge.
- Finally, as noted by Rasmussen & Williams (2006); given a fixed kernel, the GP posterior allows us to integrate exactly over competing models, hence overfitting is less of an issue than in other candidate methods.

For a more detailed discussion see (Duvenaud, 2014, §1.1.3) and (Rasmussen & Williams, 2006). We will not dwell further on the Gaussian process beyond reminding the reader, for completeness, of the fundamentals and otherwise we refer to §2.7.1 for further details.

A Gaussian process is a method for universal function approximation i.e. some realisation of a GP, with some kernel, is arbitrarily close to the function under study, to within some norm. Thus we can approach the multivariate function learning problem by placing a

[3] We derive the term from aerodynamics in which the flight envelope, service envelope or performance envelope of an aircraft refers to the capabilities of a design in terms of airspeed and load factor or altitude. The author will submit an aviation bias here: he is an aeronautical engineering by training.

prior distribution on the regression function using a GP (Rasmussen & Williams, 2006). With a GP we can define a distribution over functions

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{5.1}$$

parametrised in terms of a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. A GP is fully specified by these two functions. For a much deeper treatment of the GP, refer to the GP handbook by Rasmussen & Williams (2006).

5.4.1 Noise modelling

One of the largest assumptions in GP modelling is that of the inherent noise in the model. *Why?* Because standard GPR assumes that input locations are noise free (Rasmussen & Williams, 2006). The outputs however follow a homoscedastic (this literally means 'the same variance') noise process. Lets recall precisely the model using the standard linear model as a reference:

$$f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{x} \quad \text{s.t.} \quad y = f(\mathbf{x}) + \epsilon. \tag{5.2}$$

Above, $\mathbf{w}$ weight parameters of the linear model and y is again our observed target value and $\mathbf{x}$ our input vector. With GPs then, we assume a homoscedastic, additive, noise model, also seen in table 5.1, specifically:

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{5.3}$$

in which σ_n^2 is the variance of our noise, and is assumed the same for all dimensions of $\mathbf{x}$. The nature of this assumption cannot be understated, as Rasmussen & Williams (2006) explain: this noise assumption, together with the model, explicitly gives rise to the likelihood model which we derived in the preliminary sections.

Table 5.1: Different noise models.

Model	Functional form
Additive noise	$y = f(\mathbf{x}) + \epsilon$
Linear noise	$y = a\,\mathbf{x} + b\epsilon$
Posterior non-linear noise	$y = g(f(\mathbf{x}) + \epsilon)$
Heteroscedastic noise	$y = f(\mathbf{x}) + \epsilon g(\mathbf{x})$
Functional noise	$y = f(\mathbf{x}, \epsilon)$

Under a different model, we would *not* receive this simple likelihood model. But, that is neither here nor there, rather, what is important is the other large assumption that we make with GP modelling: that $\mathbf{x}$ is uncorrupted. For many real problems this is not a realistic example. Even in our own experiments, where data was collected under rigorous test conditions, inputs are not noiseless. Indeed, there is no such thing as noiseless measurements, since we necessarily have finite precision with any available sensor. A noiseless variable is merely a theoretical construct. Going back to GPs, some interesting recent work which does consider noisy inputs as that by McHutchon & Rasmussen (2011) where the authors present a We present simple GP model for training on input corrupted by i.i.d. Gaussian noise. Consequently they consider a model of the form:

$$y = f(\mathbf{x} + \epsilon). \tag{5.4}$$

In sum, the two limiting assumptions about GP regression noise are

- measured observations are noise-free;
- output points are corrupted by constant-variance Gaussian noise.

But as said, this is not a particularly realistic state of modelling. As Rasmussen & Williams (2006) note, we do not have access to function values themselves, but only noisy versions thereof.

Thus, for the sake of simplicity, so too shall we, but in the knowing that it may lead to errors. The assumptions noted above work for many datasets, but for others, either or both of these points are invalid and can lead to poor modelling performance (McHutchon & Rasmussen, 2011). This discussion is relevant since we are proposing an anthropomorphic control system, which will interact with a human subject. Consequently it is important that noise model assumptions are discussed.

5.4.1.1 Complex noise models

In table 5.1 we gave some examples of more complex noise models, models which do not follow the assumptions imposed by the GP model. We have discussed additive noise (assumed under the GP) and also homoscedastic noise. Lets go a step further and consider hetereoscedastic noise. The difference with this noise model is that it assumes that there

is a direct dependence between the signal characteristics and the unwanted noise model (Woodward et al., 1998).

Woodward et al. (1998) explain that many instruments and processes (indeed most) have some component of hetereoscedastic noise such that there noise characteristics are dependent on signal characteristics. One can see immediately the consequences for biological control signals. First, it follows that an assumed hetereoscedastic noise model is highly personalised and *not* independent of the sensor modality. Second, imposing the assumption of this model would allow us to perform more accurate analysis, should we be able to estimate the noise signal with high accuracy, and make predictions with high accuracy (though we can no longer employ a closed-form posterior predictive distribution). Such a model is proposed by Goldberg et al. (1998) where the dependent noise model is also modelled as a GP ($g(\cdot)$ in table 5.1).

Suppose instead that the noise is *not* Gaussian, then what? Such a scenario was entertained by Snelson et al. (2004). Therein they explain that it is indeed somewhat simplistic to assume Gaussian noise, to overcome this, they 'warp' the non-Gaussian noise input space into e.g. the log-space, and then the assumption is imposed that *this* space has Gaussian noise and can then be well modelled by a GP, with its simple construction. Though this may seem like a fanciful way to get around, or approximate, a rather difficult problem, Snelson et al. (2004) explain that this is common practise in the statistics literature, so it is not without grounding. In a similar vein is the paper by Rasmussen & Ghahramani (2002) where they discuss different noise variances, in different parts of the input space.

Consider the problem: suppose an input space is given, but it is large (high-dimensional) and may have discontinuities, a stationary covariance function will not adequately capture the latent function governing this space (Rasmussen & Ghahramani, 2002), and this naturally also includes the latent noise model. To tackle this, the authors present a mixture of experts model, but where the individual experts are GPR models (each with a different kernel). This allows the "effective covariance" (Rasmussen & Ghahramani, 2002) to vary with the inputs. This also means that the noise model can vary with the input, which is why this is an important contribution for domains which could have complex noise behaviour (such as ours – though we will not cover this extension to our model in this thesis).

5.4.2 Gait cycle stride-time regression

Having discussed noise models for the Gaussian process model, we now move onto another area of investigation: stride-time. We define stride-time as the time it takes between a foot leaving the ground, and that same foot touching the ground again. Naturally, this time incident will reduce with speed, as a consequence of bipedal locomotion. But as we go on an demonstrate, stride-time and speed, are not linearly correlated, and moreover demonstrate a high-degree of variability. As an example consider the curves in fig. 5.8. Here we show all the extracted cycles of the plantarflexion angle, plotted against the time it took for the variable to complete one full cycle, before repeating itself again.

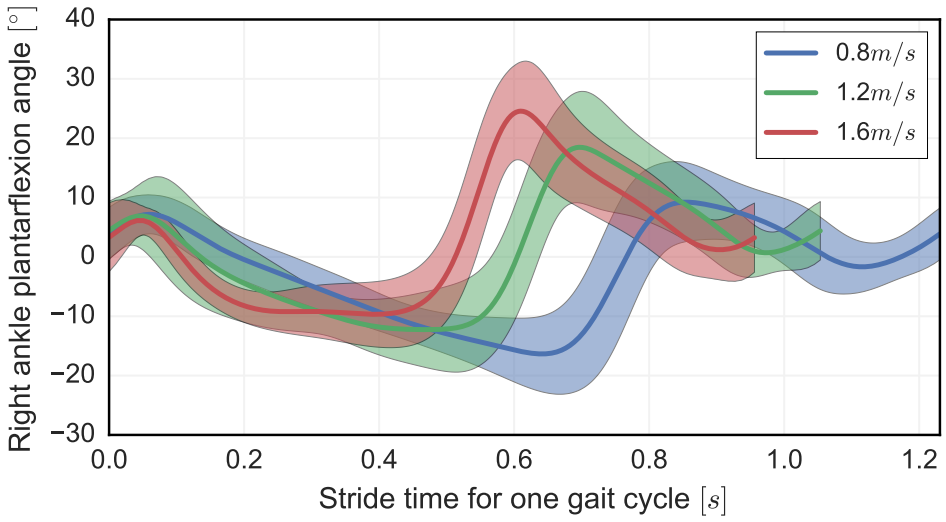
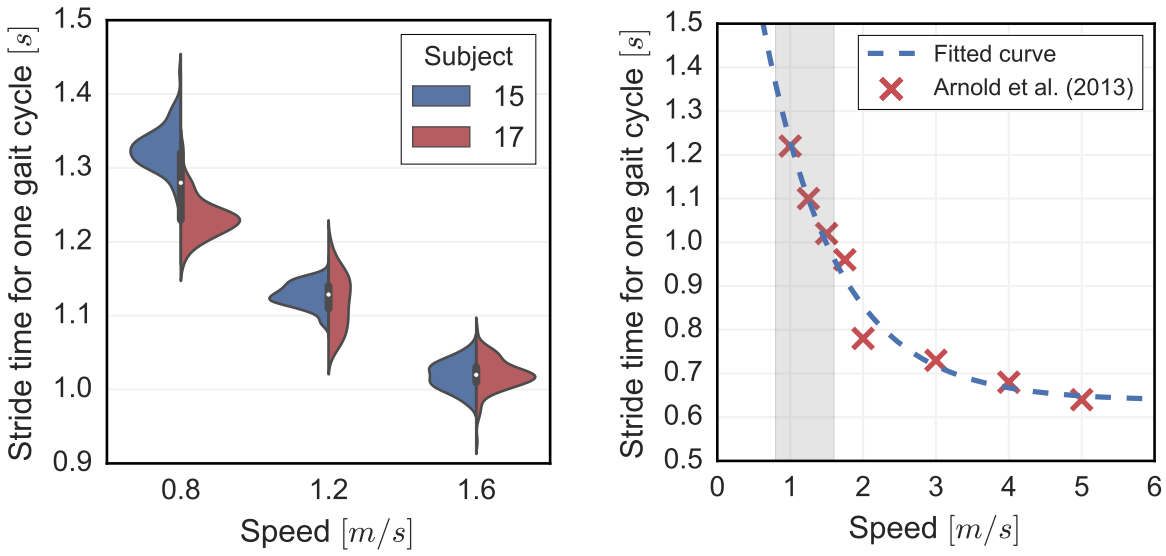


Figure 5.8: Un-normalised ankle plantarflexion angle with ± two standard deviations, for subject 6, during normal walking at three different speeds. Depiction demonstrates on the horizontal axis the stride-time required to reach various parts of the gait cycle. It is clear that just after the middle there is large acceleration in angle.

Much in the same way that we extract single-cycles from the long time-histories of observations for space variables (like the one plotted in fig. 5.8), we do the same for the time-variable (see fig. 5.9).

Because we are extracting a short quantity (stride-time) from a large one (the full time-history of a trial) we are left dealing with a lot of data. One of the primary assumptions of this work, is the notion of 'atomic' locomotion envelopes – i.e. that it is enough to find to do inference over the primary building block (one gait cycle) of locomotion, rather than the whole time-history. Hence, we adopt the same approach here: gait cycle duration is measure for each trial, then an average is taken for that trial and hence speed, and



(a) Depicting comparative stride duration violin plot.

(b) Curve fitted to Arnold et al. (2013) stride duration observations.

Figure 5.9: Left subplot depicting the comparative duration times for subject 15 and 17 from (Arnold et al., 2013). Right plot showing the logarithmic stride duration trend fitted to data from (Arnold et al., 2013). Adapted from (Dhir et al., 2018). Note that these two subjects were chosen because they showed the greatest similarity in terms of physiology as well as weight (mass). Whilst two completely different subjects could have been shown, interest lies in finding common denominators between across subjects.

then scaled according to body mass and height. For further details on this process see the primary studies from whence our data was taken (Liu et al., 2008; Arnold et al., 2013; Moore et al., 2015).

Continuing, each envelope needs to scale each manifold by time as the prosthesis accelerates or decelerates. This is a non-linear relationship, which, like the manifolds, also needs to be approximated. This, however, is a simpler task than the preceding one. To aid our exposition and comparison, we employ the study by Arnold et al. (2013), wherein motion capture data was collected for five subjects walking and running on a force-plate instrumented treadmill (for a good idea of the setup, refer to the original study Moore et al. (2015)). The subjects were all experienced long distance runners who reported running at least 30 miles per week. The subjects walked at 1.00m/s, 1.25m/s, 1.50m/s and 1.75m/s and ran at 2.0m/s, 3.0m/s , 4.0m/s and 5.0m/s.

The violin plot shown in fig. 5.9a shows the distribution of extracted stride duration, across two categorical variables, speed and time, enabling the display of the comparative distributions. Whilst it is clear that mean of the duration kernels, for both subjects,

agree for speeds 1.2m/s and 1.6m/s. They diverge at 0.8m/s. However, when compared to fig. 5.9b, both means compare well with those results, particularly when their result variance is taken into account (not shown in fig. 5.9b). *Why the divergence at 0.8m/s* It is likely to be more an artefact of the individual gait cycle – a physical manifestation of subject 15's overall slower gait at lower speeds. This is further reinforced if we consider their respective physiologies (see the full set of data by Moore et al. (2015)), they show little divergence in that regard: both weigh approximately 85kg, and are both circa 1.80m tall – as well as of similar age. Hence, their anatomies suggest that they should manifest a gait pattern of high similarity, and they do, but not in the lower speed range for reasons discussed.

The same figure also shows the speed region (in grey) investigated by Moore et al. (2015). Using a simple fitted function (inverse exponential) we are able to get a good estimate of the stride duration at our test points.

5.4.3 Analysis of human ambulation

Having established by what means we approach the nonlinear regression problem, it is pertinent to be able to understand precisely what feature we seek to regress. This calls for an understanding of human ambulation. Hence in this section, we verify common and general physical features observed in all healthy subjects as they walk faster, again drawing from the dataset produced by Moore et al. (2015) (which also limits our verification to the number of test subjects). Consider fig. 5.10 wherein we demonstrate that the power at ankle increases monotonically with an increase in speed. They yellow part of each subfigure shows when the normalised power is approaching its peak. There clear trend, for all subjects, is an increase in power injecting, with an increase in speed. The injection happens are almost exactly the same place in the gait-cycle, but the difference is its magnitude, which as can be seen; increases with speed.

The power is computed by deriving the product of ankle moment and ankle angular velocity and then normalising, which we show for three subjects. In more detail, work is given by

$$W = F \times d, \tag{5.5}$$

where F is the applied force and d is the distance moved in the direction of the force. Power, taken as a function of time, is the rate at which work is done and is expressed by

$$P(t) = \frac{W}{t}, \tag{5.6}$$

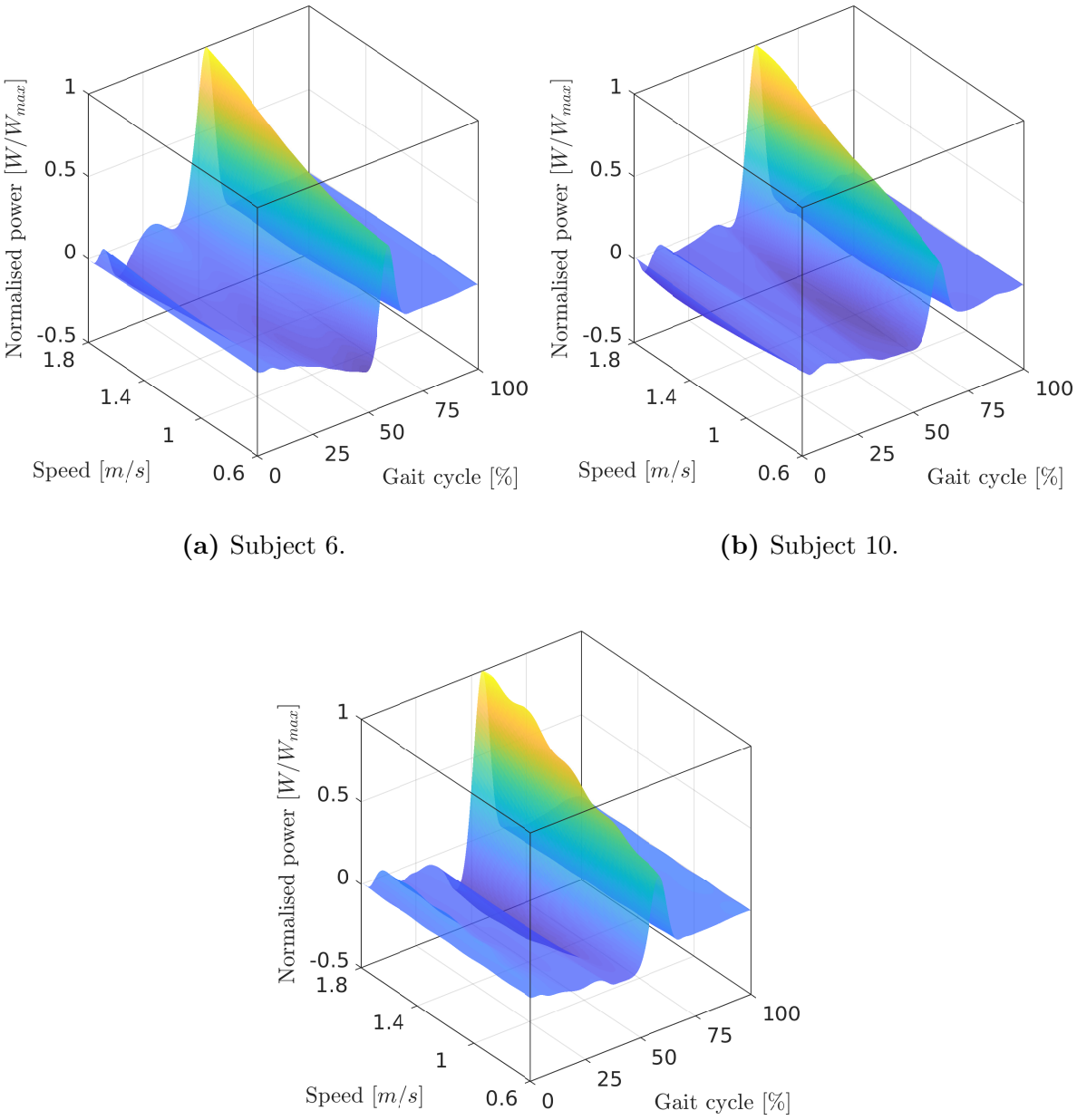
In mechanical systems, such as ours, we consider the combination of forces and the movements they give rise to (such as the ankle displacement). As can think of the ankle's movement as a rotation about a point, we can express its power as

$$\mathbf{P}(t) = \boldsymbol{\tau}^{\mathsf{T}}\boldsymbol{\omega} \tag{5.7}$$

where $\boldsymbol{\tau}$ is the torque and $\boldsymbol{\omega}$ the angular velocity, about the same point. Both quantities we can extract from the measurements, when inverse kinematics has been applied.

Figure 5.10 confirms what we already know (Farris & Sawicki, 2011): increase in walking speed occurs as a result of increase in the power exerted by humans at the ankle. Moreover, human observation suggests that this feature is unique to the ankle joint where a clear increase in power at push-off can be observed. This phenomenon is not present when looking at the knee or the hip joints in sagittal plane during human walking. Because this is a universal feature that can be observed in human observations, it can furthermore be reproduced in powered ankle-foot prostheses. That being said, this has been investigated before. See for example the work by Farris & Sawicki (2011) where the authors find that while "steady locomotion at a given speed requires no net mechanical work, moving faster does demand both more positive and negative mechanical work per stride". They also confirm that the ankle joint is chiefly responsible for the maximum percentage of total average positive power contributed, when undertaking locomotion, followed by the hip and then the knee joint. This is for all levels of human ambulation.

Required is a mechanism for measuring sensory information (such as IMUs) which inform the control system regarding the speed of ambulation. The plots shown in fig. 5.10 also clarify when the push-off phase starts and when it ends and the ankle enters the swing phase (refer back to fig. 5.4 for a refresher on the different phases). In the swing phase the power at the ankle drops to a value close to zero, as expected. Although the power plots for other subjects follow the same trend they are not included for brevity. Before moving onto the experimental section, we provide some more details on the observations used in this chapter.



(a) Subject 6.

(b) Subject 10.

(c) Subject 12.

Figure 5.10: Manifolds depicting the monotonic increase in power usage, with increase in speed, over one gait cycle, for three subjects from the Moore dataset (Moore et al., 2015). Surfaces were found using GPR and shown is the posterior mean surface, where the uncertainty surfaces have not been included. Adapted from (Dhir et al., 2018)

.

5.4.4 Experimental data

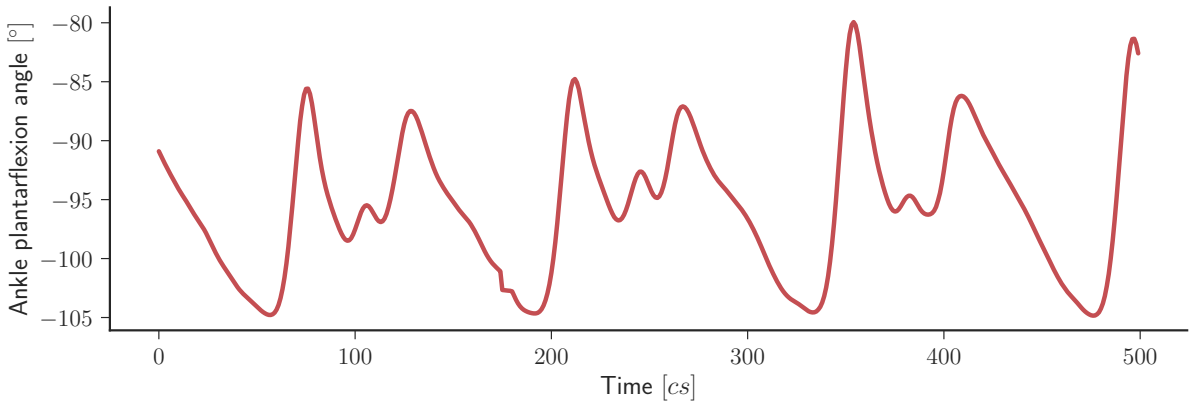
– for further details see the original study (Moore et al., 2015) The experimental data used for this study, comes from the excellent set of experiments conducted by Moore et al. (2015). Therein the authors collected a rich gait dataset with the help of fifteen

subjects, walking at three speeds (as shown in fig. 5.8) on an instrumented treadmill (a more advanced form of treadmill, with independent right and left tracks, and universal perturbation functionalities). They explain that each trial consisted of 120s of normal walking and 480s of walking while being longitudinally perturbed during each stance phase with pseudo-random fluctuations in the speed of the treadmill belt. We are primarily interested in the normal walking observations, but note that the methods developed herein would also form an interesting study if applied to the perturbed data as well. The details of the dataset are such that they contain: full body marker trajectories, ground reaction loads (labelled under the aforementioned gait events), two dimensional (2D) joint angles (i.e. those from the sagittal plane), angular rates and joint torques. All of these were collected at 0.8m/s, 1.2m/s and 1.8m/s, for each subject – for further details see the original study (Moore et al., 2015). An exposition of the raw ankle plantarflexion angle, over one gait-cycle, is presented in fig. 5.8, where the joint-angles were found using inverse kinematics. For manifold learning, in succeeding sections, all trajectories are normalised to have the same discretised length in time – unlike what is shown in the example in fig. 5.8.

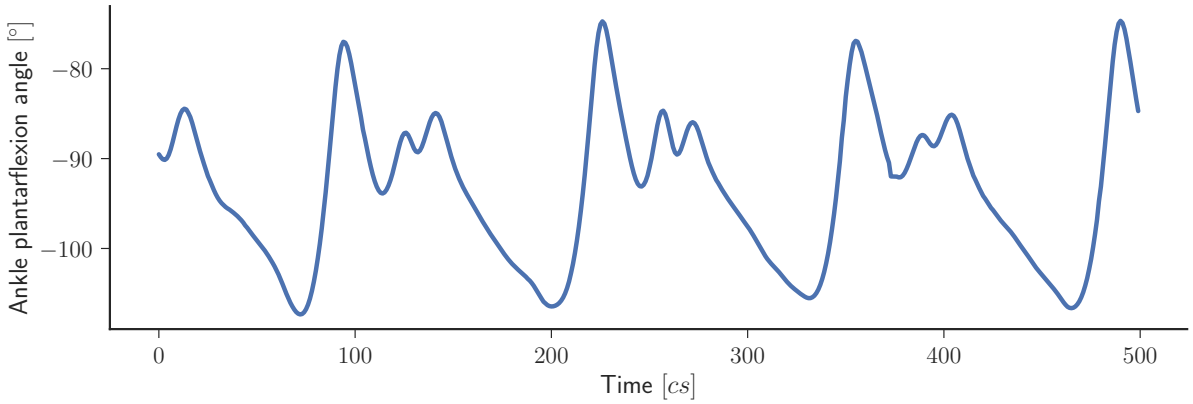
Further, we broadly follow the same protocol presented in section 'Processed data' of (Moore et al., 2015, p. 16), with minor edits which we employ to reinforce our simplifying assumption of this chapter, namely that human locomotion is, broadly, periodic. Hence, we are only interested in single-cycle evolution of the respective locomotion variable at hand, as we are then able to repeat the joint of all variables, to form our envelope, and hence can effect locomotion at any desired speed found in our set of test points $\mathbf{X}_*$, over an arbitrary number of cycles.

5.5 Simulation setup

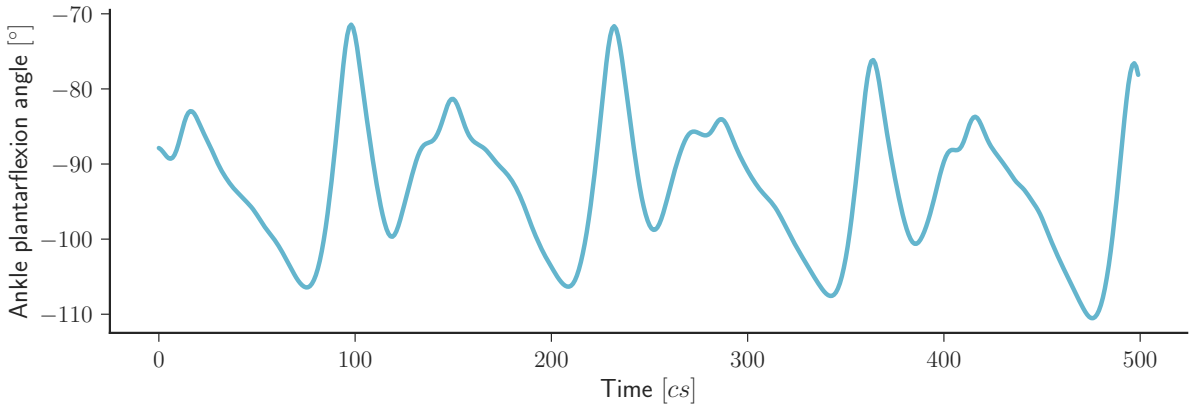
A planar model of the ankle foot prosthesis is developed in Robotran (Fisette & Samin, 1993). Like many before use we shall also employ impedance control. The justification for this is trivial; because our prosthesis is inherently interacting with its surrounding environment, we need to model that mechanical interaction. The scheme, which shows most promise in this regard, is impedance control (Hogan & Buerger, 2005). For the sake of completeness we review the fundamental ideas below.



(a) Measurements from a 19 year old male, at 1.70m and 92kg.



(b) Measurements from a 22 year old male, at 1.83m and 80.5kg.



(c) Measurements from 28 year old female, at 1.69m and 56.2kg.

Figure 5.11: Examples of three ankle plantarflextion angle evolutions, over a five second period. The angles were calculated from inverse dynamics, for three subjects with proportion, age, gender and build. Demonstrating the variation present in time-series observation which we are considering.

5.5.1 Information for control

Before moving onto the minutiae of impedance control, it is worth considering the information flow considered thus far. We have demonstrated how observations are received, extracted and regressed using Gaussian process regression. This information flow is summarised in in fig. 5.12.

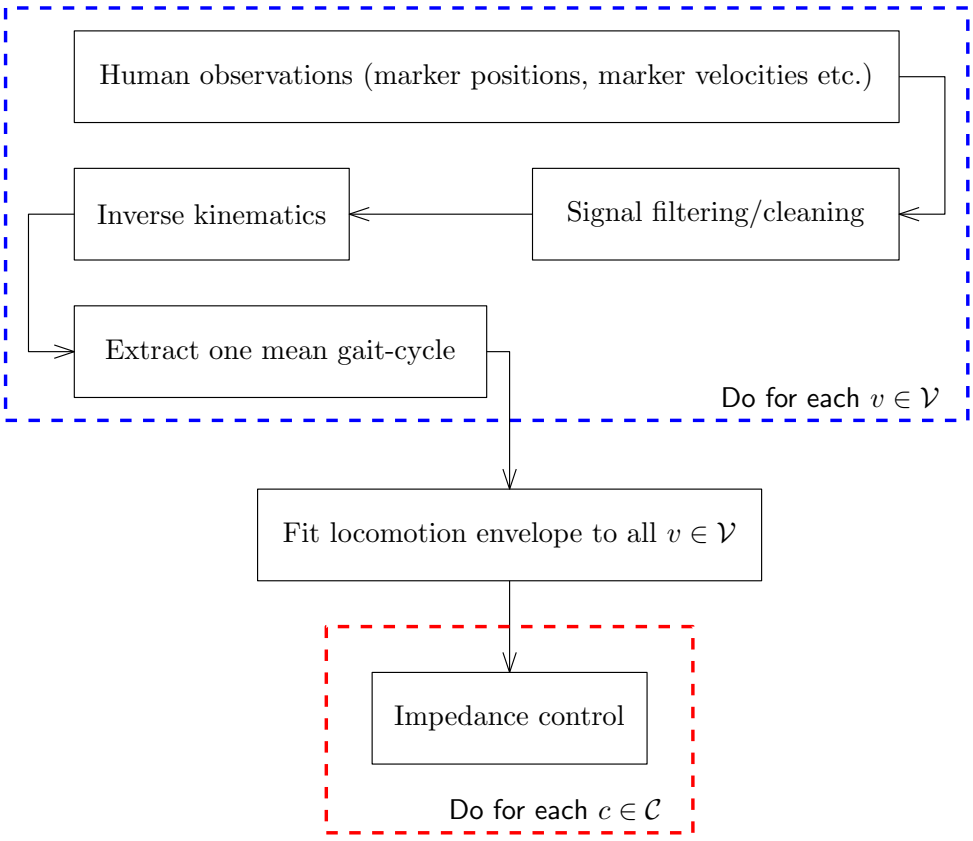


Figure 5.12: The information flow involved in using locomotion envelopes for control. The necessary training observations are extracted and preprocessed in the blue box. First, relevant signals are extracted such a MOCAP marker trajectory observations. Alternatives to MOCAP data could be e.g. accelerometry observations, taken from relevant positions of the lower body. Whichever observation set is used, these are subsequently cleaned using a variety of filtering and data cleaning processes, relevant details of which are found in (Moore et al., 2015, p. 16). Thirdly, inverse kinematics is used to find the relevant endogenous angles such as the plantarflexion angles, since we cannot measure these directly. Inverse kinematics is also used to find important gait reference events such as toe-off and heel-strike timings. Having extracted gait events which segment the entire gait-cycle , we are in a position to enumerate all of them (i.e. $\cup_{i=1}^{N} c_i = \mathcal{C}$) and subsequently find an average of the whole set of gait-cycles and standard deviation of one time-history (remember that we are only interested in one time-history of gait, in other words; one cycle). We repeat all of these steps, for all velocities v in the dataset (ideally v should be large to cover a large dynamic range of locomotion). Finally, having extracted mean gait-cycles for multiple velocities, we pass these to the locomotion envelope formalism, and subsequently the onboard impedance controller. Once on the impedance controller side (the red dashed box) the controller operates over an unbounded set $|\mathcal{C}| = [0, \infty)$, dynamically adapting throughout usage.

As fig. 5.12 shows, there are a number of steps that need to be undertaken to receive

a locomotion envelope over multiple velocities, which can be subsequently used in an impedance controller as described next.

5.5.2 Impedance control

Impedance control (IC) is an extensive control schema, in which a mass-damper-spring relationship (see example 5.1) between a position and force is established (Holgate et al., 2008, §C). Much like when a robot interacts with its environment, interaction forces result which, rather than being rejected, have to be accommodated as noted by Chan et al. (1991). The authors further note that to accomplish this, in addition to position control, force control is also required to accomplish the given task. It was explained elegantly by Mistry (2017), where he says that in order to design controllers that can cope with environmental uncertainty we must treat the robot as an impedance and its operating environment as an admittance. This is shown and explained in fig. 5.14b, and relates back to our first treatment of this topic in fig. 5.1.

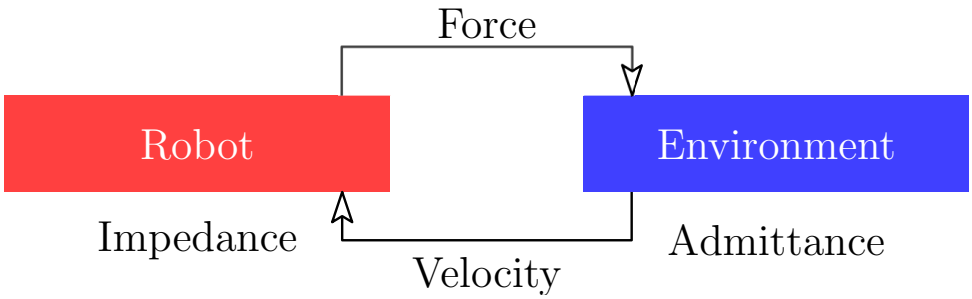


Figure 5.13: Understanding impedance control. Posit impedance as a dynamic operator that determines an output effort (force) given an input flow (velocity). Whereas an admittance is a dynamic operator that determines an output flow (velocity) given an input effort (force).

First proposed by Hogan (1984), IC yields a regime in which motion is commanded and controlled, and the "response for deviation from that motion" (Chan et al., 1991), resulting from the interaction force, is given in the form of an impedance (i.e. how much a robot resists motion when presented with an external force). This is desirable because it allows the robot the luxury of changing its effective dynamics in response to variations in its environment (Holgate et al., 2008). This can also be construed as robot changing its resistance to its surroundings, which could be, e.g. different types of ground surfaces or inclined and declined environments. Though attractive, one serious drawback is that in order for it to work properly, one needs incisive understanding of the force experienced by the robot in response to its environment.

EXAMPLE 5.1: SIMPLE IMPEDANCE CONTROL

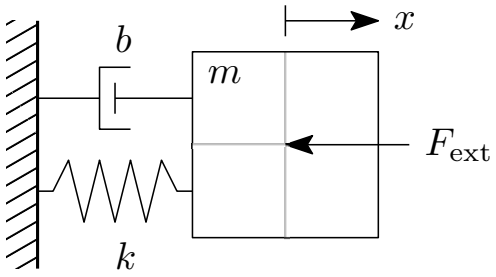
Consider an example of IC applied to a simple robot model, which can be used to simulate the actuation motion of a lower limb. Thus, in this instance, we are really looking for a specific relationship between the externally applied force and robot motion. Impedance control generalises the actuator so as to simulate a mechanical system, characterised by mass, damping and stiffness. A simple instantiation of such a robot is given by the 1-DoF linear system in fig. 5.14a. The equation of motion for this system is then given by

$$m\ddot{x} + b\dot{x} + kx = F_{\text{ext}} \tag{5.8}$$

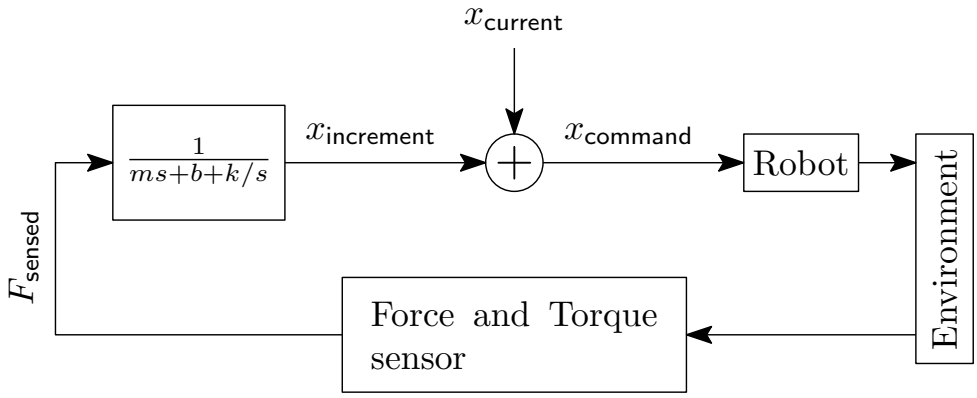
The Laplace-transformed equation of motion is given by

$$\frac{V}{F_{\text{ext}}} = \frac{1}{ms + b + k/s} \tag{5.9}$$

where V is the Laplace transform of the velocity $\dot{x}$. For a block diagram of this control scheme see fig. 5.14b. Example was inspired by Pham (2016).



(a) A robot end-effector, e.g a lower limb actuator model, idealised as a pure mechanical system. Mass, damping and stiffness parameters, are given respectively by m, b and k. An external force is applied, which is denoted by F_{ext}.



(b) Control block diagram of the example impedance controller. Where the control command sent to the robot is x_{command}, x_{current} is the current position and the required increment is $x_{\text{increment}}$.

Figure 5.14: An example of a simple one DoF system (fig. 5.14a) and its corresponding impedance controller (fig. 5.14b). Refer to example 5.1.

Having thus gained a firmer understanding of impedance control, consider now our application of the latter to our ankle prosthesis. Figure 5.7 shows four revolute joints

in total where three joints q_1, q_2, and q_3 are the floating base joints specifying the $x-y$ position and orientation of the shin in the plane (recall that for this study we are *only* operating in the sagittal plane). The fourth joint q_4, represents the planar motion of the ankle that is actively controlled during walking. The fifth joint q_5 is a passive toe with a fixed spring stiffness of 90Nm/rad and damping of 3Nms/rad that was added to better accommodate human foot kinematics. We let $\mathbf{q} \triangleq [q_1, q_2, q_3, q_4, q_5]$. With this in mind, like example 5.1, we can state the equation of motion for this system

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}^{\mathrm{T}}\mathbf{F}_{ext} \tag{5.10}$$

where $\mathbf{M}$ is the mass inertia matrix, $\mathbf{c}$ is the vector of Coriolis, centripetal and gravity forces, $\mathbf{J}$ is the Jacobian matrix, and $\mathbf{F}_{ext}$ represents the ground reactions forces. The kinematics and dynamic parameters were extracted from each subject's MOCAP data as reported in (Moore et al., 2015). See fig. 5.11 for examples of different gait-cycles, before normalising, taken from multiple subjects, at the same velocity. In particular notice the large variation in time-series, emphasising the difficulty of designing an adaptable control scheme.

The ground reaction forces and moment are applied at the foot's centre of pressure as shown in Fig. 5.7. The walking simulation consists of a swing phase where the ground reactions are zero and a stance phase where a part of the foot is in contact with the ground. For a reminder of the gait-cycle phases, please refer back to fig. 5.4.

In order to show the functionality of the proposed nonparametric regression methods, in changing the walking velocity, inverse dynamics is performed on the inferred manifolds to illustrate acceleration and deceleration during transitions from one speed to another. These transitions can occur at any moment during the gait cycle. A sensor is attached to the centre of the foot in the simulation to obtain important gait features such as step-size, step-frequency, velocity and acceleration. Further, consider that the stance phase of walking can be divided into three distinct sub-phases – see again fig. 5.4. The first is controlled plantarflexion (CP) that occurs right after heel strike. This phase is followed by controlled dorsiflextion (CD) that occurs when the angle between the shin and ankle starts to decrease. Finally, this is followed by powered plantarflexion or the push-off phase that is the main focus of this paper and it is where energy and power is injected into the walking gait (please refer back to the related work section, where this concept was

exhaustively discussed). The event timings of swing phase starts and ends are measured in the simulation.

5.5.3 Prosthesis impedance controller

The prosthesis used in this chapter is the device developed by Ficanha et al. (2016). It is a prosthesis designed to "meet the mechanical characteristics of the human ankle including power, range of motion and weight" (Ficanha et al., 2016). To allow for optimal placement of motors and gearboxes, transfer of power from the motors and gearboxes to the ankle-foot mechanism, uses a Bowden cable system. To control the prosthesis, impedance controllers in both sagittal and frontal planes were developed. We only consider the sagittal plane DoF. The impedance controllers used torque feedback from strain gages installed on the foot.

Two impedance controllers are required to control each motor independently. Each impedance controller uses an external position controller with an internal torque controller. The external position controller tracks a reference trajectory (generated through GPR) and uses the angle feedback from the encoders on each motor θ_m. The block diagram of the impedance controller is shown in fig. 5.15. The motors actuate the ankle in dorsiflexion-plantarflexion (DP) and inversion-eversion (IE) directions using Bowden cables that form a differential drive mechanism.

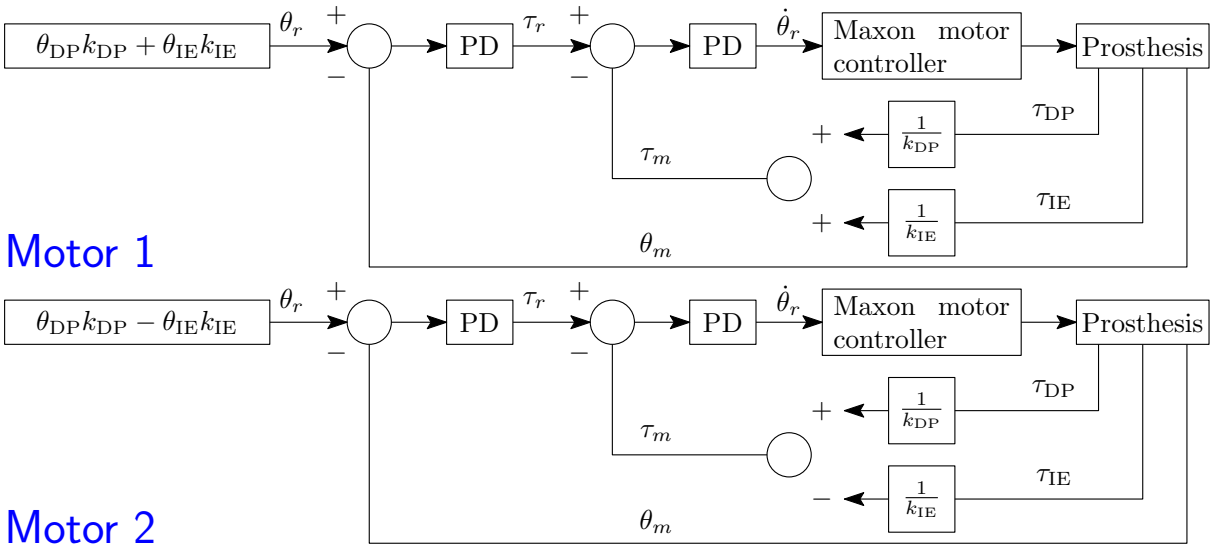


Figure 5.15: Impedance controllers for prosthesis motors. Adapted from Dhir et al. (2018).

The output of the external position controller is the desired torque to be generated by the motor τ_r and is the input to the internal torque controller. The internal torque controller uses torque feedback from strain gauges mounted on the foot. The strain gauges provide both the torque in dorsiflexion-plantarflexion (τ_{DP}) and the torque in inversion-eversion (τ_{IE}). Torques τ_{DP} and τ_{IE} are calculated from the strain gauges voltage outputs as described in previous work (Ficanha et al., 2016). Due to the differential drive nature of the mechanical setup, the sum of τ_{DP} and τ_{IE} is used for the reference torque (τ_r) for one of the motor controllers, and the difference of τ_{DP} and τ_{IE} is used in the other motor controller. Similarly, the reference trajectory θ_r for one of the motor controllers is the sum of the desired ankle angles in DP (θ_{DP}) and in IE (θ_{IE}), and for the other motor controller, the reference trajectory θ_r is the difference of the desired ankle angles in DP and in IE. The torque controller output is the desired motor velocity $\dot{\theta}_r$ and used as input to the Maxon motor controllers (Ficanha et al., 2016). The impedance controllers were implemented with a real-time frequency of 200Hz while the Maxon motor controllers used a proportional-integral controller running at 53kHz. The plus-derivate (PD) block shown in fig. 5.15 is used control the prosthesis in both the frontal and sagittal planes. As noted, we only consider the sagittal plane. For further details see the work by Ficanha et al. (2016).

5.5.4 Trajectory Generation

Our method inherently relies on the generation of trajectories which our controller can follow, to a high degree of accuracy. These trajectories contain the nominal phases; CP, CD and push-off, to simulate as closely as possible inferred gait patterns at velocities outside our training data.

The trajectories are extracted from the inferred manifolds in the envelope, to show cyclic behaviour at any speed and acceleration-deceleration transitions at any time during the gait cycle. These are the two key properties that are highly desirable to endow upon powered ankle-foot prostheses; speed adaptation and control repeatability. This is illustrated in fig. 5.17, where trajectories extracted at different velocities are shown in different colours. During human experiments the subject may walk with a certain self-selected step size and speed that can be both extracted from this manifold.

5.6 Empirical evaluation

In this section we apply our methodology to a set of experiments, demonstrating the utility in using locomotion envelopes for synthesising robust locomotion. First, however, we provide a brief exposition on our choice of kernels. Their selection is paramount for accurate application of these methods. The reasons for this is intuitive. Like so many problems in machine learning, we are interesting in the underlying mechanism that gave rise to the observations that we are trying to model. In this chapter, we have chosen to employ Gaussian processes, as our weapon of choice, for this task. Furthermore, we have also noted that the Gaussian process is defined by $m(\cdot)$ and $k(\cdot, \cdot)$ and since we will be setting the mean function to zero, we are left with a process that only depends on the covariance function.

5.6.1 Kernel design

The use of kernel-based nonparametric GPs has been alluded to in previous sections. Good performance for these methods is highly conditional on the choice of kernel structure *as it encodes our assumptions about the function which we wish to learn* (Rasmussen & Williams, 2006, §4). Typically this choice can be somewhat difficult, and methods have been proposed for automating the selection process (Duvenaud et al., 2013), in our case, however, we have substantial prior information regarding the nature of our multivariate regression surfaces.

The kernel function $k(\mathbf{x}, \mathbf{x}')$ determines how correlated or similar our outputs y and y' are expected to be at inputs $\mathbf{x}$ and $\mathbf{x}'$. These inputs could be e.g. time-dependent joint angles at a set velocity at some part of the gait-cycle, respectively. By defining the measure of similarity between inputs, the kernel determines the pattern of inductive generalisation.

In table table 5.2 we have summarised the kernel structure for the present variables in the envelope. These are, in no particular order: GRF (x, y), GRM (x, y), marker position (x, y) as well as ankle and knee joint-angles. In fig. 5.16 we compare some different kernels on the same regression problem, for a comparison of outcome.

Table 5.2: Compositional kernels used for kriging.

Target	ARD	Periodic	SE	RQ	M_{52}
Joint angles	✓				✓
GRF (x, y)	✓	✓	✓		
Marker Positions (x, y)	✓			✓	
GRM (x, y)	✓	✓	✓		

Figure 5.16 shows the posterior mean function without uncertainty bounds. To begin our discussion, most interesting is fig. 5.16b because although it generalises poorly in the velocity direction (recall: we are trying to infer locomotion parameters across a range of velocities) it overfits the training data. The squared exponential kernel with isotropic distance measure is given by

$$k_{\mathsf{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2l^2}\right) \tag{5.11}$$

where it can be seen that the length scales do not scale the inputs according to relevance of that input dimension. Unlike the other kernels we investigate, which all use this type of dimensionality scaling, fig. 5.16b does not and it can be seen that the temporal dimension dominates the others (i.e. the evolution of the gait cycle).

As we are only considering one period of the gait-cycle, as this is one of our primary modelling assumptions, we use periodic kernels where beneficial to do so, yielding functions of the form $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x} + P)$ where the P is the period of the gait-cycle. At times the addition of a periodic kernel has no benefit to the predictive performance, in which cases it has been omitted. Further, manifestly it is clear that the function values of $\mathbf{f}(\cdot)$ change faster, and more slowly, depending on which input dimension $\mathbf{x} \in \mathbb{R}^2$ we are considering. Intuitively this means that directional changes is of no importance. This is too strong an assumption in our case hence why isotropic kernels are unsuitable for our application domain. Instead we employ automatic relevance determination (ARD), which appropriately scales the inputs, thus determining the 'relevance' of each dimension. Consider the ARD SE kernel

$$k_{\mathsf{SE}}(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} \exp\left(-\frac{||\mathbf{x}_d - \mathbf{x}'_d||^2}{2l_d^2}\right) \tag{5.12}$$

where l_d is the length scale as a function of input dimension d. Note that small length scale value means that function values can change quickly, large values characterise functions that change only slowly.

The final part of our kernel design is rather more crucial as it concerns the innate periodicity assumptions of our data. Whilst it is clear that gait-cycles are periodic, they are *not exactly* periodic. This is further reinforced by our usage of the mean gait-cycle (for one period). Thus to allow for realistic variations over time, by design we make our kernels locally periodic, by multiplying by a local kernel. This allows us to model functions that are only locally periodic, the shape of the repeating part of the function can now change over time (Duvenaud, 2014). We experimented with different local kernels[4], each of which makes different smoothness assumptions about our data. The final form for each variable group, is shown in table table 5.2, where our design objective was to find natural looking, numerically consistent (with experimental data) simulations. Finally, as example exposition, a locally periodic version of the kernel presented in eq. (2.84), with ARD i.e. $k_{\mathsf{PER}\times\mathsf{SE}}(\mathbf{x}, \mathbf{x}')$, is given by

$$\prod_{d=1}^{D} \sigma_d^2 \exp\left(-\frac{2\sin^2\pi\left(\frac{||\mathbf{x}_d-\mathbf{x}'_d||}{p_d}\right)}{l_{1,d}^2}\right)\exp\left(-\frac{||\mathbf{x}_d-\mathbf{x}'_d||^2}{2l_{2,d}^2}\right) \tag{5.13}$$

where $l_{1,d}$ is the length scale as a function of input dimension d, for the k_{PER} kernel as is the periodicity p_d. Where $l_{2,d}$ is the length scale as a function of input dimension d, for the k_{SE} kernel. Having considered our kernel choices, we are in a position to use them for experimental evaluation. Before that, consider again the brief experimental comparison undertaken in fig. 5.16.

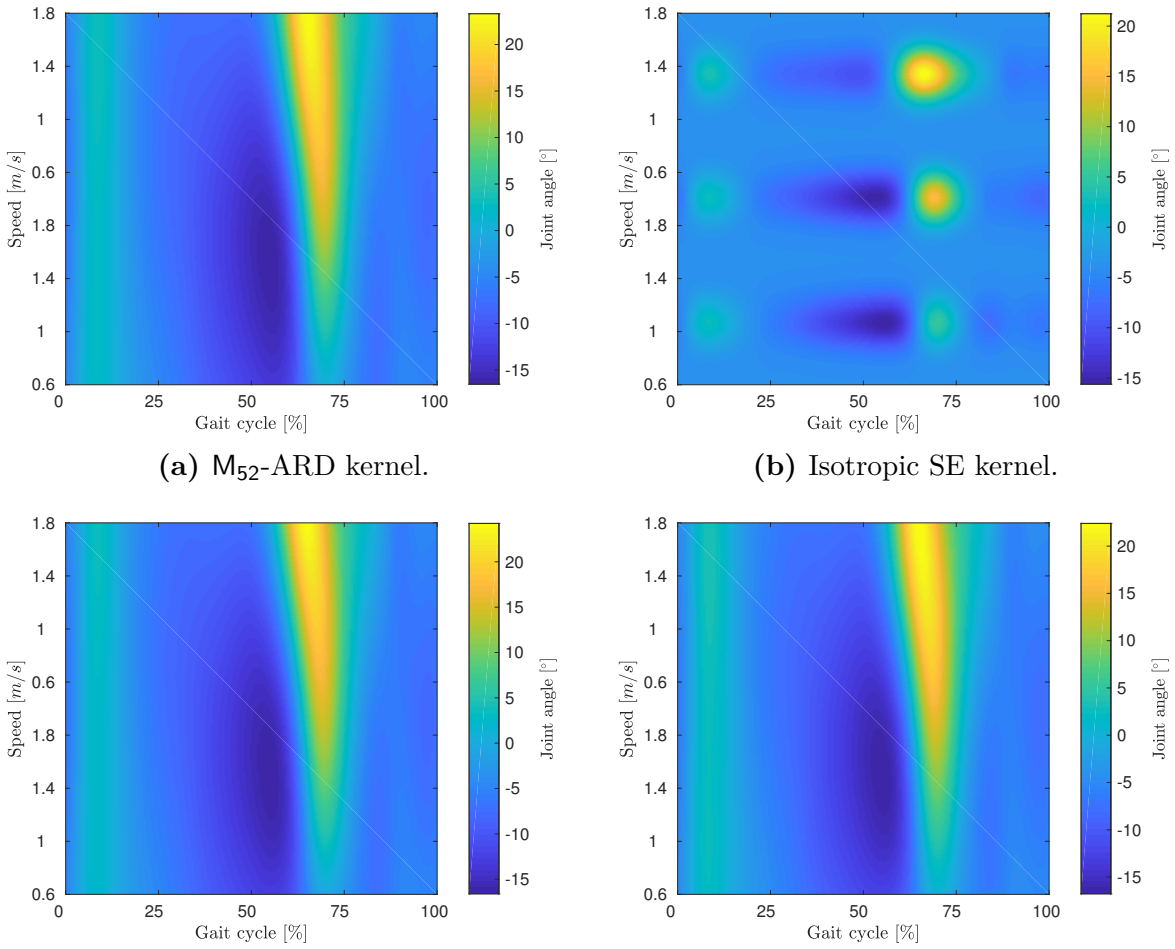
The numerical comparison of these kernels follow in table 5.2 where the negative log-likelihood (NLL) has been computed.

Kernel	NLL
Squared exponential with ARD	-3.1643×10^3
Isotropic squared exponential	-3.1139×10^3
Matérn 5/2 with ARD	-2.8275×10^3
Compound (Per-ARD with isotropic SE, multiplied w. SE-ARD)	-3.1785×10^3

Table 5.3: Minimisation of the negative log marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ with respect to the hyperparameters and noise level, under the chosen kernel.

Table 5.2 shows that as far as the NLL is concerned, there is not much difference between the kernel. Broadly they settle on the same posterior mean function, practically they do

[4]For details on the Matérn kernels with $\nu = 5/2$ (M_{52}) , and the rational quadratic (RQ) kernel, see (Rasmussen & Williams, 2006, §4.2).



(a) M_{52}-ARD kernel. **(b)** Isotropic SE kernel.

(c) SE-ARD kernel. **(d)** Compound kernel.

Figure 5.16: Investigating subject 10 and regression over the plantarflexion angle, the above plots depict different posterior predictive mean surfaces, using various kernel choices. As can be seen, aside an isotropic squared exponential kernel in fig. 5.16b, there is little to differentiate the posterior mean surfaces, and they evaluate to approximately the same mean posterior.

not, as the example with isotropic SE kernel shows. Without scaling the input dimensions according to their relevance, overfitting such as this will occur, hence the importance of using ARD for many problems. On the other, from a smoothness point of view, the posterior mean functions looks broadly the same, save for small differences, meaning that they are all viable candidates for our experiments.

5.6.2 Accelerating and decelerating

As stated at the beginning of this chapter, there is a gap in the literature concerning transitions between different velocities for powered prostheses, i.e. few are able to smoothly transition between user-selected velocities. As this is a key capability that we wish to incorporate into future devices, our first simulation experiments seeks to emulate this

behaviour. Figure 5.17 describe the scenario we are interested in; the different coloured paths represent gait-cycle at different velocities. A transition event seeks to smoothly go from one cycle to another, 'smoothly' here is a misnomer, as what we mean to say is a transition that is realistic and anthropomimetic.

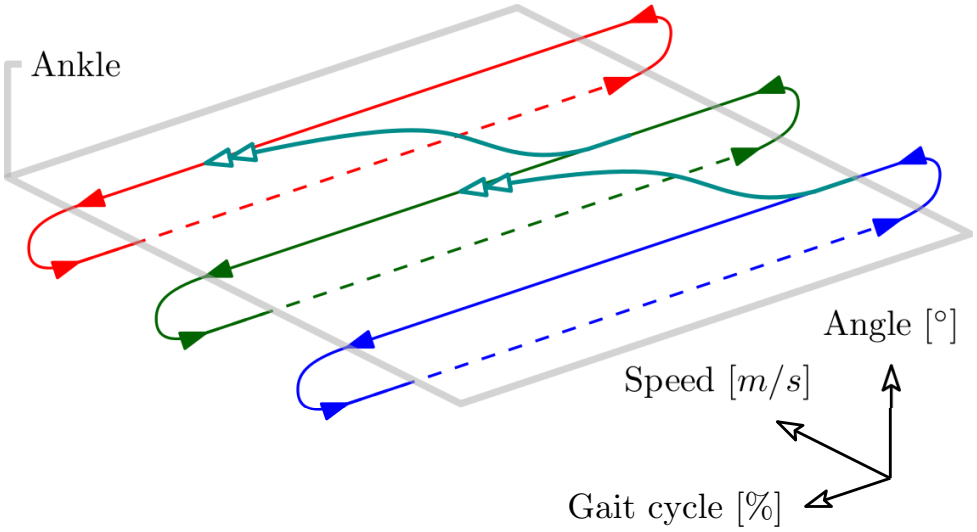
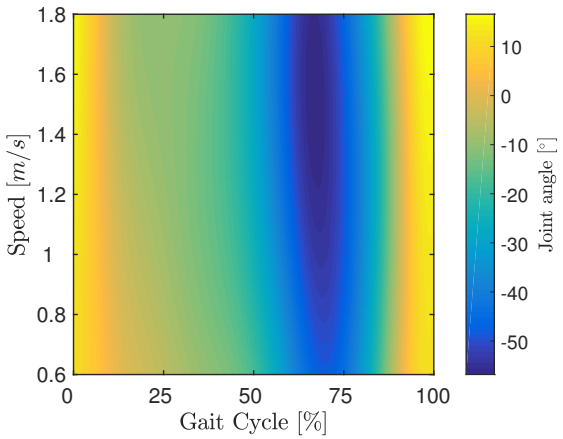


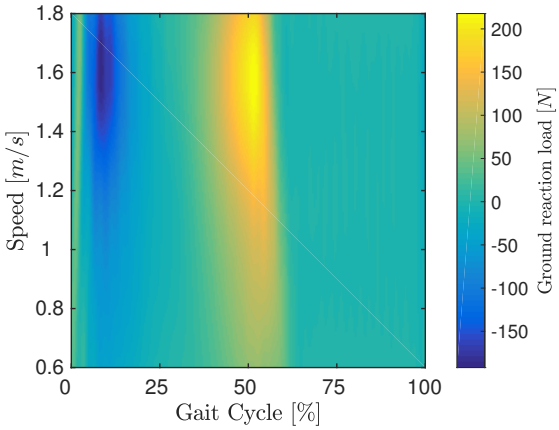
Figure 5.17: A mock-up scenario of potential transition paths, taking place on the inferred ankle manifold. Shown are possible gait-transition functions (double-headed cyan arrow paths). Trajectories of different colour, indicate trajectories at different velocities. The last curve, from the coordinate frame seen, operates under the highest velocity. The vertical direction of the coordinate system is a reference to any of the posterior mean function plots, see e.g. fig. 5.18, where the direction out of the page represents the target variable, and in this instance the target for the ankle is the plantarflexion angle.

In this experiment, a representative path on the inferred manifold (the posterior mean function) is chosen to simulate three speeds and their smooth transitions. The inferred joint angles and ground reaction forces and moments are used in inverse dynamics calculations to illustrate walking with speed adaptation at three walking speeds 0.6m/s, 1.2m/s and 1.8m/s. Hence, we seek to demonstrate our methodology simply by instantiating a simple forward simulation under our model and controller, to show that it is able to smoothly transition between gait-cycles as shown in the mock-up in fig. 5.17. The model is trained on plantarflexion curves at 0.8m/s, 1.2m/s and 1.6m/s, the simulation transitions between simulated curves (through GPR) at 0.6m/s, 1.2m/s and 1.8m/s. This is summarised in table 5.4.

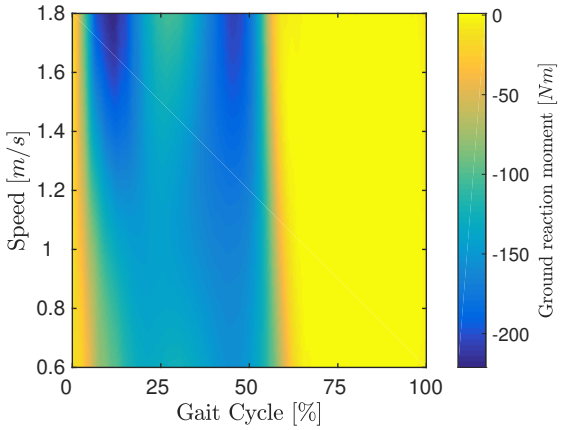
The walking speed is measured by placing a sensor on the foot in simulation that illustrates the instantaneous position, velocity and acceleration. Figure 5.20 demonstrates accelerations using our methodology. Each speed profile is repeated over three cycles before



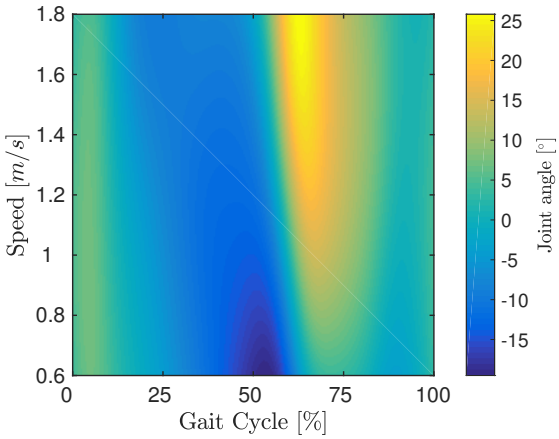
(a) Knee flexion angle.



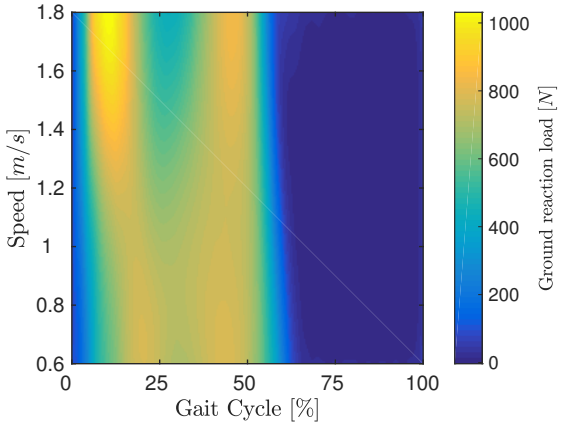
(b) Ground reaction load in x.



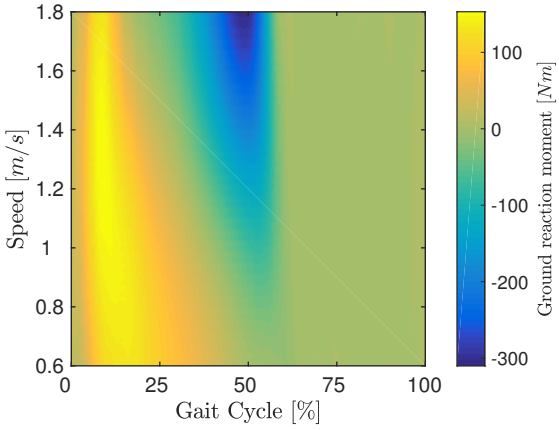
(c) Ground reaction moment in x.



(d) Ankle plantarflexion angle.



(e) Ground reaction load in y.



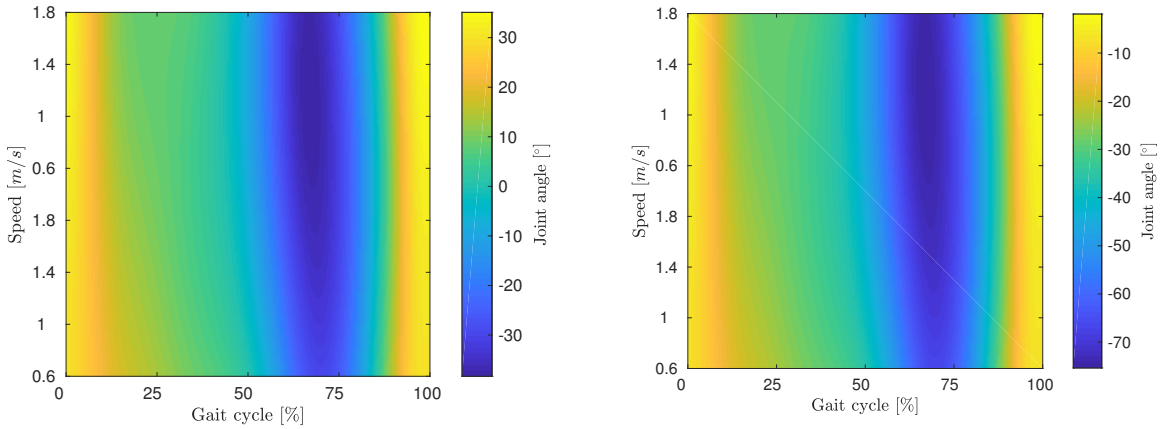
(f) Ground reaction moment in y.

Figure 5.18: A partial locomotion envelope, with constituent regression manifolds, for subject 6 from the Moore dataset (Moore et al., 2015). This envelope is used to instantiate a forward simulation. As before, for each manifold, training data was used at speeds 0.8m/s, 1.2m/s and 1.6m/s and GPR used to regress each variable in the envelope, to a speed domain of 0.6m/s to 1.8m/s, over one gait cycle. Refer to fig. 5.7 for the reference frame. For a summary of the velocities used for training and tests, refer to table 5.4.

Data	Low [m/s]	Mid [m/s]	High [m/s]
Train	0.8	1.2	1.6
Test	0.6	1.2	1.8

Table 5.4: Velocities used for training and testing in the locomotion envelope formalism. Note that testing takes place well outside the range of the training data.

acceleration to the next speed profile. It is interesting to note that the lowest (0.6m/s) and the highest speed (1.8m/s) were not available from the experimental data and they were inferred by GPR. Granted, many interpolation and extrapolation methods (e.g. support vector regression, neural networks, splines, piecewise linear interpolant, pure radial basis funcions etc.) are able to extrapolate well outside their optimisation domain. But as has been noted, GPR, this this of spatial regression, has particular benefits the primary of which is the uncertain incorporation. Though not shown in this example, we do have uncertainty bounds on all our posterior mean functions in fig. 5.18 – see fig. 5.19 for an example using the knee flexion angle (we use this as an example, because it is not an active degree of freedom in our simulation, hence serves merely as a demonstration of spatial regression with GPs).



(a) Posterior mean surface plus uncertainty surface.

(b) Posterior mean surfaces minus the uncertainty surface.

Figure 5.19: Knee flexion angle uncertainty bounds. Depicted are the posterior mean surfaces plus and minus t posterior uncertainty. Looking at the colorbar, it can be seen that the variability in the uncertainty is considerable.

Having extracted a locomotion manifold (see a partial envelope in fig. 5.18 for a set of

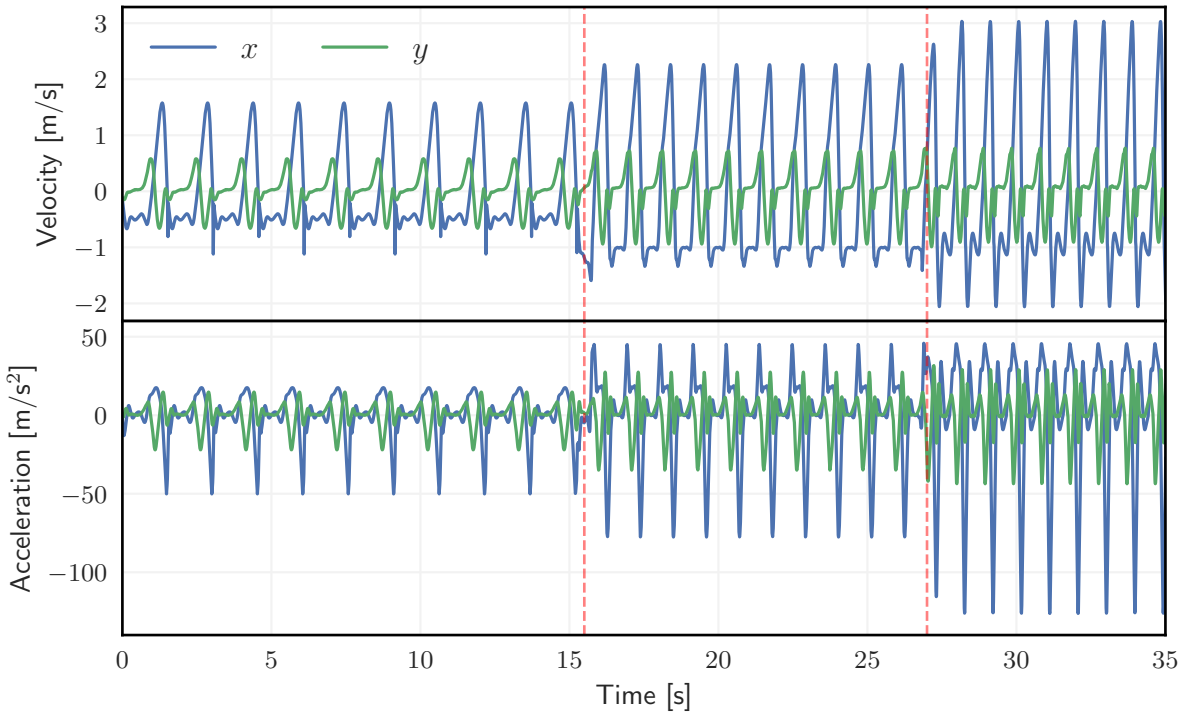


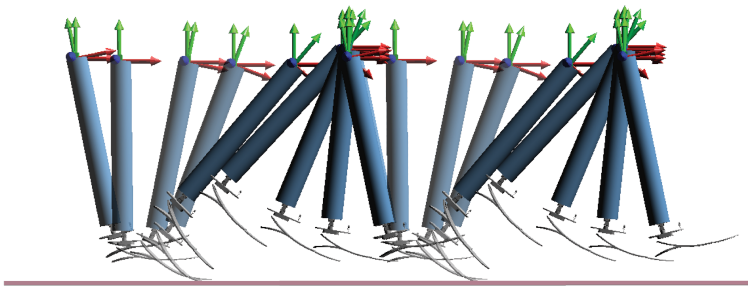
Figure 5.20: Forward simulation results for simulated transfemoral prosthesis, multiple walking cycles at each velocity, before transitioning to the next. Cartesian velocity and acceleration are shown of the foot, measured during two speed transitions, indicated by vertical dashed lines. Adapted from (Dhir et al., 2018).

training points that lay appreciably far away from the training data (circa ±15% for speed), we are in a position to synthesise locomotion sequences, consisting of several speeds, durations, accelerations and decelerations[5].

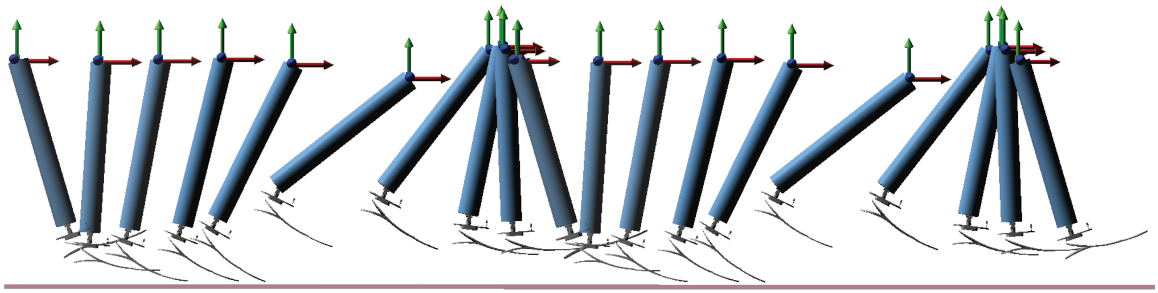
As can be seen from fig. 5.22 and fig. 5.21 we are able to demonstrate multiple properties of our method in this simulation. At the start of this chapter we emphasised the need for a prosthesis to be able to accelerate and decelerate, at the will of the user. In fig. 5.20 we see that acceleration is taking place is a smooth fashion (for more robust evidence see accompanying video in the supplementary materials). Taken from the regression manifolds, we enable the controller to generate motion that is consistent with those manifolds, as well as the transitions between the cycles on the manifold.

Secondly, the gait-cycle path was tracked using concatenated gait-cycles (see fig. 5.21), found from the learned (offline) multivariate regression functions. The gait is realistic and mimics subject six's gait well. More importantly, we demonstrate a physically sound gait, not just for GRFs and GRMs, but also joint-angle torques. The inferred joint trajectories

[5]See the supplementary material at https://youtu.be/iU7hNKLUX7c



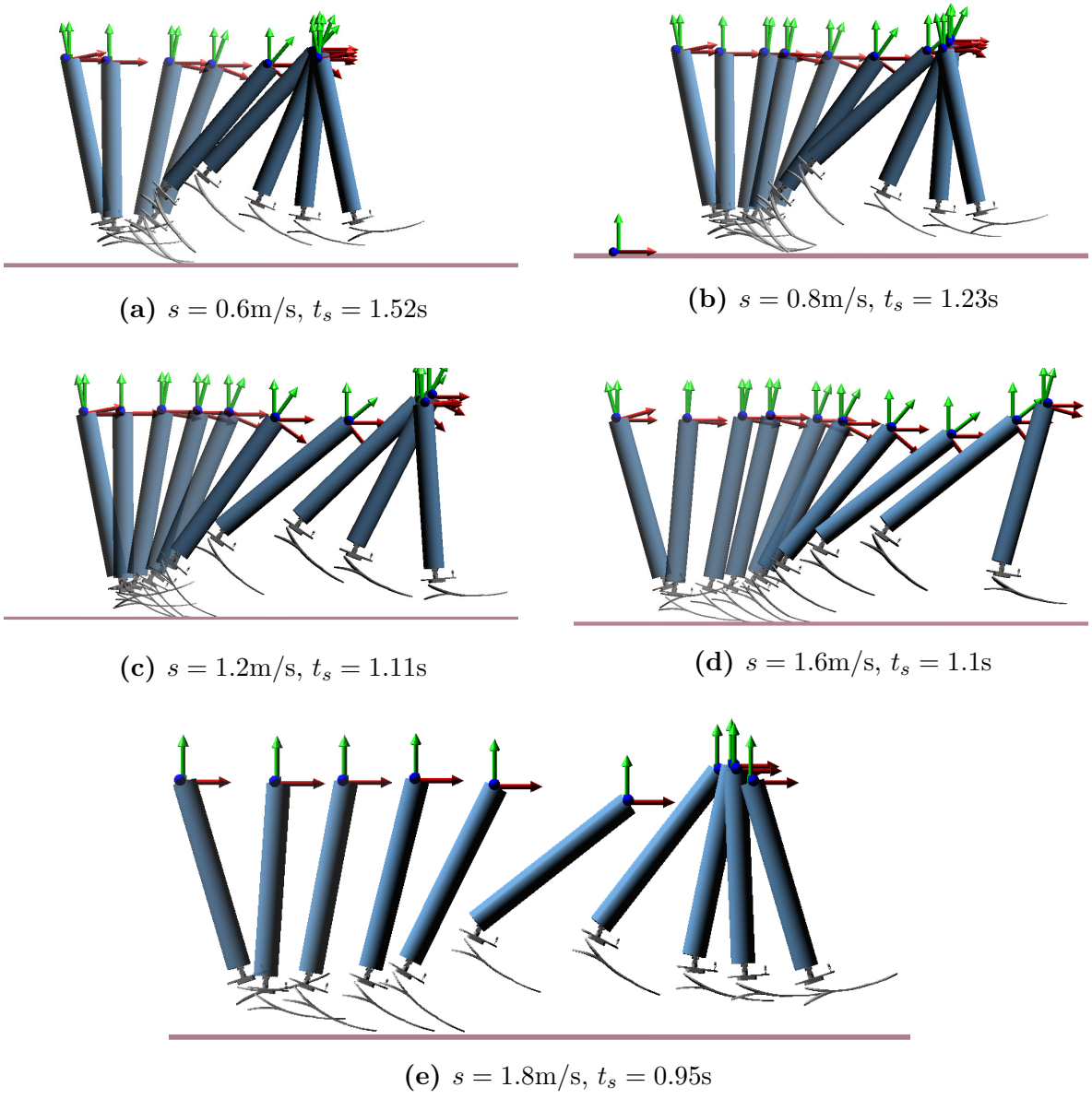
(a) $v = 0.6\text{m/s},\ t_s = 1.52\text{s}$



(b) $s = 1.8\text{m/s},\ t_s = 0.95\text{s}$

Figure 5.21: Long simulation experiments. Panels show gait cycle simulations for velocities outside the training data, where t_s is the stride time. These longer simulations, compared to fig. 5.22, demonstrate that the method does generate the expected (i.e. realistic looking) gait and cadence. Again, see the supplementary material for a video depicting this gait sequence. Adapted from (Dhir et al., 2018).

and ground reaction locomotion envelopes are used to simulate speed transitions from slow to normal to fast walking speeds. The snapshots of the inverse dynamics simulations different speeds are shown in fig. 5.22.



(a) $s = 0.6\text{m/s}$, $t_s = 1.52\text{s}$

(b) $s = 0.8\text{m/s}$, $t_s = 1.23\text{s}$

(c) $s = 1.2\text{m/s}$, $t_s = 1.11\text{s}$

(d) $s = 1.6\text{m/s}$, $t_s = 1.1\text{s}$

(e) $s = 1.8\text{m/s}$, $t_s = 0.95\text{s}$

Figure 5.22: Simulation experiments. Panels show gait cycle simulations for velocities outside the training data, where t_s is the stride time. Adapted from (Dhir et al., 2018).

5.6.3 Torque-angle relationship at test points

From simulation we can extract torque-angle curves at the knee and angle angles in the sagittal DoF, and compare them to empirical results found in (Moore et al., 2015). In fig. 5.23 we have have super-imposed curves for speeds $s = 0.6\text{m/s}$, 1.8m/s, to the experimental ones found at speeds 0.8m/s, 1.2m/s and 1.6m/s. It is clear that the method faithfully extrapolates the curves, by carrying curve shape and appearance, as speed increases, or decreases, where the extrema of the test points are shown with two sets of dashed curves. The curves compare well with the observed simulated gait-cycles in

fig. 5.22.

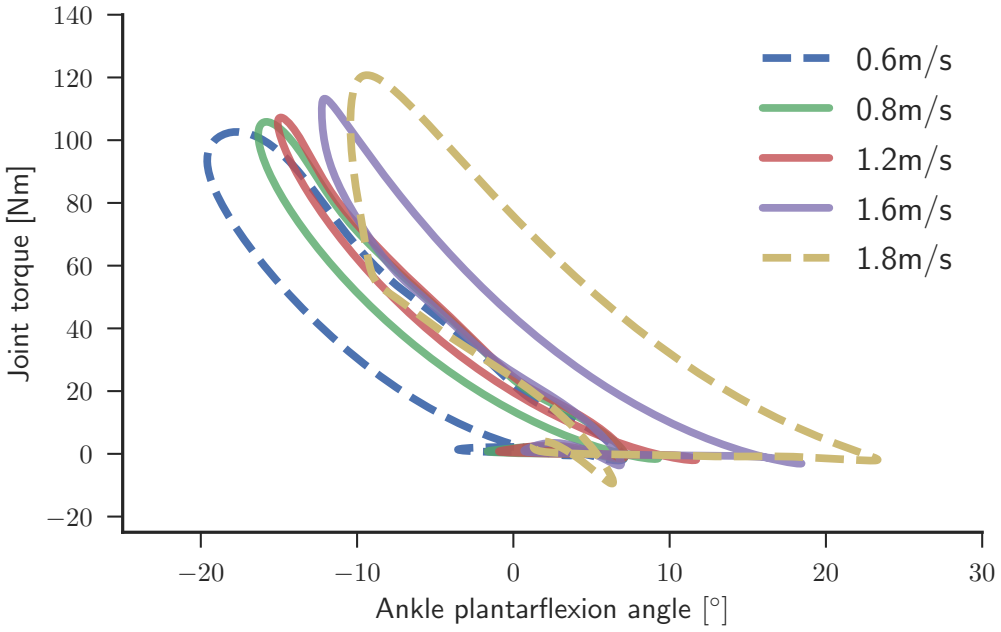


Figure 5.23: Measured and inferred ankle kinetics for subject 6. Curves are shown for velocities inside and outside the training data. Dashed curves lie on the extrema of the test points and the *experimental* values for the curves of 0.8m/s, 1.2m/s and 1.6m/s are shown as well. Adapted from Dhir et al. (2018).

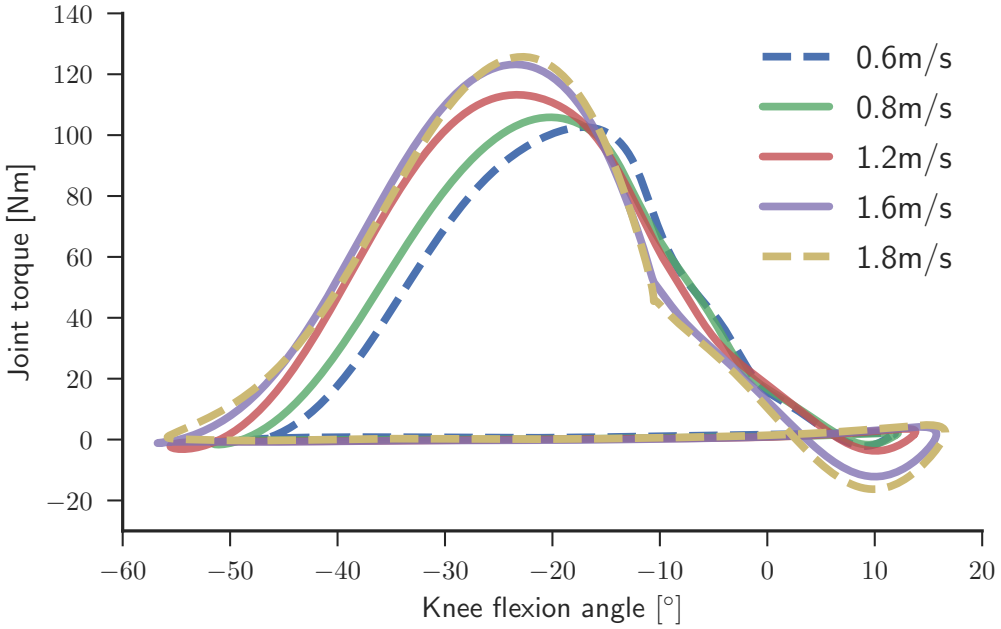


Figure 5.24: Measured and inferred ankle kinetics for subject 6. Curves are shown for velocities inside and outside the training data. Dashed curves lie on the extrema of the test points and the *experimental* values for the curves of 0.8m/s, 1.2m/s and 1.6m/s are shown as well. Adapted from Dhir et al. (2018).

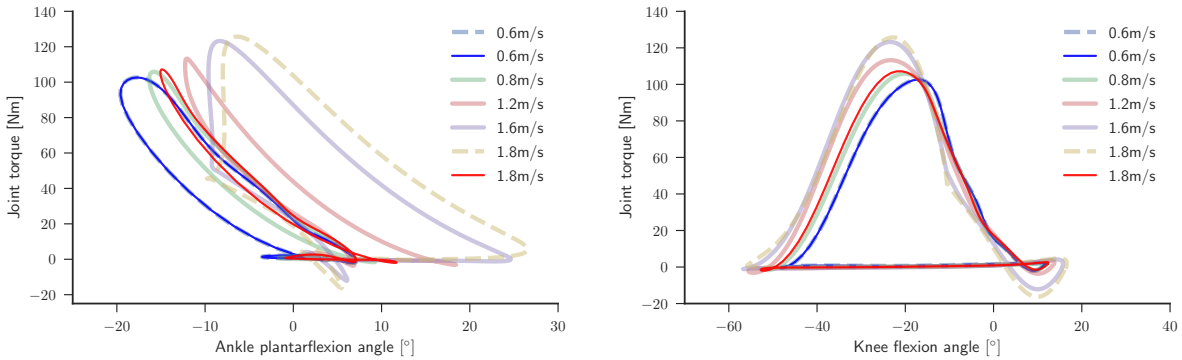
The prior expectation of our experiments, is that the generated curves should fall within

the same domain as the experimentally found curve. As we are not comparing to held-out data here (we will however in §5.6.4), we are merely interested in seeing how GPR compares to other methods with respect to propagating the aforementioned curve properties.

In the following figures we compare GPR to common interpolation and extrapolations methods. We apply these methods to the three training velocities used thus far, for the ankle and knee moments and knee inflexion and ankle plantarflexion angles. Following this we interpolate and extrapolate the manifold up to and including 0.6m/s and 1.8m/s, in order to allow us to compare their regression properties to GPR.

5.6.3.1 Nearest neighbour interpolation

In fig. 5.25 the results for nearest neighbour interpolation have been superimposed on those of GPR for the same task. The GPR used M_{52}-ARD and SE-ARD kernels for angles and moments respectively (throughout). As can be seen from both subplots, the method captures the lower bound well (0.6m/s) but does not capture the correct shape of the upper range of velocity. *Why?* The algorithm selects the value of the nearest point but does not consider the values of other, neighbouring points, yielding a piecewise-constant interpolant. For a shape nonlinear as the torque-angle curve, this yields an unsatisfactory result. Secondly, this algorithm is primarily intended for unstructured inputs, hence why it is used in this 'naive' comparison fashion. Our data, though very sparse, is no unstructured when passed to the interpolant.

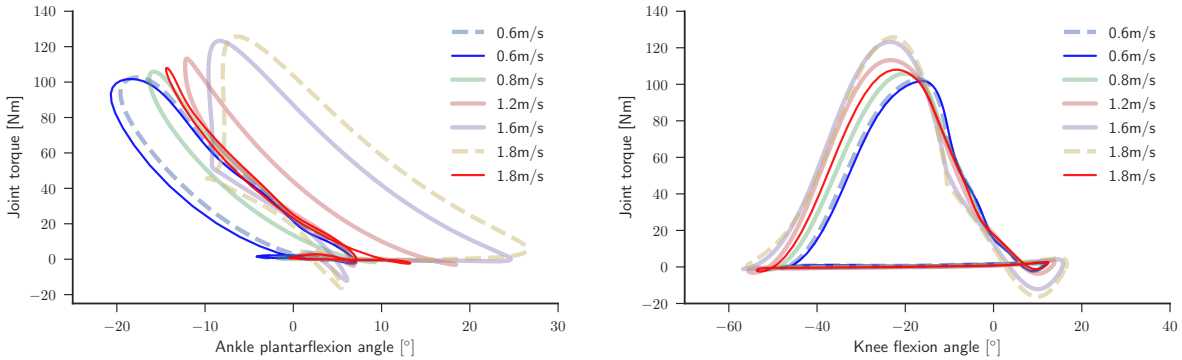


(a) Ankle. **(b)** Knee.

Figure 5.25: Nearest neighbour interpolation superimposed on the same regression task as GPR. Shaded curves show the GPR results, and solid lucent curves, show the comparison method. For each manifold, training data was used at speeds 0.8m/s, 1.2m/s and 1.6m/s and GPR used to regress each variable in the envelope, to a speed domain of 0.6m/s to 1.8m/s, over one gait cycle. Refer to fig. 5.7 for the reference frame.

5.6.3.2 Linear regression

In fig. 5.26 we employ a piecewise linear interpolant. We receive much the same results as for the nearest neighbour interpolant in fig. 5.25. Again, it is not difficult to see why this produces poor results. We are trying to linearise a highly nonlinear surface, with a method poorly suited for that purpose. That being said, the benefits of these linear methods, are that they are simple, do not require any function selection (i.e. a kernel) and they are comparatively fast (this is not really an issue for us, but for larger problems it could be). Continuing, see that the solid lines in fig. 5.26 do a poor job of regressing the surface to the extremities, worse in fact than the nearest neighbour interpolant. Granted, the performance is somewhat better for the knee curve in fig. 5.26b than the ankle curves in fig. 5.26b. In detail, the interpolant constructs a triangulation of the inputs using convex hull constructor (the convex hull in Euclidean space is the smallest convex set, that contains the set of points) and then, on each triangle, performs linear interpolation. This triangulation will be significantly warped when applied to semi-structured data such as ours, possibly way regression performs as badly as shown.



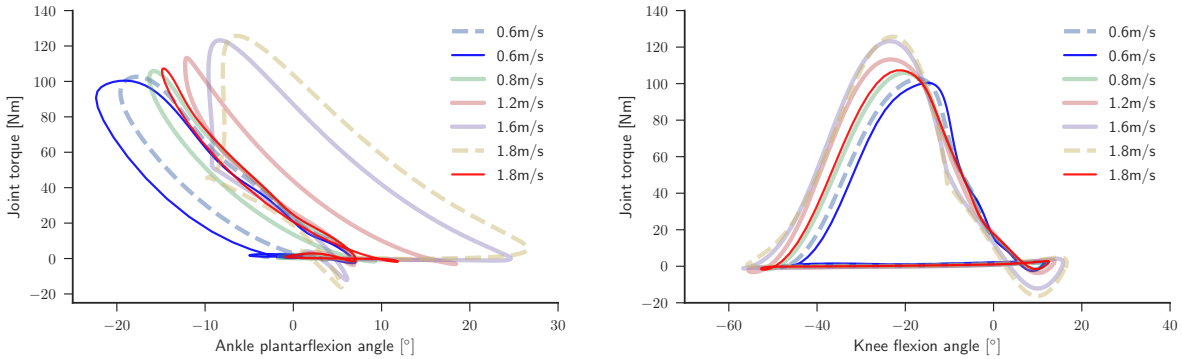
(a) Ankle. **(b)** Knee.

Figure 5.26: Linear regression superimposed on the same regression task as GPR. Shaded curves show the GPR results, and solid lucent curves, show the comparison method. For each manifold, training data was used at speeds 0.8m/s, 1.2m/s and 1.6m/s and GPR used to regress each variable in the envelope, to a speed domain of 0.6m/s to 1.8m/s, over one gait cycle. Refer to fig. 5.7 for the reference frame.

5.6.3.3 Piecewise cubic curvature-minimising interpolation

In fig. 5.27 results are shown for cubic spline interpolation. Again, as this is a naïve method we do not expect good results on this task. Because this methods operates much as the

nearest neighbour interpolant, with regards to the unstructured nature of the data, and the subsequent triangulation, a similarly poor result is achieved. Of the three methods, this is perhaps the worst, at least for the ankle regression, where the methods do not have a similar spread as the experimental data, nor GPR. Naturally we should not be comparing GPR and this method on a on-by-one basis, rather comparison is done by considering how well they place onto the experimentally obtained curves. We expect the solid lucent curve, in fig. 5.27a to be aligned with, or above, the experimentally obtained counterpart at 1.6m/s. This is not observed for the ankle. Rather, it is wedged in between 0.8m/s and 1.2m/s, thus underestimating the torque-angle relationship.



(a) Ankle. **(b)** Knee.

Figure 5.27: Cubic spline interpolation superimposed on the same regression task as GPR. Shaded curves show the GPR results, and solid lucent curves, show the comparison method. For each manifold, training data was used at speeds 0.8m/s, 1.2m/s and 1.6m/s and GPR used to regress each variable in the envelope, to a speed domain of 0.6m/s to 1.8m/s, over one gait cycle. Refer to fig. 5.7 for the reference frame.

Having employed naïve methods for comparing GPR, we now employ slightly more involved schemes for measuring, numerically, the performance of GPR on a held-out task.

5.6.4 Torque-angle relationship for held-out observations

In the study by Liu et al. (2008), the authors studied muscle contributions related to the providing vertical support and forward progression of the mass centre of the human body. To quantify these contributions, over a range of walking speeds, three-dimensional muscle-actuated simulations of gait were generated and analysed for eight subjects walking overground at very slow, slow, free, and fast speeds (Liu et al., 2008). To examine the contributions of muscles to the acceleration of the mass centre, they gait analysis data

at four walking speeds. We will use these observations in our held-out experiment, to measure the NLL predictive performance, using the posterior mean surface, compared to other involved methods.

What are involved methods? By 'involved' we mean methods that bear some resemblance to GPs. Snelson (2006) and Rasmussen & Williams (2006), note that there are a number of methods that can be interpreted as instances of GPs. Some include: generalised linear regression, neural networks (become GPs when there are infinite amount of hidden units), spline models and support vector regression. We shall consider splines for their inherent simplicity.

All models are trained on three sets of curves, constituting the plantarflexion angle, recorded at three different velocities: 0.8m/s, 1.17m/s and 1.64m/s. The regression problem seeks to estimate, under these training examples ($N = 3$), the response at 0.61m/s. These data points are selected from an 18 year old female subject, weighing 63.1kg. She was chosen because she bears the greatest physiological resemblance to the others subjects in this chapter (the other subjects in that cohorts are all children).

Before looking at the results, lets consider some radial basis functions (RBF). Note that radial basis function interpolation is a common approach to scattered data interpolation (Anjyo & Lewis, 2011). The RBF and GPR models initially seem quite different. GPR is a weighted sum of the data, whereas RBF is a weighted sum of a kernel indexed by the distances between the data. Anjyo & Lewis (2011) demonstrates that under some conditions they are in fact the same, in particular when one employs a Gaussian RBF, the GPR model is received. For this reason we shall not employ a Gaussian RBF, as we are more interested in properties of the RBF model, that do not yield a model equivalence.

We consider the multiquadric and cubic RBFs:

$$\phi(r) = \sqrt{1 + (\epsilon r)^2} \tag{5.14}$$

and

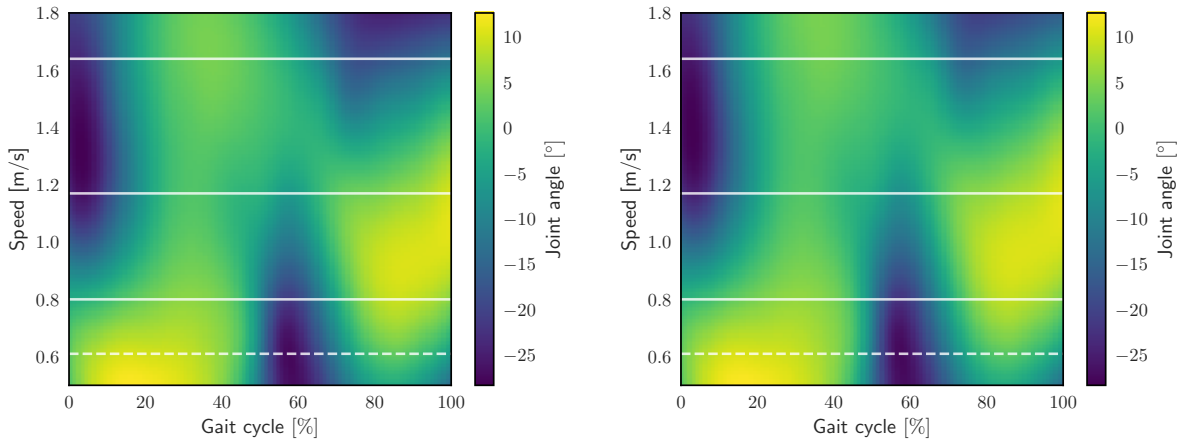
$$\phi(r) = r^3 \tag{5.15}$$

where $||\mathbf{x} - \mathbf{x}'||$. Interpolation functions generated (Rocha, 2009) from an RBF, are represented as

$$g(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi(||\mathbf{x} - \mathbf{x}^j||). \tag{5.16}$$

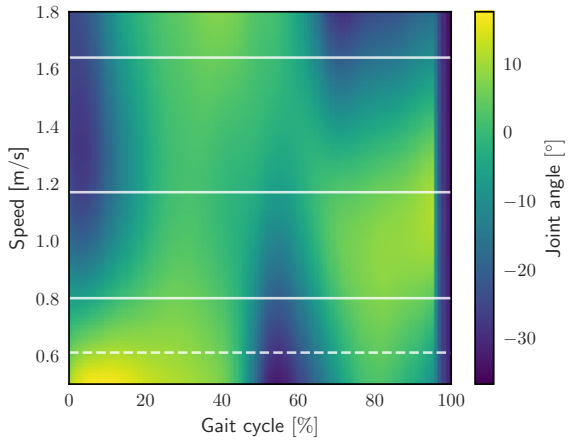
the goal is the to construct an estimation model to $f(\mathbf{x})$ – i.e. the latent underlying dynamics in our plantarflexion angle for example. We will not provide more detail on this model, but refer the interested reader to the references for further details.

The results from the regression task is pictorially shown with function heatmaps in fig. 5.28 and numerical results are found in table 5.5 as well as a method-by-method graphical comparison shown in fig. 5.29.

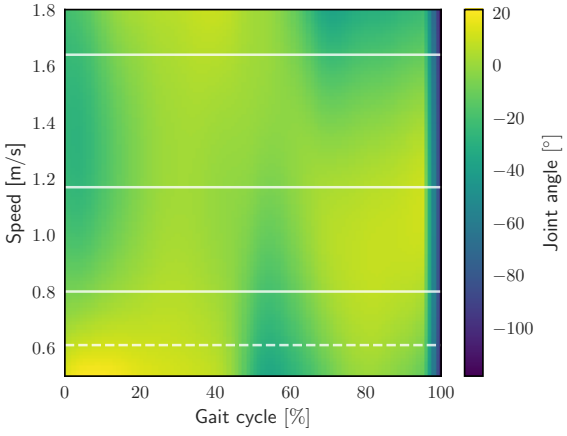


(a) SE-ARD kernel.

(b) M_{52}-ARD kernel.



(c) Multiquadric RBF.



(d) Cubic RBF.

Figure 5.28: Heatmaps of various multivariate regression techniques, applied to the same problem. The top row shows GPR with two different kernels. The bottom rows shows RBF interpolation with two different basis functions. The solid horizontal white lines correspond to the training observations $\mathbf{X}$ and the test-input $\mathbf{x}_*$ is depicted by the dashed horizontal white line. The purpose of this exercise it to predict the plantarflexion angle response at the dashed line.

Model	Heatmap	RMSE [°]
GPR with SE-ARD kernel	fig. 5.28a	3.73 ± 1.48
GPR with M_{52}-ARD kernel	fig. 5.28b	6.58 ± 0.17
RBF with multiquadric kernel	fig. 5.28c	7.30
RBF with cubic kernel	fig. 5.28d	9.51

Table 5.5: Root mean squared error (RMSE) on held-out data from (Liu et al., 2008). The related regression task for these results are shown in fig. 5.28. The results from the GPR are the posterior predictive mean values.

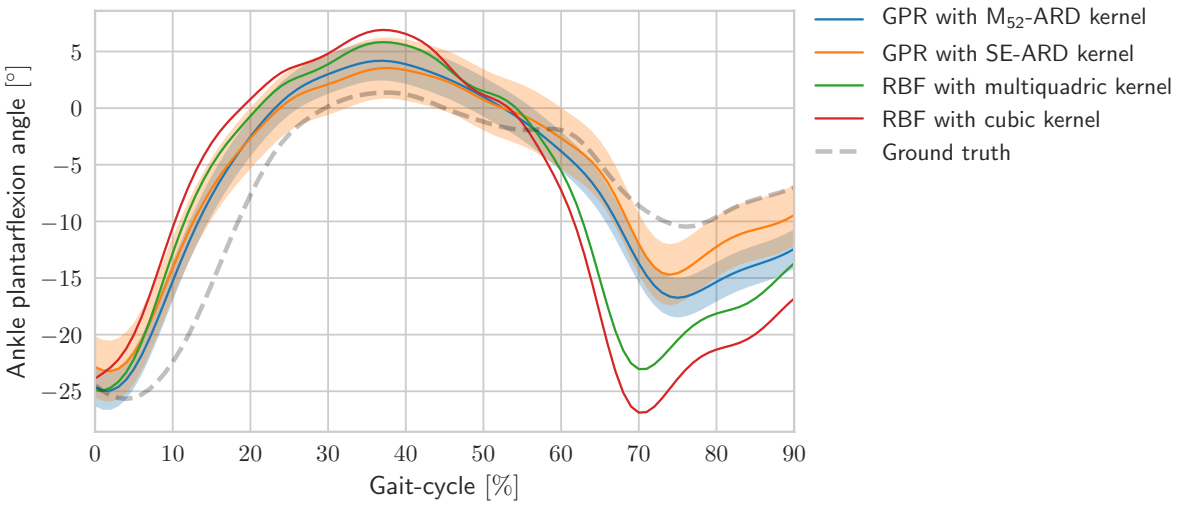


Figure 5.29: Shown are the results, for different methods, at inferring the ankle plantarflexion angle for subject 8, from the (Liu et al., 2008) dataset. Superimposed are the resulting curves from the respective regression manifolds shown in fig. 5.28 (cross-section at the dashed white line is shown above). The ground-truth, i.e. the experimentally measured values, are shown by the opaque dashed curve. Corresponding numerical RMSE results are shown in table 5.5. The posterior uncertainty estimates for the GPR model are shown with opaque bounds on the posterior mean function for both model instances (i.e. different kernels).

With fig. 5.28, table 5.5 and fig. 5.29, we are in a position to make some concluding remarks regarding these regression models. First, the RBF model is competitive when compared on a held-out data task such as this one. Indeed, the plantarflexion angle is tracked almost as well, as for with the GPR model. However, GPR does record a higher RMSE, especially using the Matërn kernel, as it it a very smooth covariance function, which is a sound choice given the smooth nature of our test target, as well as our training data. Moreover, GPR gives us uncertainty bounds – as fig. 5.29 shows. These uncertainties are numerically quantified in table 5.5 as RMSE scores. Alas, we demonstrate that GPR, for this task, is superior to competitive methods on the same task. No doubt there will be other methods, that perform better, but most likely not at the same cost. The GPR

model, for a small dataset like this, performs very well, on what is a difficult nonlinear manifold estimation task. Certainly, the RBF model does well too, but with the drawback that this model does not yield uncertainty bounds.

5.6.5 Hardware experiments

In this section, the results of a healthy subject walking in a straight line at two different speeds (outside our training data) with the help of the ankle-foot prosthesis are presented. In this experiment, only the DP DoF of the ankle-foot prosthesis is used to provide push off while IE is controlled to stay at zero. Figure 5.30 illustrates the amount of push-off trajectory and power provided by the ankle-foot prosthesis at two different speeds. It is shown that in order to sustain the balance at higher walking speeds, the increased power at the ankle-foot prosthesis is necessary. A video[6] of this experiment is also provided with this chapter to better illustrate the results. Current devices attempting the same task are typically equipped with controlled actuators which can replicate biomechanical characteristics of the human ankle (in as much as they have been tuned for that specific gait), improve the amputee gait and reduce the amount of metabolic energy consumed during locomotion. However, as also noted by Mai & Commuri (2016), the functioning of such devices on human subjects is difficult to test due to:

- changing gait;
- unknown ankle dynamics;
- complicated interaction between the foot and the ground;
- complicated interaction between between the residual limb and the prosthesis.

We do not propose to deal with all of these raised issues. But we can deal with the first one. Alas, the observations $\mathbf{X}$ that we use for this experiment, are those extracted from the (Moore et al., 2015) dataset. Explicitly this means that for that regression task, demonstrated in the previous simulation section; $N = 3$. Which it to say for multivariate regression, we use three curves, evolving in time, of various features related to locomotion. These features are listed in table 5.2 as well as the kernel combinations we use for each, as applied to the GPR model.

[6] See the supplementary material at: https://youtu.be/FGnhBkR7xD0

5.6.5.1 Pre-processing and single-cycle extraction

There are two main tasks we deal with in this section 1) pre-processing of the data and 2) single gait-cycle extraction. We start with pre-processing.

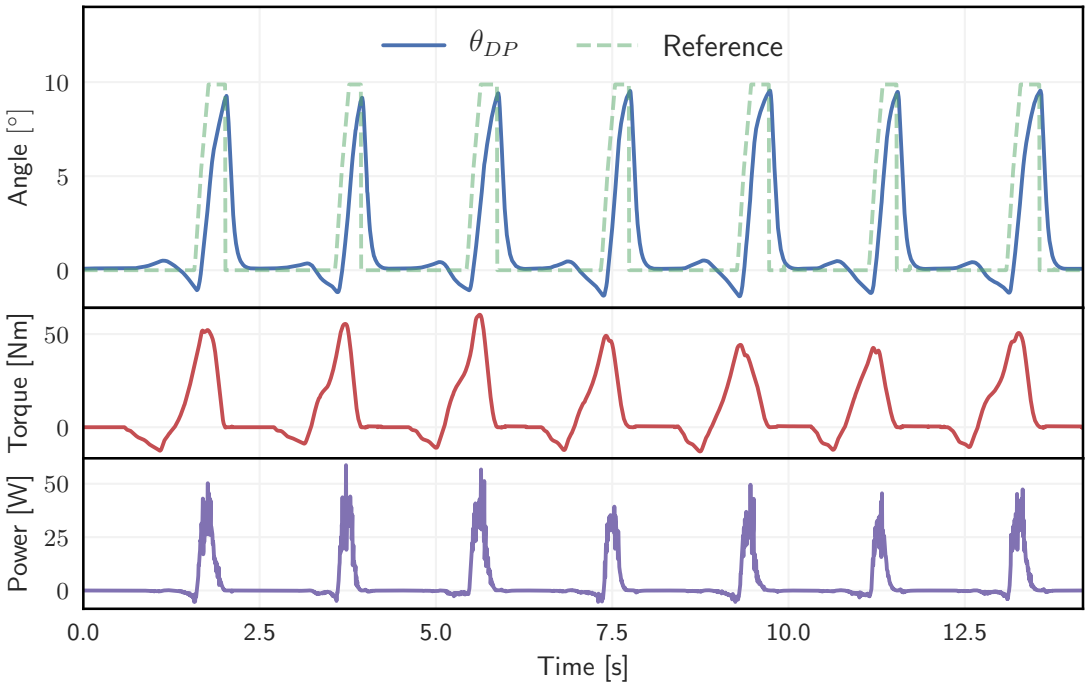
In total, the data from Moore et al. (2015) has fifteen participants. They collected over 25,000 gait cycles, which is why this is such a useful dataset for our study. We used subject six from the study, because he has the most similar physiology to our own test subject. In the study, each subject walked at three different speeds on an instrumented treadmill while they collected full body marker locations (i.e. MOCAP) and ground reaction loads from a pair of force plates, also located on the treadmill. The final protocol for the majority of the trials included two minutes of normal walking and eight minutes of walking under the influence of pseudo-random belt speed fluctuations (Moore et al., 2015). We only use the two-minutes of normal walking and leave the perturbed experiments for future work. The original study contains an excellent and elaborate 'method' section which outlines in detail how their experiments were performed. We now move onto how we processed the raw data, to extract a mean gait-cycle from normal walking, for three different speeds, used in our regression task.

The data was recorded with a 100Hz sampling rate, the dataset is cleaned such that missing values are interpolated using a variety of interpolation methods. Mostly though, the data was complete, and less than 0.5% were missing, hence little data in-painting was required. Following the procedure in (Moore et al., 2015) we compute gait events (toe off and heel strike times), basic 2D inverse kinematics and dynamics, to get the actual endogenous angles from exogenous marker trajectories. This pre-processing protocol was constructed so that each gait event was stored in an array, each on a separate row, such that the mean gait-cycle could be calculated, with uncertainty bounds as shown in fig. 5.8. We use the mean gait-cycle for all ours experiments, but note that it is fully possible to incorporate the uncertainty bounds in our analysis as well. However, given that in this chapter, we are only interested in a proof-of-concept, we leave that exercise for future work.

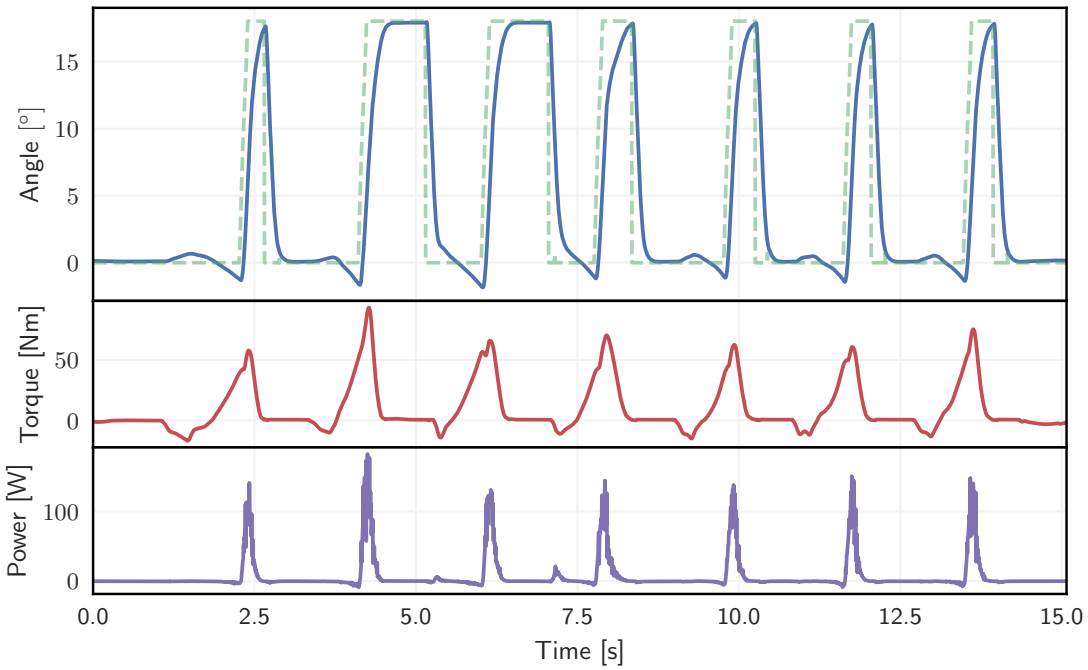
Hence, in this set of experiments, we revert to being parsimonious with our data: we use mean quantities unless otherwise specified.

5.6.5.2 Results

The main results are shown in fig. 5.30.



(a) $v_1 = 0.37\text{m/s};\ \text{RMSE} = 2.54°,\ R^2 = 0.47$



(b) $v_2 = 0.6\text{m/s},\ \text{RMSE} = 4.51°,\ R^2 = 0.67$

Figure 5.30: Experimental results of walking at two different speeds v_1 and v_2 with R^2 and RMSE measured for each trial. On the left hand side the walking speed is v_1 and the injected push off power is well below 50W. On the right hand side the walking speed is v_2 and the injected push off power is over 100W. Adapted from (Dhir et al., 2018).

Our results contrast well with one of the studies which most similar to ours, namely those of Lenzi et al. (2014b); Markowitz et al. (2011); Mai & Commuri (2016). Starting with Lenzi et al. (2014b), they integrate biologically accurate torque–angle curves (two curves sampled at 0.5m/s and 1.75m/s) into their controller, by encoding "a few speed-specific curves from able-bodied studies" (Lenzi et al., 2014b), and then interpolating (but *not* extrapolating) between them. The nature of their implementation means that they do not have uncertainty feedback incorporated into their controller. Nonetheless, whilst comparison is difficult (owing to the different nature of our experiments), similar root mean squared error (RMSE) and coefficient of determination (R^2) scores are recorded. We measure the metrics between the experimental values and the simulated values, for the same experiment. Joint angles are computed from the mathematical model in simulation, thus allowing us to track the error. Returning to Lenzi et al. (2014b), for their mid-stance to late-stance, they recorded an RMSE of 3.05° and R^2 of 0.7154 when subjects walked "on a treadmill at continuously varying walking speeds" (Lenzi et al., 2014b). Their reference is derived from able-bodied subjects, not a simulated one, so our R^2 are not quite comparable. Moreover, they use the full manifold regressed, whereas we are only using the push-off part in our simulations, to elicit realistic power, torque and angle profiles.

Another study, which bears comparing against, is the seminal study by Markowitz et al. (2011). They use a muscle-tendon model, which, conditioned on observations, produces estimates of the activation, force, length and velocity of the main muscles spanning the ankle. These are used derive control feedback loops that may (from a neural perspective) be critical in the control of those muscles during walking. This allows them to closely reproduce the muscle dynamics estimated from biological data. They produced similar results, to those we display in fig. 5.23. A rather simplistic result, but nonetheless one which suggest that our method is salient and concurrent with other state-of-the-art methods. 'Similar' here means that like us, they carry shape, size and placement of the torque-angle curves shown in fig. 5.23. Which is to say, their methods supports their hypothesis, as does ours: the curves should away from the centre with an increase or decrease in speed. Whilst we naturally cannot confirm that our model forms the underlying mechanism for this behaviour, but empirical evidence suggest that it supports that thesis. Further, they present results which demonstrate the prosthesis ankle and knee angles and torques, measured during their clinical trials, against those from a height and weight -matched

healthy subject. Though they do not provide RMSE or R^2 scores, inspection suggests that our method is comparable in speed adaptability. That being said, because we propose a probabilistic method, the variance on our predictions *may* reduce as the size of $\mathbf{X}$ grows, given that new samples increase our knowledge of the underlying state-space of the latent manifold which we are trying to estimate with our GPR model. This is not true for their comparable controller, which, although it is a tunable system, is not nonparametric.

Finally and more recently, a study was presented by Mai & Commuri (2016), which shows very impressive results. Like ourselves, Markowitz et al. (2011) and Lenzi et al. (2014b), they study transtibial (amputation between knee and ankle joint) amputees. They present an artificial neural network-based hierarchical controller that recognises the amputees' intent from measured gait observations. Once this is done the controller selects a displacement profile for the prosthetic joint based on the amputees' intent, and then adaptively "compensates for the un-modelled dynamics and disturbances for closed loop stability with guaranteed tracking performance" (Mai & Commuri, 2016). Effectively the result of their approach used gait-based quantities, collected from a group of nine transtibial amputees, to calculate an appropriate control torque for the recognised gait. Regrettably their results are only simulated, using their collected data. However their ankle plantarflexion angle tracking is impressive and comparable to our own results. That being said, whilst their tracking performance is impressive, their transition tracking could be improved see (Mai & Commuri, 2016, fig. 8). Not withstanding they present two challenging scenarios: one with artificial noise added, and one involving velocity changes to the profile that their controller is trying to track. This was achieved by a neural network was to learn the nonlinear ankle dynamics, and the interaction between the foot and the walking terrain was compensated by an empirical model of the ground reaction force (Mai & Commuri, 2016). To this end they receive a control torque which they apply to the prosthetic ankle joint to track the reference ankle displacement during gait. Now, earlier on we noted that their results were comparable to our own. This is under the proviso of merely estimating the error from their plots, as they have not presented RMSE or NLL scores of their method. However, as their goal is virtually identical to our own, they manage to track their reference trajectory 'better' than ourselves (still; only in simulation), in as much as the error can be estimated from their plots.

Finally, whilst we present competitive results to our peers, there are drawbacks to our method. Whilst Gaussian process regression is attractive for its many aforementioned properties, it also contains many drawbacks, and the way in which we have used the model, may not be realistic. Lets start with our primary assumption in this work: *gait-cycles are atomic*. This means that we posit that we can construct gait-cycles of arbitrary lengths, whose coordinates, be it exogenous or endogenous, we can pass as reference trajectories to a controller, which can then be tracked to elicit locomotion. The first form of criticism that can be levelled at this assumption, is that it could be a incorrect modelling assumption. Locomotion, from the body's point of view, could be a long-term planning process, which does not constitute atomic features of the nature that we propose. Indeed, we posit that the human onboard control system is Bayesian in that it stores local model of motion, such as the ones we demonstrate in this chapter, and deploys them according to user intent. For example, if the user wants to run 100m then the biological control system, adapts, online, to create gait-cycles that correspond to this user intent, with bounds placed by biomechanics, agility and resources (i.e. available energy). An alternate strategy suggests that the control system plans well in advance, the whole 100m stretch, in intricate detail, and the slightly adapts this principle during the run, with minor adjustments throughout. We will discuss more the computational and technical drawbacks of our method in the next section, hence for now lets focus on the main assumptions of this chapter.

Secondly, we have not covered *how* we propose to combine atomic gait-cycles to elicit natural looking locomotion. Whilst we have combined atoms to render longer sequences, these were merely concatenated, and where there was missing data, points were interpolated. This, as it turns out, was an adequate strategy, which did not present any significant problems, neither methodologically nor in the end result. Realistic however, it would be fair to suggest that were this model deployed on a real, commercial, prosthesis, procedures would be required for interpolating between atoms as they are deployed, based on user intent. This is not to say that it is not possible, it is, as we have demonstrated. It is rather to emphasise that we have yet to conduct incisive experiments into this area, hence it would be hard deign at this point what the optimal strategy is for concatenating atomic gait features online (or indeed offline).

How do humans transition? There is preciously few studies that actually consider *how* we move from one velocity to another, to take the simplest of examples. Consider a complex

one instead: how do we go from walking to climbing up a staircase – what is the dynamic response, once intent has been established? In this study, for simplicity, we assumed that the transition was linear. Hence, in our transition experiments above, we simply apply a linear transition from one velocity to another. It is highly improbable that this is the actual mechanism employed by the human body. This is simply because human motion is nonlinear, highly nonlinear, and thus it is also plausible to suggest that so too is human activity transition.

5.7 Discussion and conclusion

We have demonstrated the utility of using GP regression for finding what we term a *locomotion envelope*, which can serve to synthesise a high variety of locomotion for a powered ankle-foot prosthesis. There is a need to develop a control strategy, which can provide biologically sound torques across a wide range of walking speeds without requiring velocity-specific control tuning. The benefits of our approach, in particular, is its offline construction, thus making it is fast and robust (this does not mean that it can adapt online, but in its present incarnation it does not have to). More importantly, since we use supervised learning methods, we directly employ human demonstrations, hence achieving prosthesis locomotion that is natural looking (though we must stress that this is only yet at the simulation stage), and numerically consistent with experimental human locomotion – see fig. 5.23.

There are a number of issues to consider henceforth. First, while it is useful to have the capacity to arbitrarily switch and transition between velocities, we have not discussed intention estimation, or a specific perception layer of our method (we have alluded to the methods in chapter 4 and chapter 3 as one way of performing intention estimation). This was deliberate but it bears considering, given the medical nature of our chosen domain. First, in future work, in a clinical setting, we propose to place markers, in the sagittal plane, of the ankle and knee of the subject in order to measure their speed. Depending on which range they fall into, an appropriate speed will be selected by the controller. This, of course, is easy in such a controlled environment. We require high-level controls (Tucker et al., 2015), of which there are many, such as decision trees and finite state machines. These methods, albeit simple, are robust and operate on a set of identified rules, which

dictate when a mode-transition takes place. As input they could take the user-state (e.g. acceleration measurements from the healthy leg) or environmental queues such as frictional response of the walking surface. There are also promising methods to be found in more classical machine learning classifiers such as Gaussian mixture models (Varol et al., 2010) or support vector machines (Kilicarslan et al., 2013).

For Gaussian processes, the limiting factor with long-range predictions for all the manifolds, is the mean function posterior predictive uncertainties. We typically use a zero mean function, but there is ample evidence to suggest that using a more domain-specific mean function will allow us to make robust long-range predictions (i.e. inferring the manifold shape, magnitude and temporal evolution at speeds far away from the training data $s_*^- \ll s \ll s_*^+$ where s are the speeds present in the design matrix $\mathbf{X}$). Moreover, as we have already alluded to; kernel tuning is still required. This is a drawback, but necessary to incorporate our prior domain knowledge into the predictive framework. But we are rewarded with a smaller variance in our prediction should we tune the kernel accurately. This is something that should not be understated. The size of our variance implicitly gives us a *confidence level* of our prediction. This means that we can assign appropriate action to prediction results, depending on how certain we are of them. Equally, we can determine if our training database is sufficiently and appropriately broad for the range of motion, which we want the prosthesis to be able to undertake (Yun et al., 2014). Though we only discuss Gaussian processes in the context of nonlinear regression, there are many contenders for this role, such as nonlinear least squares regression, neural networks and support vector regression.

The second limiting factor that we mentioned at the start of the previous paragraph, is the predictive uncertainty – given our model, how sure can we be that its predictions are accurate? Osborne & Roberts (2007) discuss this problem in detail. We employ Bayesian probability to perform (Osborne & Roberts, 2007) inference about quantities which are unknown to us, or which we seek to increase our knowledge of. The GP paradigm is a good tool for achieving such information exploration. But even through principled model design, and considerable prior knowledge, we cannot avoid uncertainty. Osborne & Roberts (2007) and Rasmussen & Williams (2006) relate that this uncertainty can take many forms. For example, in real experiments data is typically: missing as a results of sensor failure; multiple sensors will often be correlated; there may be a complex noise which cannot be

assuaged by assuming a simple additive noise model and many others. All of these items play into our predictive uncertainty, which we can regularise in a few ways: by collecting more observations, to overcome sensor failure and eventually to overwhelm an misspecified prior (e.g. picking the wrong kernel). Better sensors are always desirable, but this is more subject to resources, than any explicit fault of the theory.

These are but general aspects of GP regression, for our setup we performed each surface-learning in isolation which is to say that no information was passed between manifolds during regression – they were all independent. We posit that by jointly learning the envelope, with a regression taking place over all constituent variables, with a global envelope training target, we would have received better posterior predictive performance. This could in the future be tackled by employing twin GPs (TGP) – a generic structured prediction method that uses GP priors on both covariates and responses. The TGP method models interdependencies between covariates, as in a typical GP, but also those between responses, hence correlations among both inputs and outputs are accounted for (Bo & Sminchisescu, 2010). More promising still is the notion of coregionalised Gaussian processes (Alvarez et al., 2012), as an extension to our locomotion envelopes. We discuss this further in the future work section of chapter 6.

In addition, it is possible, using neural networks, to learn the whole locomotion envelope in one go – see the work by Holden et al. (2015). Like other kernel methods, GPs are useful because they have a covariance functions whose hyperparameters can be learned by maximising the marginal likelihood (Barber, 2012) (as ever though, there is risk of overfitting). Given the values of their hyper-parameters, and this often allows a fine and precise trade-off between fitting the data and smoothing. As we demonstrated in eq. (2.93). While we are using a comparatively small number of data points to describe our variables over one-gait cycle, this is still a computationally viable approach. But one could imagine a scenario where there is a need to regress over several gait-cycles (to capture a long-term dependency, e.g. of a subject's particular gait), then the method needs to scale to accommodate this larger dataset. This, GPs cannot do well.

As noted by Rasmussen & Williams (2006); Deisenroth & Ng (2015) and Saatci (2012), the generic inference and learning algorithms for GPR has a runtime of $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$ memory complexity when N is the number of observations in $\mathbf{X}$. To be a bit more specific, prediction cost per test case is $\mathcal{O}(N^3)$ for the mean and $\mathcal{O}(N^2)$ for the variance

(Snelson, 2006; Rasmussen & Williams, 2006). This penalty results from using the Cholesky decomposition for finding the inverse of the covariance matrix. Recent work by, e.g. Wilson & Nickisch (2015) and Saatci (2012) demonstrate that sparse approximations to the full GPR problem, allow the latter to scale without unreasonable time and space penalties. Looking further afield there are alternatives, such as the modern version of neural-nets; deep-learning, which could also prove useful in the setting of nonlinear regression for manifold learning. The obvious drawback with these methods, as mentioned earlier, is that they will not yield uncertainty bounds on the predictions. We have not used our uncertainty bounds in this chapter, but we have the option of doing so, whereas a neural network for example, has a more involved methodology for producing error bars on the predictions.

Finally, we found information lacking in the precise nature of gait-cycle transition i.e. the way by which locomotion transitions occur between gait-cycles at different speeds. Whilst most current control methods divide the gait cycle into several sequential periods, each with independent controllers. It is not clear how precisely current switching modalities mimic those exhibited by human locomotion. Whilst we have taken the view in this study that there is large acceleration at the beginning of the gait-cycle (see fig. 5.17) required to accurately mimic human transition, Van der Noot et al. (2015) enforce a fixed time for each transitions in between speeds. Overall we have found the literature in this area wanting, thus revealing a need for further studies in that domain.

5.7.1 Conclusion

In this chapter, we have presented a data-driven control strategy for ankle-foot prostheses. We have demonstrated (by way of simulation and initial hardware tests) that the methodology has the capacity to allow the user to walk over a wide range of speeds, whilst also providing for fast variations outside of the training data. In future work, we intend to expand upon the speed range by adopting a more domain-specific mean function for GP regression as well as employing more advanced forms of information sharing in the GP framework.

Though this work is primarily intended for the rehabilitation robotics domain, it may prove insightful to the field of bipedal humanoids. Although we have implemented the work for

prostheses, the underlying theory concerns basic understanding of human locomotion, and how to adapt those insights to generalise our control strategies, to effect a singular, or desired set of locomotions. Further, we have demonstrated that taking a broadly cyclical view of human locomotion is useful since it means we can extrapolate over gait-cycles rather than full time-series data, across speeds. More importantly we have demonstrated a method for generating biologically *plausible* torque-angle curves.

As ever though, there is much room for improvement, and we have mentioned a few already. A large problem which we came upon several times during this work, was the difficulty in comparing with other powered prostheses control strategies, thus making it more difficult to quantitatively compare our computational simulations to those of our peers.

This was not merely an issue of implementing their methods. Because almost all studies in this area employ some form of training data (which is sensible, given that we are trying to mimic human physiological output), which they use for their controllers, they are usually loathe to share their data from particular studies. Most likely because it was expensive to obtain and is bound by institution-specific rules and regulations. However, the datasets that we do use in this study (Liu et al., 2008; Arnold et al., 2013; Moore et al., 2015) are not specifically designed for prostheses controller design. They are more concerned with muscle force generation, and modelling how the human body actually does this. This is a different goal to our own, where we are trying to artificially create that force generation and anthropomorphically pass it to the end effectors of our device. Certainly though, our goals are very much aligned, but an optimal dataset for our purposes would involve e.g. multiple subjects walking at multiple speeds (> 10), for several hours each. This would no doubt be expensive to collect.

Moreover, there are issues with scaling these datasets. Because we suppose that there are latent, common dynamics, to human gait, the observations that we do receive have to appropriately scaled – a plantarflexion angle for a child will be different to that of an adult. There is no standard protocol to receive nondimensional metrics of curves, and many studies employ their own for various quantities. For example, Liu et al. (2008) scales the velocities in their study by $\sqrt{gL_{\mathsf{leg}}}$ where g is gravitational acceleration and L_{leg} is the measured length of a subject's leg. This is good approach to nondimensionality, however not one that can be carried across to the study by Moore et al. (2015), since they do not measure the length of participants' legs. Consequently, as there is as of yet no large body

of work concerning control schemes for speed-variations for prostheses (granted, a rather niche topic), common metrics will have to designed for *what* data needs to be collected for robust control design, and *how* it is collected so that it can easily be *shared* with the wider community. For now though, to quote Moore et al. (2015):

> Even though years of data on thousands of subjects now exist, this data is not widely disseminated, well organized, nor available with few or no restrictions.

However, as the area matures, especially with bipedal robots, datasets will surely (we say with optimistic hesitation) become more widely available.

A final though on the future; we plan to implement the proposed methodology on a powered two DoF ankle-foot prosthesis test-bed, and test the method under various loading and speeds. An experimental test rig with a circular treadmill is available to the study, that will be used to provide insights on how to improve the limited adaptability of existing powered prostheses.

CHAPTER 6

Conclusion and future directions

In this section we provide a summary of the main contributions in each chapter, and offer thoughts and ideas for future directions.

This thesis has presented several application-driven tools and models, founded on the principles of time-series analysis. We started off with a simpler exercise in chapter 3 in which we demonstrated that standard state-space modelling with the Kalman filter, and dimensionality reduction via the MLDA, are adequate tools to sufficiently raise the bar on the assigned classification task. The chapter demonstrated that certain problems do not require novel models, merely a better combination of current ones, applied in a way more suitable to the task in hand. At the same time, we demonstrated one of the main problems with parametric methods, namely the need to a priori fix the number of classes available in the observations. This issue we investigated further in chapter 4.

Being able to analyse observations without the limiting factor of pre-specifying the state cardinality, is a feature whose utility cannot be understated. From statistical machine translations to language modelling, Bayesian nonparametrics has found widespread use. Hence, in chapter 4 we sought to extend its use further by (1) developing novel state-space models for time-series analysis and (2) deliver inference via the probabilistic programming paradigm. We demonstrated three novel models: the stateful HDP-HMM, the IDHMM and the stateful IDHMM. All models were empirically evaluated on a number of challenging synthetic and real segmentation tasks. Approximate inference (sequential Monte Carlo, particle Gibbs and particle independent Metropolis-Hastings) was used to find the posterior estimate of the state sequence in each task, where in some instances we also employed

Bayesian optimisation to find good model parameters. We continued with Bayesian nonparametric methods in chapter 5 where we focused on control.

One of the simplest functions that healthy subjects take for granted is the ability to decelerate and accelerate at will. This function is not easily restored to people with mobility problems, particularly those whose locomotion is aided by a powered prosthesis. In chapter 5 we demonstrated a control methodology that relies on Gaussian process regression as well as impedance control. The latter ingredient finds the multivariate manifolds on which gait-cycle variates live, over different speeds. By allowing a mapping of that field, conditioned from human motion-capture observations, we were able to provide a reference trajectory for the prosthesis, for any self-selected velocity within the test-range.

Incidence identification

In chapter 3 we took the well cited study by Luštrek & Kaluža (2009), one step further. In that piece of work the authors' classification performance on an activity recognition task, focusing primarily on detecting fall events, was impressive but could be improved. To that end we applied standard methods from state-space modelling and dimensionality reduction, and applied the same classification schemes as employed by Luštrek & Kaluža (2009), to demonstrate a substantial classification accuracy increase. We also went a step further than the authors and investigated classification accuracy across multiple points on the test subjects' bodies. No one location was found to possess substantially more discriminatory information compared to the others points.

Dynamics identification via time-series segmentation

In chapter 4 we introduced three new Bayesian nonparametric state-space models: the stateful HDP-HMM, the IDHMM and the stateful IDHMM. All models were tested on a number of segmentation tasks, real and synthetic. It was found that the stateful IDHMM performed best on the synthetic task when compared alongside the proposed models but also the HDP-HMM and the sticky HDP-HMM. We also demonstrated increased segmentation performance when comparing to the HDP-HMM, the sticky HDP-HMM and the stateful HDP-HMM, whilst simultaneously using Bayesian optimisation for finding

optimal model parameters. This combination of inference and optimisation was applied to two human activity recognition tasks where it was shown that nonparametric state-space models performed comparatively well to hand-labelled segmentation. Without using Bayesian optimisation we applied all aforementioned models to the PAMAP2 human activity dataset and the study of lion ecology with fuzzy ground-truth. We showed that these models could indeed be used in this complex domain, but ascertaining which one is 'best' remains an open question, primarily because it depends on which question is being asked. Instead of writing bespoke samplers for each model, we demonstrated that probabilistic programming could be of high value when iterating over novel models, to find one which is best suited to the problem domain.

Gaussian process regression for prosthesis control

In chapter 5 we combined Gaussian process regression with impedance control, to effect a robust and adaptive control scheme for prosthesis control. By framing the current state-of-the-art in this area it was clear that current methods could benefit from using Bayesian nonparametrics to take into account not just uncertainty, but also the ability to grow the model complexity with more observations. For prostheses, especially the latter function, means that prostheses controllers will become better, and can learn new tasks, with time and more observations. This functionality currently has to be encoded in the controllers which do not make use of Gaussian processes. Through synthetic and empirical experiments, we demonstrated the key utility which we sought from our design, namely the ability to smoothly transition between self-selected velocities. In the empirical case we tested this by seeking to match the power output of our prosthesis to that of a reference trajectory, for the same task.

6.1 Directions for future research

What this thesis has not addressed is how to combine the methodologies proposed within. As we opened with a reference to the review paper by Tucker et al. (2015), it is also appropriate that we close with that same paper. Ultimately we want a control system that can seamlessly integrate with any new user, but from thereon continue to learn. Not just about the user's gait patterns, gait cycle and locomotion variates, but also about the

operating environment as well as specific user incidents like stair climbing or walking whilst carrying a heavy load. All such events call for an adaptive general control framework.

The generalised control framework proposed by Tucker et al. (2015) is represented as a three-tier hierarchical architecture, which they say closely resembles the structure and functionality of the human central nervous system. Therein our methods firmly fall within the high-level tier or the *perception* layer and, to some extent, the middle *translation* layer, where we map from intent to state. Though we have made steps to fill the gaps in this structure, unsurprisingly there are currently no control schemes that fit the proposed general framework. However, a multitude of studies are filling in the missing pieces. Garate et al. (2016) present a study which is one of the furthest along in establishing a fully cooperative rehabilitation robot. But the enormity of the task means that it will take some time before all pieces are fully connected and deployed, as they are dependent on major advances in machine learning, control theory, wearable sensor technologies and portable computational resources (Tucker et al., 2015).

⋆⋆⋆

Here we suggest some other strands of research that build upon the ideas presented in this thesis, and others which are at a stage of fruition. Although there has been a large focus on human-centric data in this thesis, the following directions can be applied generally too.

Coregionalised locomotion envelopes

‘Sharing of statistical strength’ is a phrase often employed in machine learning and signal processing. In sensor networks, for example, missing signals from certain sensors may be predicted by exploiting their correlation with observed signals acquired from other sensors (Alvarez et al., 2012). For humans, our hands move synchronously with our legs. We can exploit these implicit correlations for predicting new poses and for generating new natural-looking walking sequences. We can also go much further and exploit this form of transfer learning, to develop new control schemas for robust control of rehabilitation robots. In this final section we introduce *coregionalised locomotion envelopes* – a method for multi-dimensional manifold regression, on human locomotion variates.

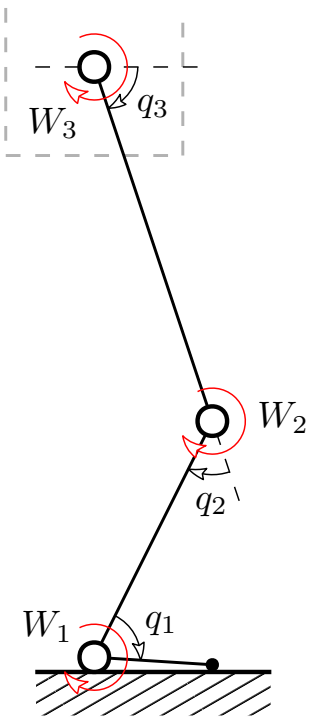


Figure 6.1: Rigid body diagram for lower-body functions, seen here in the sagittal plane.

We have previously considered novel control strategies for powered ankle-foot prostheses, using a data-driven approach which employs a combination of Gaussian processes regression and impedance control. Therein we learned the nonlinear functions which dictate how locomotion variables temporally evolve using the aforementioned nonparametric method, and regress that surface over several velocities to create a manifold, per variable. The joint set of manifolds, as well as the temporal evolution of the gait-cycle duration is what we termed a locomotion envelope.

The problem with our initial approach is that it did not consider the dependency (nor correlation), of the input training variables, nor the output training variables. Instead, assuming that regression variates are independent (strictly I.I.D.). Our initial approach is not an uncommon first pass. Indeed, consider the remarks by Mai & Commuri (2016) on the topic of commercially available below-knee controlled prostheses:

> In general, control algorithms neglect the dynamics of the ankle joint, the interaction of the ankle with the remaining healthy joints of the residual limb, and the effect of the ground reaction torque. These devices are based on the linearized dynamics of the joint and use proportional-derivative control with fixed control gains.

Given the strongly dependent nature of human locomotion, in everything from the cyclical nature of exogenously measured metrics (such as the angular velocity of the knee-angle – see fig. 6.1), to the proprioceptive nature of our indodgenous control mechanisms, there is ample evidence to suggest that control variates *are* dependent. This strong assumption will form the basis for this future direction; a sequel to our initial contribution which takes into account human locomotion correlates.

The issue which we seek to tackle is similar to the *inverse dynamics problem* for robotic manipulators. In that domain we seek to compute the torques $\boldsymbol{\tau}$ required at the joints, to drive a manipulator along some given trajectory. That trajectory could e.g. be specified as the temporal evolution of the joint-angles; $\mathbf{q}(t)$, velocities; $\dot{\mathbf{q}}(t)$ and accelerations; $\ddot{\mathbf{q}}(t)$. See fig. 6.1 wherein $\mathbf{q} \triangleq [q_1, q_2, q_3]^\mathsf{T}$. In that familiar problem, it then becomes a quest to a find a model for $\boldsymbol{\tau}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q})$. As Williams et al. (2009) note, this is hard enough in robotics proper, in our domain it is harder still. Not only do we have to contend with the uncertainty of the physical parameters of the robot (or in our case; the active prosthesis – not to mention the dynamical properties of the subject whose mass e.g. will change on a daily basis), we also need to model such things as human-robot interactions, ground friction, changing operational environments, human adaptability to the device and long-term rehabilitation – to mention but a few. Instead of torque, our dependent variable is the work $\mathbf{W} \triangleq [W_1, W_2, W_3]^\mathsf{T}$, resulting from the velocity of locomotion v. Where, to echo the study by Williams et al. (2009), a *context* in our study refers to the different velocities we seek to regress over, such that the inverse dynamics function depends on the different contexts. To find this field of functions we can turn to GPR, but this time for vector valued outputs as shown by the graphical model in fig. 6.2.

Combining chaotic embeddings with variational autoencoders

Whilst we managed to demonstrate that certain discriminator information lives on a low-dimensional space, we did not consider the temporal evolution of that feature. This type of dynamic latent variable model (such as the GPDM (Wang, 2005) or the GPLVM (Lawrence, 2003)) have been used before to study high-dimensional time-series observations. But, some temporal evolutions are fiendishly difficult to reason about, let alone make predictions of (certainly not predictions that lend themselves to being treated as accurate). It is only recently that *chaotic* complex natural phenomena have become amenable to

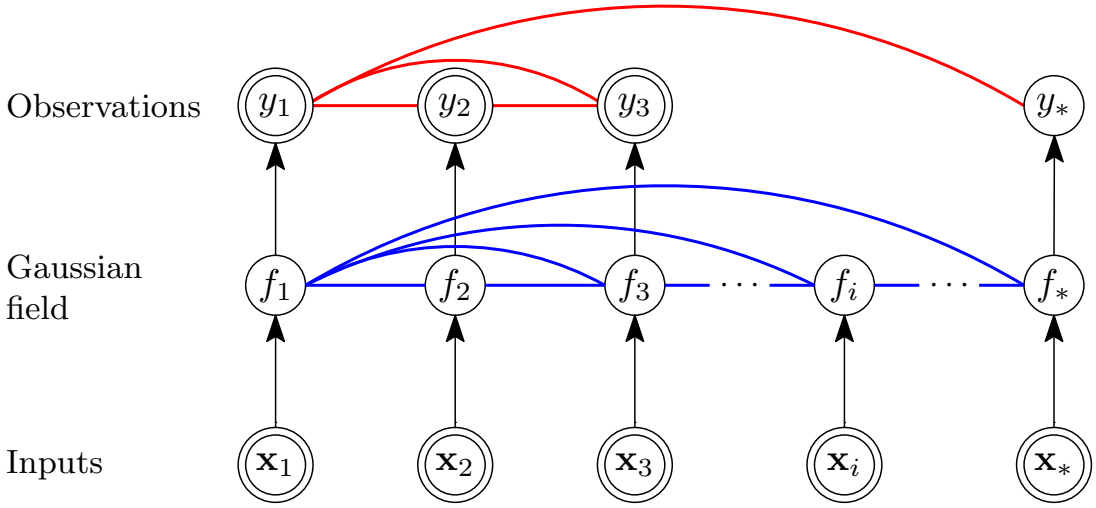


Figure 6.2: Graphical model of the coregionalised Gaussian process. All hidden nodes $f_i = f(\mathbf{x}_i)$ are interconnected by undirected edges forming the Gaussian process. For brevity, and to avoid clutter, we only show conditional dependencies of f_1 s.t. $f_1 \mid f_2, f_3, f_i, f_*$, the same for the first output.

analysis, and in some cases, even prediction (Ruelle, 1990, §1). Chaos has structure, and is more than just noise. Because there is structure, that also means that there are numerous potential engineering applications of sources of chaotic time-series (Abarbanel et al., 1993, §2.2), which could take advantage of that structure. This means that we are at liberty to both predict and control the sources of the chaotic observations, which is why this is a worthwhile direction to pursue.

The field of studies which has grown out of this work has rather unhelpfully been baptised to what we now refer to as 'chaos'. Whilst a deterministic system is one whose present state is in principle determined by its initial condition, a chaotic one labours under no such restrictions. With the advent of greater computational power, machine learning algorithms, we stand now at the precipice; our data is abundant, but the governing equations of nature remain as of yet elusive. Combine the two however, and we enter the era of "data-driven discovery of dynamics" (Brunton et al., 2016, §1).

Hence we can combine elements of chaos theory and deep learning, especially by employing the autoencoder formalism, reviewed in chapter 2. Specifically by combining lessons from embedding theory and variational autoencoders, to derive representations of time-varying nonlinear dynamical systems, responsible for generating many complex natural phenomena, manifested through spatio-temporal patterns. By employing a Bayesian encoder-decoder assumption on part of the phase-space embedding, it is possible to

reconstruct the observations as our objective, whilst also keeping the embedding parameters low-dimensional through priors. Early results suggest that we can reconstruct a number of canonical chaotic systems (such as the Lorenz attractor), for further details see (Dhir et al., 2017b).

Bayesian system identification of operational space controllers

The theory of operational space control (OSC) is one of the most elegant approaches to task control due to its potential for dynamically consistent control, compliant control and many other favourable properties, with applications from end-effector control of manipulators up to balancing and gait execution for humanoid robots (Peters & Schaal, 2008). Indeed, from a purely theoretical stand-point it remains the most advanced control architecture for redundant robots (Vuong et al., 2010; Khatib, 1987). But a fundamental problem with OSC and indeed general control theory, is to learn a model of a system from observations that is useful for controller synthesis (Ross & Bagnell, 2012). In so doing we are try to solve the system identification problem.

In chapter 4 we introduced Bayesian nonparametric state-space models, and while state-space models have been used for system identification (Schön et al., 2015), in an approximate inference framework, only recently (Eleftheriadis et al., 2017) has that been extended to Bayesian nonparametric state-space models. It would result in an attractive fusion of ideas because it would enable the robot to learn a new control model if the machine changed in some way or form. Normally this would require the whole control scheme to be re-written accounting for an updated set of e.g. kinematics.

Having a prosthesis which is able to update its control parameters based on the user, would make for a truly adaptive AAM, particularly if OSC is used. More importantly having a prosthesis that has an unbounded state-space would also allow it to store internal models of specific dynamic events (such as walking up a staircase) where one set of control parameters, could result in reduced use of metabolic energy. For the same purpose there is also space for Bayesian optimisation which could be used to optimise the prosthesis to that target, for a specific user and for a specific task.

Bibliography

Abarbanel, H. D., Brown, R., Sidorowich, J. J., and Tsimring, L. S. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65(4):1331, 1993.

Abdi, H. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):97–106, 2010.

Aertbeliën, E. and De Schutter, J. Learning a predictive model of human gait for the control of a lower-limb exoskeleton. In *International Conference on Biomedical Robotics and Biomechatronics*, 2014.

Airoldi, E. M., Blei, D., Erosheva, E. A., and Fienberg, S. E. *Handbook of mixed membership models and their applications*. CRC Press, 2014.

Albert, M. V., Kording, K., Herrmann, M., and Jayaraman, A. Fall classification by machine learning using mobile phones. *PloS one*, 7(5), 2012.

Aldous, D. J. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*, pp. 1–198. Springer, 1985.

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

Anava, O., Hazan, E., and Zeevi, A. Online time series prediction with missing data. In *International Conference on Machine Learning*, pp. 2191–2199, 2015.

Andrieu, C., Doucet, A., Singh, S. S., and Tadic, V. B. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004.

Andrieu, C., Doucet, A., and Holenstein, R. Particle markov chain monte carlo for efficient numerical simulation. *Monte Carlo and quasi-Monte Carlo methods 2008*, pp. 45–60, 2009.

Andrieu, C., Doucet, A., and Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Anjyo, K. and Lewis, J. Rbf interpolation and gaussian process regression through an rkhs formulation. *Journal of Math-for-Industry*, 3(6):63–71, 2011.

Argall, B. D. Machine Learning for Shared Control with Assistive Machines. In *ICRA Workshop on Autonomous Learning: From Machine Learning to Learning in Real-world Autonomous Systems*. ICRA, 2013.

Arnold, E. M., Hamner, S. R., Seth, A., Millard, M., and Delp, S. L. How muscle fiber lengths and velocities affect muscle force generation as humans walk and run at different speeds. *Journal of Experimental Biology*, 216(11):2150–2160, 2013.

Au, S., Berniker, M., and Herr, H. Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits. *Neural Networks*, 21(4):654–666, 2008.

Barber, D. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

Bates, D. M. and Watts, D. G. *Nonlinear regression analysis and its applications*, volume 2. Wiley Online Library, 1988.

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Beal, M. and Krishnamurthy, P. Gene expression time course clustering with countably infinite hidden markov models. *arXiv preprint arXiv:1206.6824*, 2012.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. The infinite hidden markov model. In *Advances in Neural Information Processing Systems*, pp. 577–584, 2001.

Bengio, Y. et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Bilmes, J. A. What HMMs can do. 89(3):869–891, 2006.

Bingham, E. and Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 245–250. ACM, 2001.

Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.

Blackwell, D. and MacQueen, J. B. Ferguson distributions via pólya urn schemes. *The Annals of Statistics*, pp. 353–355, 1973.

Blais, E., Brody, J., and Ghazi, B. The information complexity of hamming distance. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 28. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

Blake, A. and Isard, M. The condensation algorithm-conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems*, pp. 361–367, 1997.

Blunsom, P. and Cohn, T. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 865–874. Association for Computational Linguistics, 2011.

Bo, L. and Sminchisescu, C. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, 2010.

Bostrom, N. *Superintelligence: Paths, dangers, strategies*. Oxford University Press, 2014.

Bourke, A. and Lyons, G. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics*, 30(1):84–90, 2008.

Bourke, A., O'brien, J., and Lyons, G. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, 26(2):194–199, 2007.

Brochu, E., Cora, V. M., and De Freitas, N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Brown, D. D., Kays, R., Wikelski, M., Wilson, R., and Klimley, A. P. Observing the unwatchable through acceleration logging of animal behavior. 1(1):1, 2013.

Brunton, S. L., Brunton, B. W., Proctor, J. L., Kaiser, E., and Kutz, J. N. Chaos as an intermittently forced linear system. *arXiv preprint arXiv:1608.05306*, 2016.

Bull, A. D. Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904, 2011.

Cahill, N. D. Normalized measures of mutual information with general definitions of entropy for multimodal image registration. In *Biomedical Image Registration*, pp. 258–268. Springer, 2010.

Calandra, R., Gopalan, N., Seyfarth, A., Peters, J., and Deisenroth, M. P. Bayesian gait optimization for bipedal locomotion. In *International Conference on Learning and Intelligent Optimization*, pp. 274–290. Springer, 2014.

Calandra, R., Peters, J., Rasmussen, C. E., and Deisenroth, M. P. Manifold gaussian processes for regression. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 3338–3345. IEEE, 2016a.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23, 2016b.

Carroll, G., Slip, D., Jonsen, I., and Harcourt, R. Supervised accelerometry analysis can identify prey capture by penguins at sea. 217(24):4295–4302, 2014.

Chan, S., Yao, B., Gao, W., and Cheng, M. Robust impedance control of robot manipulators. *International Journal of Robotics & Automation*, 6(4):220–227, 1991.

Cheng, C.-A., Huang, T.-H., and Huang, H.-P. Bayesian human intention estimator for exoskeleton system. In *Advanced Intelligent Mechatronics (AIM), IEEE/ASME*, pp. 465–470. IEEE, 2013.

Chib, S. and Greenberg, E. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

Choudhry, T. and Wu, H. Forecasting ability of garch vs kalman filter method: evidence from daily uk time-varying beta. *Journal of Forecasting*, 27(8):670–689, 2008.

Cohen, W., Ravikumar, P., and Fienberg, S. A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pp. 73–78, 2003.

Collins, S. H. *Dynamic walking principles applied to human gait*. PhD thesis, University of Michigan, 2008.

Crooks, G. E. On measures of entropy and information. *Tech. Note*, 9:v4, 2017.

Deisenroth, M. P. and Ng, J. W. Distributed gaussian processes. *arXiv preprint arXiv:1502.02843*, 2015.

Dekel, O. and Shamir, O. Multiclass-multilabel classification with more classes than examples. In *International Conference on Artificial Intelligence and Statistics*, pp. 137–144, 2010.

Dewar, M., Wiggins, C., and Wood, F. Inference in hidden markov models with explicit state duration distributions. 19(4):235–238, 2012.

Dhir, N. and Wood, F. Improved activity recognition via Kalman smoothing and multiclass linear discriminant analysis. In *Proceedings of the Engineering in Medicine and Biology Society (EMBC), IEEE*, pp. 582–585. IEEE, 2014.

Dhir, N., Perov, Y., Wijers, M., Wood, F., Markham, A., Trethowan, P., du Preez, B., Loveridge, A., and Macdonald, D. Tracking african lions with nonparametric hierarchical models using probabilistic programming. In *Proceedings of the International Society of Bayesian Analysis (ISBA)*, 2016a.

Dhir, N., Perov, Y., and Wood, F. Nonparametric Bayesian models for unsupervised activity recognition and tracking. In *Intelligent Robots and Systems (IROS), IEEE/RSJ*, pp. 4040–4045. IEEE, 2016b.

Dhir, N., Dallali, H., and Rastgaar, M. Coregionalised Locomotion Envelopes. In *Proceedings of the Adaptive Control Methods in Assistive Technologies Workshop, IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, 2017a.

Dhir, N., Kosiorek, A. R., and Posner, I. Bayesian Delay Embeddings for Dynamical Systems. In *NIPS Timeseries Workshop*, 2017b.

Dhir, N., Vakar, M., Markham, A. C., Wijers, M., Wood, F., Trethowan, P., Du Preez, B., Loveridge, A., and Macdonald, D. Interpreting lion behaviour with nonparametric probabilistic programs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017c.

Dhir, N., Dallali, H., Ficanha, E. M., Ribeiro, G. A., and Rastgaar, M. Locomotion Envelopes for Adaptive Control of Powered Ankle Prostheses. In *Robotics and Automation (ICRA)*. IEEE, 2018.

Diaconis, P. Finite forms of de finetti's theorem on exchangeability. *Synthese*, 36(2):271–281, 1977.

Diaz, A., Prado, M., Roa, L., Reina-Tosina, J., and Sánchez, G. Preliminary evaluation of a full-time falling monitor for the elderly. In *Engineering in Medicine and Biology Society, IEEE*, volume 1, pp. 2180–2183. IEEE, 2004.

Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

Doucet, A., De Freitas, N., and Gordon, N. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pp. 3–14. Springer, 2001a.

Doucet, A., De Freitas, N., and Gordon, N. *Sequential Monte Carlo methods in practice*. Springer, 2001b.

Doughty, K., Lewis, R., and McIntosh, A. The design of a practical and reliable fall detector for community and institutional telecare. *Journal of Telemedicine and Telecare*, 6(suppl 1): 150–154, 2000.

Dounskaia, N. Control of human limb movements: the leading joint hypothesis and its practical applications. *Exercise and sport sciences reviews*, 38(4):201, 2010.

Duckworth, D. pykalman, December 2012. URL http://pykalman.github.io/. Accessed: 11/03/2017.

Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., and Courville, A. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Durbin, J. and Koopman, S. J. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

Duvenaud, D. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Duvenaud, D. K., Lloyd, J. R., Grosse, R. B., Tenenbaum, J. B., and Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. In *ICML (3)*, pp. 1166–1174, 2013.

Eilenberg, M. F., Geyer, H., and Herr, H. Control of a powered ankle–foot prosthesis based on a neuromuscular model. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(2):164–173, 2010.

Eleftheriadis, S., Nicholson, T. F., Deisenroth, M. P., and Hensman, J. Identification of gaussian process state space models. *arXiv preprint arXiv:1705.10888*, 2017.

Eriksson, K., Estep, D., and Johnson, C. *Applied Mathematics: Body and Soul: Volume 1: Derivatives and Geometry in IR3*. Springer Science & Business Media, 2013.

Farag, A. A. and Elhabian, S. Y. A tutorial on data reduction: Linear discriminant analysis (LDA). Technical report, University of Louisville, October 2008.

Farris, D. J. and Sawicki, G. S. The mechanics and energetics of human walking and running: a joint level perspective. *Journal of The Royal Society Interface*, 2011.

Fearnhead, P. Modern Computational Statistics: Alternatives to MCMC, 2012. URL http://www.maths.lancs.ac.uk/~fearnhea/GTP/GTP_Slides.pdf. Accessed: 22/09/2015.

Fearnhead, P. and Meligkotsidou, L. Augmentation schemes for particle mcmc. *arXiv preprint arXiv:1408.6980*, 2014.

Ferguson, T. S. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pp. 209–230, 1973.

Ferreira, C., Reis, L. P., and Santos, C. P. Review of control strategies for lower limb prostheses. In *Robot 2015: Iberian Robotics Conference*, pp. 209–220. Springer, 2016.

Ficanha, E. M., Ribeiro, G. A., Dallali, H., and Rastgaar, M. Design and preliminary evaluation of a two dofs cable-driven ankle–foot prosthesis with active dorsiflexion–plantarflexion and inversion–eversion. *Frontiers in Bioengineering and Biotechnology*, 4, 2016.

Fisette, P. and Samin, J. Robotran: Symbolic generation of multi-body system dynamic equations. In *Advanced Multibody System Dynamics*, pp. 373–378. Springer, 1993.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

Fleet, D. J. Motion Models for People Tracking. In *Visual Analysis of Humans: Looking at People*, chapter 10. Springer, 2011.

Flowers, W. C. and Mann, R. W. An electrohydraulic knee-torque controller for a prosthesis simulator. *Journal of Biomechanical Engineering*, 99(1):3–8, 1977.

Fox, E., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. Bayesian nonparametric inference of switching dynamic linear models. *Transactions on Signal Processing, IEEE*, 59(4):1569–1585, 2011.

Fox, E. B., Sudderth, E. B., and Willsky, A. S. Hierarchical dirichlet processes for tracking maneuvering targets. In *Information Fusion*, pp. 1–8. IEEE, 2007.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. An HDP-HMM for systems with state persistence. In *Proceedings of the International Conference on Machine learning*, pp. 312–319. ACM, 2008.

Fox, E. B. *Bayesian nonparametric learning of complex dynamical phenomena.* PhD thesis, Massachusetts Institute of Technology, 2009.

Friedland, B. *Control system design: an introduction to state-space methods.* Courier Corporation, 2012.

Frigola, R., Lindsten, F., Schön, T. B., and Rasmussen, C. E. Bayesian inference and learning in gaussian process state-space models with particle mcmc. In *Advances in Neural Information Processing Systems*, pp. 3156–3164, 2013.

Fulton, J., Bitmead, R. R., and Williamson, R. C. Smoothing approaches to reconstruction of missing data in array processing. In *Defence Applications of Signal Processing. Proceedings of the US/Australia Joint Workshop on Defence Applications of Signal Processing. New York: Elsevier*, 2001.

Garate, V. R., Parri, A., Yan, T., Munih, M., Lova, R. M., Vitiello, N., and Ronsse, R. Walking assistance using artificial primitives: a novel bioinspired framework using motor primitives for locomotion assistance through a wearable cooperative exoskeleton. *IEEE Robotics & Automation Magazine*, 23(1):83–95, 2016.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.

Genin, J. J., Bastien, G. J., Franck, B., Detrembleur, C., and Willems, P. A. Effect of speed on the energy cost of walking in unilateral traumatic lower limb amputees. *European Journal of Applied Physiology*, 103(6):655, 2008.

Geyer, H. and Herr, H. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(3):263–273, 2010.

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*, pp. 493–499, 1998.

Goldwater, S. and Griffiths, T. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 744–751, 2007.

Gonczarek, A. and Tomczak, J. M. Manifold regularized particle filter for articulated human motion tracking. In *Advances in Systems Science*, pp. 283–293. Springer, 2014.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Goodman, N. D. and Stuhlmüller, A. The Design and Implementation of Probabilistic Programming Languages, 2014. URL http://dippl.org. [Accessed on: 15-09-2015].

Gordon, A. D., Henzinger, T. A., Nori, A. V., and Rajamani, S. K. Probabilistic programming. In *Proceedings of the on Future of Software Engineering*, pp. 167–181. ACM, 2014.

Gordon, N. J., Salmond, D. J., and Smith, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing*, volume 140, pp. 107–113. IET, 1993.

GPyOpt. Gpyopt: A bayesian optimization framework in python, 2016. URL http://github.com/SheffieldML/GPyOpt. Accessed: 02/02/2016.

Grimes, D., Flowers, W., and Donath, M. Feasibility of an active control scheme for above knee prostheses. *Journal of Biomechanical Engineering*, 99(4):215–221, 1977.

Grimmer, M. *Powered lower limb prostheses*. PhD thesis, Technische Universität, 2015.

Gritli, H., Belghith, S., and Khraief, N. Ogy-based control of chaos in semi-passive dynamic walking of a torso-driven biped robot. *Nonlinear Dynamics*, 79(2):1363–1384, 2015.

Grosse, R., Salakhutdinov, R., Freeman, W., and Tenenbaum, J. Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*, 2012.

Grünewälder, S., Broekhuis, F., Macdonald, D. W., Wilson, A. M., McNutt, J. W., Shawe-Taylor, J., and Hailes, S. Movement activity based classification of animal behaviour with an application to data from cheetah (acinonyx jubatus). 7(11):e49120, 2012.

Gu, B., Sheng, V. S., Wang, Z., Ho, D., Osman, S., and Li, S. Incremental learning for ν-support vector regression. *Neural Networks*, 67:140–150, 2015.

Gutzwiller, K. J., Wiedenmann, R. T., Clements, K. L., and Anderson, S. H. Effects of human intrusion on song occurrence and singing consistency in subalpine birds. pp. 28–37, 1994.

Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011a.

Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011b.

Hargrove, L. J., Simon, A. M., Young, A. J., Lipschutz, R. D., Finucane, S. B., Smith, D. G., and Kuiken, T. A. Robotic leg control with emg decoding in an amputee with nerve transfers. *New England Journal of Medicine*, 369(13):1237–1242, 2013.

Hecht-Nielsen, R. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence: Imitating Life, IEEE Press*, pp. 43–56, 1994.

Heger, H., Millstein, S., and Hunter, G. Electrically powered prostheses for the adult with an upper limb amputation. *Bone & Joint Journal*, 67(2):278–281, 1985.

Heller, K. A., Teh, Y. W., and Görür, D. Infinite hierarchical hidden markov models. In *International Conference on Artificial Intelligence and Statistics*, pp. 224–231, 2009.

Herr, H. M. and Grabowski, A. M. Bionic ankle–foot prosthesis normalizes walking gait for persons with leg amputation. In *Proc. R. Soc. B*, volume 279, pp. 457–464. The Royal Society, 2012.

Hinrichsen, D. and Pritchard, A. J. *Mathematical systems theory I: modelling, state space analysis, stability and robustness*, volume 48. Springer Berlin, 2005.

Hitt, J. K., Sugar, T. G., Holgate, M., and Bellman, R. An active foot-ankle prosthesis with biomechanical energy regeneration. *Journal of Medical Devices-Transactions of the Asme*, 4 (1), 2010.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Hogan, N. and Buerger, S. P. Impedance and interaction control, robotics and automation handbook, 2005.

Hogan, N. Impedance control: An approach to manipulation. In *American Control Conference*, pp. 304–313. IEEE, 1984.

Hol, J. D., Schon, T. B., and Gustafsson, F. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82. IEEE, 2006.

Holden, D., Saito, J., Komura, T., and Joyce, T. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia Technical Briefs*, pp. 18. ACM, 2015.

Holgate, M. A., Bohler, A. W., and Suga, T. G. Control algorithms for ankle robots: A reflection on the state-of-the-art and presentation of two novel algorithms. In *Biomedical Robotics and Biomechatronics, IEEE*, pp. 97–102. IEEE, 2008.

Hong, J., Chun, C., and Kim, S.-J. Gaussian process gait trajectory learning and generation of collision-free motion for assist-as-needed rehabilitation. In *Humanoid Robots (Humanoids), IEEE-RAS*, pp. 181–186. IEEE, 2015.

Hu, N., Lou, Z., Englebienne, G., Kröse, B., et al. Learning to recognize human activities from soft labeled data. 2014.

Hu, N., Englebienne, G., Lou, Z., and Krose, B. A hierarchical representation for human activity recognition with noisy labels. In *Intelligent Robots and Systems (IROS), IEEE/RSJ*, pp. 2517–2522. IEEE, 2015.

Huggins, J. H. and Wood, F. Infinite structured hidden semi-markov models. *arXiv preprint arXiv:1407.0044*, 2014.

Hwang, J., Kang, J., Jang, Y., and Kim, H. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *Engineering in Medicine and Biology Society, IEEE*, volume 1, pp. 2204–2207. IEEE, 2004.

Iida, F. and Tedrake, R. Minimalistic control of biped walking in rough terrain. *Autonomous Robots*, 28(3):355–368, 2010.

Ijspeert, A. J. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.

Jang, E. A Beginner's Guide to Variational Methods: Mean-Field Approximation, 2016. URL https://blog.evjang.com/2016/08/variational-bayes.html. Accessed: 01/02/2018.

Ji, S., Xu, W., Yang, M., and Yu, K. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

Jiménez-Fabián, R. and Verlinden, O. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Medical Engineering and Physics*, 34(4):397 – 408, 2012.

Johnson, M. J. and Willsky, A. S. The hierarchical dirichlet process hidden semi-markov model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, UAI, pp. 252–259, 2010.

Johnson, M. J. and Willsky, A. S. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(1):673–701, 2013.

Johnson, M. J., Duvenaud, D., Wiltschko, A. B., Datta, S. R., and Adams, R. P. Composing graphical models with neural networks for structured representations and fast inference. In *Neural Information Processing Systems*, 2016.

Johnson, M. J. et al. *Bayesian time series models and scalable inference*. PhD thesis, Massachusetts Institute of Technology, 2014.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

Khandoker, A. H., Lai, D. T., Begg, R. K., and Palaniswami, M. Wavelet-based feature extraction for support vector machines for screening balance impairments in the elderly. *Neural Systems and Rehabilitation Engineering, IEEE*, 15(4):587–597, 2007.

Khatib, O. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Journal on Robotics and Automation*, 3(1):43–53, 1987.

Kilicarslan, A., Prasad, S., Grossman, R. G., and Contreras-Vidal, J. L. High accuracy decoding of user intentions using eeg to control a lower-body exoskeleton. In *Engineering in medicine and biology society (EMBC), IEEE*, pp. 5606–5609. IEEE, 2013.

Kimeldorf, G. S. and Wahba, G. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

King, R. A review of bayesian state-space modelling of capture–recapture–recovery data. *Interface Focus*, pp. rsfs20110078, 2012.

Kingma, D. P. Fast gradient-based inference with continuous latent variable models in auxiliary form. *arXiv preprint arXiv:1306.0733*, 2013.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kitagawa, G. and Gersch, W. *Smoothness priors analysis of time series*, volume 116. Springer Science & Business Media, 2012.

Kleijnen, J. P. Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192(3):707–716, 2009.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Krige, D. G. *A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige*. PhD thesis, University of the Witwatersrand, 1951.

Lawrence, N. D. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, volume 2, pp. 5, 2003.

Lawson, B. E., Mitchell, J., Truex, D., Shultz, A., Ledoux, E., and Goldfarb, M. A robotic leg prosthesis: Design, control, and implementation. *Robotics & Automation Magazine, IEEE*, 21 (4):70–81, 2014.

Lenzi, T., Hargrove, L., and Sensinger, J. Speed-adaptation mechanism: Robotic prostheses can actively regulate joint torque. *IEEE Robotics Automation Magazine*, 21(4):94–107, 2014a.

Lenzi, T., Hargrove, L., and Sensinger, J. Speed-adaptation mechanism: Robotic prostheses can actively regulate joint torque. *Robotics & Automation Magazine, IEEE*, 21(4):94–107, 2014b.

Leos-Barajas, V., Photopoulou, T., Langrock, R., Patterson, T. A., Watanabe, Y. Y., Murgatroyd, M., and Papastamatiou, Y. P. Analysis of animal accelerometer data using hidden markov models. 8(2):161–173, 2017.

Liberty Mutual Research Institute. 2008 workplace safety index, September 2008. URL http://www.lexisnexis.com/documents/pdf/20090217075757_large.pdf. Accessed: 15/09/2017.

Linderman, S. W., Miller, A. C., Adams, R. P., Blei, D. M., Paninski, L., and Johnson, M. J. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.

Lindsten, F., Schön, T. B., et al. Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.

Liu, M. Q., Anderson, F. C., Schwartz, M. H., and Delp, S. L. Muscle contributions to support and progression over a range of walking speeds. *Journal of Biomechanics*, 41(15):3243–3252, 2008.

Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pp. 944–949, 2007.

Lloyd, J. R., Duvenaud, D., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. Automatic construction and Natural-Language description of nonparametric regression models. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

Lo, A. Y. et al. On a class of bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357, 1984.

Lush, L., Ellwood, S., Markham, A., Ward, A., and Wheeler, P. Use of tri-axial accelerometers to assess terrestrial mammal behaviour in the wild. 2015.

Luštrek, M. and Kaluža, B. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):197–204, 2009.

MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

MacKay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Mai, A. and Commuri, S. Intelligent control of a prosthetic ankle joint using gait recognition. *Control Engineering Practice*, 49:1–13, 2016.

Manne, R. Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2(1):187–197, 1987.

Markowitz, J., Krishnaswamy, P., Eilenberg, M. F., Endo, K., Barnhart, C., and Herr, H. Speed adaptation in a powered transtibial prosthesis controlled with a neuromuscular model. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1570): 1621–1631, 2011.

McAllester, D. A pac-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013.

McClune, D. W., Marks, N. J., Wilson, R. P., Houghton, J. D., Montgomery, I. W., McGowan, N. E., Gormley, E., and Scantlebury, M. Tri-axial accelerometers quantify behaviour in the eurasian badger (meles meles): towards an automated interpretation of field data. 2(1):5, 2014.

McHutchon, A. and Rasmussen, C. E. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems*, pp. 1341–1349, 2011.

Menon, A. K. *Random projections and applications to dimensionality reduction*. PhD thesis, University of Sydney, 2007.

Mezard, M. and Montanari, A. *Information, physics, and computation*. Oxford University Press, 2009.

Miller, W. C., Speechley, M., and Deathe, A. B. Balance confidence among people with lower-limb amputations. *Physical Therapy*, 82(9):856–865, 2002.

Mirchevska, V., Luštrek, M., and Gams, M. Combining domain knowledge and machine learning for robust fall detection. *Expert Systems*, 2013.

Mistry, M. A Tutorial on Impedance Control and Physical Human-Robot Interaction. http://www.robot-manipulation.uk/impedance_control_tutorial.pdf, July 2017. Presented at the second UK Robot Manipulation Workshop. Accessed: 13/09/2017.

Mockus, J., Tiesis, V., and Zilinskas, A. *Toward Global Optimization, volume 2, chapter Bayesian Methods for Seeking the Extremum*. Elsevier, 1978.

Moore, J. K., Hnat, S. K., and van den Bogert, A. J. An elaborate data set on human gait and the effect of mechanical perturbations. *PeerJ*, 3:e918, 2015.

Morimoto, J., Noda, T., and Hyon, S. Extraction of latent kinematic relationships between human users and assistive robots. In *Robotics and Automation (ICRA), IEEE*, pp. 3909–3915. IEEE, 2012.

Mosby, I. *Mosby's dictionary of medicine, nursing & health professions*. Elsevier Health Sciences, 2013.

Mubashir, M., Shao, L., and Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.

Murphy, K. P. Hidden semi-markov models HSMMs, 2002.

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

Navarro, G. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

Neiswanger, W., Wood, F., and Xing, E. The dependent dirichlet process mixture of objects for detection-free tracking and object modeling. In *Artificial Intelligence and Statistics*, pp. 660–668, 2014.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

Newman, M. E. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5): 323–351, 2005.

Noonan, M. J., Markham, A., Newman, C., Trigoni, N., Buesching, C. D., Ellwood, S. A., and Macdonald, D. W. Climate and the individual: inter-annual variation in the autumnal activity of the european badger (meles meles). 9(1), 2014.

Noury, N., Barralon, P., Virone, G., Boissy, P., Hamel, M., and Rumeau, P. A smart sensor based on rules and its evaluation in daily routines. In *Engineering in Medicine and Biology Society, IEEE*, volume 4, pp. 3286–3289. IEEE, 2003.

Noury, N., Fleury, A., Rumeau, P., Bourke, A., Laighin, G., Rialle, V., and Lundy, J. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, IEEE*, pp. 1663–1666. IEEE, 2007.

Olivieri, D. N., Gómez Conde, I., and Vila Sobrino, X. A. Eigenspace-based fall detection and activity recognition from motion templates and machine learning. *Expert Syst. Appl.*, 39(5): 5935–5945, April 2012.

Osborne, M. and Roberts, S. J. Gaussian processes for prediction. *Technical Report PARG-07-01*, 2007.

Osborne, M. A. Lecture notes on C24 Advanced Probability Theory, Michaelmas 2015.

O'Searcoid, M. *Metric spaces*. Springer Science & Business Media, 2006.

OSHA. Commonly used statistics, September 2012. URL https://www.osha.gov/oshstats/commonstats.html. Accessed: 15/09/2017.

Pagano, A., Rode, K., Cutting, A., Owen, M., Jensen, S., Ware, J., Robbins, C., Durner, G., Atwood, T., Obbard, M., et al. Using tri-axial accelerometers to identify wild polar bear behaviors. 32:19–33, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Pepels, T., Cazenave, T., Winands, M. H., and Lanctot, M. Minimizing simple and cumulative regret in monte-carlo tree search. In *Workshop on Computer Games*, pp. 1–15. Springer, 2014.

Peters, J. and Schaal, S. Learning to control in operational space. *International Journal of Robotics Research*, 27(2):197–212, 2008.

Pham, Q.-C. Examples: hybrid control and impedance control, 2016. URL http://osrobotics.org/pages/examples_force_control.html. Accessed: 15/09/2017.

Phillips, J. S., Patterson, T. A., Leroy, B., Pilling, G. M., and Nicol, S. J. Objective classification of latent behavioral states in bio-logging data using multivariate-normal hidden markov models. 25(5):1244–1258, 2015.

Pilon, C. Probabilistic programming and bayesian methods for hackers, 2015.

Pons-Moll, G. and Rosenhahn, B. Motion Models for People Tracking. In *Model-Based Pose Estimation*, chapter 9. Springer, 2011.

Puolamäki, K. and Kaski, S. Bayesian solutions to the label switching problem. In *International Symposium on Intelligent Data Analysis*, pp. 381–392. Springer, 2009.

Quintero, D., Villarreal, D. J., and Gregg, R. D. Preliminary experiments with a unified controller for a powered knee-ankle prosthetic leg across walking speeds. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5427–5433, 2016.

Quintero, H., Farris, R., Hartigan, C., Clesson, I., and Goldfarb, M. A powered lower limb orthosis for providing legged mobility in paraplegic individuals. *Topics in Spinal Cord Injury Rehabilitation*, 17(1):25–33, 2011.

Rabiner, L. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Rahman, A., Smith, D., Hills, J., Bishop-Hurley, G., Henry, D., and Rawnsley, R. A comparison of autoencoder and statistical features for cattle behaviour classification. In *Neural Networks (IJCNN)*, pp. 2954–2960. IEEE, 2016.

Rainforth, T., Le, T. A., van de Meent, J.-W., Osborne, M. A., and Wood, F. Bayesian Optimization for Probabilistic Programs. In *Advances in Neural Information Processing Systems*, pp. 280–288, 2016.

Rasmussen, C. E. and Ghahramani, Z. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pp. 881–888, 2002.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. The MIT Press, 2006.

Reiss, A. and Stricker, D. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC)*, pp. 108–109. IEEE, 2012.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Rocha, H. On the selection of the most adequate radial basis function. *Applied Mathematical Modelling*, 33(3):1573–1583, 2009.

Rocke, D. M. Constructive statistics: estimators, algorithms, and asymptotics. *Computing Science and Statistics*, pp. 3–14, 1998.

Ronao, C. A. and Cho, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59(Supplement C):235–244, 2016.

Ross, S. and Bagnell, J. A. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

Ruelle, D. The claude bernard lecture, 1989. deterministic chaos: the science and the fiction. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 427, pp. 241–248. The Royal Society, 1990.

Saatci, Y. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2012.

Saeedi, A., Hoffman, M., Johnson, M., and Adams, R. The segmented iHMM: A simple, efficient hierarchical infinite HMM. In *Proceedings of the International Conference on Machine Learning (ICML-16)*, 2016.

Särkkä, S. *Bayesian filtering and smoothing*. Number 3. Cambridge University Press, 2013.

Schaller, G. B. *The Serengeti Lion: A Study of Predator-prey Relations. Drawings by Richard Keane*. University of Chicago Press, 1974.

Schön, T. B., Lindsten, F., Dahlin, J., Wågberg, J., Naesseth, C. A., Svensson, A., and Dai, L. Sequential monte carlo methods for system identification. *IFAC-PapersOnLine*, 48(28): 775–786, 2015.

Seeger, M. Bayesian gaussian process models: Pac-bayesian generalisation error bounds and sparse approximations. 2003.

Sethuraman, J. A constructive definition of dirichlet priors. *Statistica Sinica*, pp. 639–650, 1994.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Shakespeare, W. *Henry V, Act III*. Modern Library, 1598.

Shannon, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

Siddiqi, S. M., Gordon, G. J., and Moore, A. W. Fast state discovery for hmm model selection and learning. In *Artificial Intelligence and Statistics*, pp. 492–499, 2007.

Smola, A. J. and Bartlett, P. L. Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pp. 619–625, 2001.

Snelson, E. Tutorial: Gaussian process models for machine learning. *Gatsby Computational Neuroscience Unit, UCL*, 2006.

Snelson, E., Ghahramani, Z., and Rasmussen, C. E. Warped gaussian processes. In *Advances in neural information processing systems*, pp. 337–344, 2004.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 2951–2959, 2012.

Song, S. and Geyer, H. Regulating speed in a neuromuscular human running model. In *Humanoid Robots (Humanoids), IEEE-RAS*, pp. 217–222. IEEE, 2015.

Stein, J. L. and Flowers, W. C. Stance phase control of above-knee prostheses: knee control versus sach foot design. *Journal of Biomechanics*, 20(1):19–28, 1987.

Stepleton, T. S., Ghahramani, Z., Gordon, G. J., and Lee, T. S. The block diagonal infinite hidden markov model. In *International Conference on Artificial Intelligence and Statistics*, pp. 552–559, 2009.

Stevens, J. A., Corso, P. S., Finkelstein, E. A., and Miller, T. R. The costs of fatal and non-fatal falls among older adults. *Injury prevention*, 12(5):290–295, 2006.

Stevens, J. A., Ballesteros, M. F., Mack, K. A., Rudd, R. A., DeCaro, E., and Adler, G. Gender differences in seeking care for falls in the aged medicare population. *American Journal of preventive medicine*, 43(1):59–62, 2012.

Sudderth, E. B. *Graphical models for visual object recognition and tracking.* PhD thesis, Massachusetts Institute of Technology, 2006.

Sung, J., Ponce, C., Selman, B., and Saxena, A. Human activity detection from rgbd images. 64, 2011.

Sup, F., Bohara, A., and Goldfarb, M. Design and control of a powered transfemoral prosthesis. *The International Journal of Robotics Research*, 27:263–273, 2008.

Sup, F., Varol, H. A., Mitchell, J., Withrow, T. J., and Goldfarb, M. Preliminary evaluations of a self-contained anthropomorphic transfemoral prosthesis. *IEEE/ASME Transactions on mechatronics*, 14(6):667–676, 2009.

Sup, F., Varol, H. A., and Goldfarb, M. Upslope walking with a powered knee and ankle prosthesis: Initial results with an amputee subject. *Neural Systems and Rehabilitation Engineering, IEEE*, 19(1):71–78, 2011.

Taylor, G. W., Hinton, G. E., and Roweis, S. T. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*, pp. 1345–1352, 2006.

Taylor, G. W. *Composable, distributed-state models for high-dimensional time series.* PhD thesis, University of Toronto, 2009.

Teh, Y. W. Dirichlet processes: Tutorial and practical course. *Gatsby Computational Neuroscience Unit, University College London*, 2007.

Teh, Y. W. Dirichlet process. In *Encyclopedia of Machine Learning*, pp. 280–287. Springer, 2011.

Teh, Y. W. and Jordan, M. I. Hierarchical bayesian nonparametric models with applications. *Bayesian Nonparametrics*, 1, 2010.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476), 2006.

Tenorth, M., Bandouch, J., and Beetz, M. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV*, 2009.

Thatte, N. and Geyer, H. Towards local reflexive control of a powered transfemoral prosthesis for robust amputee push and trip recovery. In *Intelligent Robots and Systems (IROS), IEEE/RSJ*, pp. 2069–2074. IEEE, 2014.

Thelen, D. G. and Anderson, F. C. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of Biomechanics*, 39(6): 1107–1115, 2006.

Tolpin, D., van de Meent, J.-W., Paige, B., and Wood, F. Output-sensitive adaptive metropolis-hastings for probabilistic programs. In Appice, A., Rodrigues, P. P., Santos Costa, V., Gama, J., Jorge, A., and Soares, C. (eds.), *Machine Learning and Knowledge Discovery in Databases*, volume 9285, pp. 311–326. Springer, 2015.

Trethowan, P., Fuller, A., Haw, A., Hart, T., Markham, A., Loveridge, A., Hetem, R., Preez, B., and Macdonald, D. W. Getting to the core: Internal body temperatures help reveal the ecological function and thermal implications of the lions' mane. 7(1):253–262, 2017.

Tromp, A., Pluijm, S., Smit, J., Deeg, D., Bouter, L., and Lips, P. Fall-risk screening test: a prospective study on predictors for falls in community-dwelling elderly. *Journal of Clinical Epidemiology*, 54(8):837–844, 2001.

Tucker, M. R., Olivier, J., Pagel, A., Bleuler, H., Bouri, M., Lambercy, O., Millán, J. d. R., Riener, R., Vallery, H., and Gassert, R. Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of NeuroEngineering and Rehabilitation*, 12(1):1, 2015.

Van der Noot, N., Ijspeert, A. J., and Ronsse, R. Biped gait controller for large speed variations, combining reflexes and a central pattern generator in a neuromuscular model. In *Robotics and Automation (ICRA), IEEE*, pp. 6267–6274. IEEE, 2015.

Van Gael, J. *Bayesian Nonparametric Hidden Markov Models*. PhD thesis, University of Cambridge, 2011.

Van Gael, J., Saatci, Y., Teh, Y. W., and Ghahramani, Z. Beam sampling for the infinite hidden markov model. In *Proceedings of the International Conference on Machine learning*, pp. 1088–1095. ACM, 2008.

van Hoof, H., Chen, N., Karl, M., van der Smagt, P., and Peters, J. Stable reinforcement learning with autoencoders for tactile and visual data. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 3928–3934. IEEE, 2016.

van Wilgenburg, E. and Elgar, M. A. Confirmation bias in studies of nestmate recognition: a cautionary note for research into the behaviour of animals. 8(1):e53548, 2013.

Vandekerckhove, J., Matzke, D., and Wagenmakers, E.-J. Model comparison and the principle. *The Oxford handbook of computational and mathematical psychology*, pp. 300, 2015.

Varol, H. A., Sup, F., and Goldfarb, M. Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Transactions on Biomedical Engineering*, 57(3):542–551, 2010.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

Vrieling, A., Van Keeken, H., Schoppen, T., Otten, E., Halbertsma, J., Hof, A., and Postema, K. Uphill and downhill walking in unilateral lower limb amputees. *Gait & Posture*, 28(2): 235–242, 2008.

Vukobratović, M. and Borovac, B. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

Vuong, N. D., Ang, M. H., Lim, T. M., and Lim, S. Y. An analysis of the operational space control of robots. In *Robotics and Automation (ICRA)*, pp. 4163–4168. IEEE, 2010.

Wahlström, N., Kok, M., Schön, T. B., and Gustafsson, F. Modeling magnetic fields using gaussian processes. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE*, pp. 3522–3526. IEEE, 2013.

Wallach, H., Jensen, S., Dicker, L., and Heller, K. An alternative prior process for nonparametric bayesian clustering. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 892–899, 2010.

Wang, C., Paisley, J., and Blei, D. Online variational inference for the hierarchical dirichlet process. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 752–760, 2011.

Wang, J. M. Gaussian Process Dynamical Models for Human Motion. Master's thesis, University of Toronto, 2005.

Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L. Deep learning for sensor-based activity recognition: A survey. *arXiv preprint arXiv:1707.03502*, 2017a.

Wang, Y., Wu, K., and Ni, L. M. Wifall: Device-free fall detection by wireless networks. *Transactions on Mobile Computing, IEEE*, 16(2):581–594, 2017b.

Wang, Z., Deisenroth, M. P., Amor, H. B., Vogt, D., Schölkopf, B., and Peters, J. Probabilistic modeling of human movements for intention inference. *Proceedings of Robotics: Science and systems, VIII*, 2012.

Waters, R., Perry, J., Antonelli, D., and Hislop, H. Energy cost of walking of amputees: the influence of level of amputation. *Journal of Bone and Joint Surgery*, 58(1):42–46, 1976.

Weigend, A. S. and Gershenfeld, A. S. *Time series prediction: forecasting the future and understanding the past*. Addison Wesley, 1994.

Weisstein, E. W. Countably infinite. From MathWorld – A Wolfram Web Resource, 2000. URL http://mathworld.wolfram.com/CountablyInfinite.html. Accesed on: 23/08/2017.

Williams, C., Klanke, S., Vijayakumar, S., and Chai, K. M. Multi-task gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems*, pp. 265–272, 2009.

Williams, C. K. and Rasmussen, C. E. Gaussian processes for machine learning. *MIT Press*, 2 (3):4, 2006.

Wilson, A. G. and Nickisch, H. Kernel interpolation for scalable structured gaussian processes (kiss-gp). *arXiv preprint arXiv:1503.01057*, 2015.

Winter, D. A. Energy generation and absorption at the ankle and knee during fast, natural, and slow cadences. *Clinical Orthopaedics and Related Research*, 175:147–154, 1983.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

Wolpert, D. M. and Ghahramani, Z. Computational principles of movement neuroscience. *Nature neuroscience*, 3(11s):1212, 2000.

Wolpert, D. H. and Macready, W. G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

Wood, F., van de Meent, J. W., and Mansinghka, V. A new approach to probabilistic programming inference. In *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, 2014.

Wood, F. Nips probabilistic programming tutorial 2015, 2015.

Woodbury, M. A. Inverting modified matrices. *Memorandum report*, 42(106):336, 1950.

Woodward, A. M., Alsberg, B. K., and Kell, D. B. The effect of heteroscedastic noise on the chemometric modelling of frequency domain data. *Chemometrics and intelligent laboratory systems*, 40(1):101–107, 1998.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

Yun, Y., Kim, H.-C., Shin, S. Y., Lee, J., Deshpande, A. D., and Kim, C. Statistical method for prediction of gait kinematics with gaussian process regression. *Journal of Biomechanics*, 47 (1):186–192, 2014.

Zhang, A., Gultekin, S., and Paisley, J. Stochastic variational inference for the hdp-hmm. In *Artificial Intelligence and Statistics*, pp. 800–808, 2016.

Zhang, T., Wang, J., Liu, P., and Hou, J. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *International Journal of Computer Science and Network Security*, 6(10):277–284, 2006.

Appendices

APPENDIX A

Approximate inference algorithms

Algorithm 2: Metropolis-Hastings MCMC sampling

Input : $\mathbf{x}^1 \leftarrow$ Set starting point
$L \leftarrow$ Number of samples

for $l = 2$ ***to*** L **do**
  Draw a candidate sample $\mathbf{x}^{\mathrm{c}}$, from $q(\mathbf{x}' \mid \mathbf{x}^{l-1})$
  Let $a = \min\left(1, \frac{\mathbb{P}(\mathbf{x}')q(\mathbf{x}'|\mathbf{x})}{\mathbb{P}(\mathbf{x})q(\mathbf{x}|\mathbf{x}')}\right)$
  if $a \geq 1$ **then**
    $\mathbf{x}^l = \mathbf{x}^{\mathrm{c}}$
  else
    Draw a random value u uniformly from the unit interval $[0, 1]$.
    if $u < a$ **then**
      $\mathbf{x}^l = \mathbf{x}^{\mathrm{c}}$
    else
      $\mathbf{x}^l = \mathbf{x}^{l-1}$

Output : In the limit, samples drawn from the stationary distribution.

Algorithm 3: Conditional sequential Monte Carlo

Input : $L \leftarrow$ Number of particles
$\mathbf{x}_{1:T}^{(l)} \leftarrow$ Particle path with lineage $B_{1:T}^{l}$, parametrised by $\boldsymbol{\theta}$
$\mathbf{x}_{1:T}^{*} \leftarrow$ Arbitrary initial path

for $t = 1$ **do**
  for $l \in \{1, \ldots, L\} \setminus B_{t=1}^{l}$ **do**
    Set $\mathbf{x}_1^{(l)} \leftarrow \mathbf{x}_1^{*}$
    Simulate $\mathbf{x}_1^{(l)} \sim \mathbb{P}(\mathbf{x}_1 \mid \boldsymbol{\theta})$
    Calculate $w_1^{(l)} \propto \mathbb{P}\Big(\mathbf{y}_1 \mid \mathbf{x}_1^{(l)}\Big)$
  $\overline{w}_1^{1:L} \leftarrow$ Normalise weights
for $t = 2$ ***to*** T **do**
  Set $\mathbf{x}_t^{(1)} \leftarrow \mathbf{x}_t^{*}$ **for** $l \in \{2, \ldots, L\} \setminus B_t^{l}$ **do**
    Sample ancestral indices
    $A_{t-1}^{(l)} \sim \text{Unif}\left(\cdot \mid \overline{w}_{t-1}^{(1)}, \ldots, \overline{w}_{t-1}^{(L)}\right)$
    Simulate $\mathbf{x}_t^{(l)} \sim \mathbb{P}\Big(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{A_t^{(l)}}\Big)$
    Calculate $w_t^{(l)} \propto \mathbb{P}(\mathbf{y}_t \mid \mathbf{x}_t^{(l)})$
  $\overline{w}_t^{1:L} \leftarrow$ Normalise weights

Output : Particle-conditional posterior samples of all latent state-space model parameters.

Algorithm 4: Particle Gibbs

Input : $\mathbf{x}_{1:T} \leftarrow$ Initialise latent state sequence
$L \leftarrow$ Number of particles
$\boldsymbol{\theta}^{(L)} \leftarrow$ Initialise model parameters
$S \leftarrow$ Number of sweeps (MCMC iterations)

for $s = 1$ ***to*** S **do**
  $\boldsymbol{\theta} \sim \mathbb{P}(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T})$
  Run Conditional SMC with $\{\mathbf{x}_{1:T}, \boldsymbol{\theta}, L\}$, returning L particles $\mathbf{x}_{1:T}$ Simulate $\mathbf{x}_{1:T}^{(L)} \sim \text{Unif}\left\{\mathbf{x}_{1:T}^{(1)}, \ldots, \mathbf{x}_{1:T}^{(L)}\right\}$

Output : Posterior samples of (i) state-space model parameters and (ii) latent state sequence.

APPENDIX B

Additional results for PAMAP2 dataset

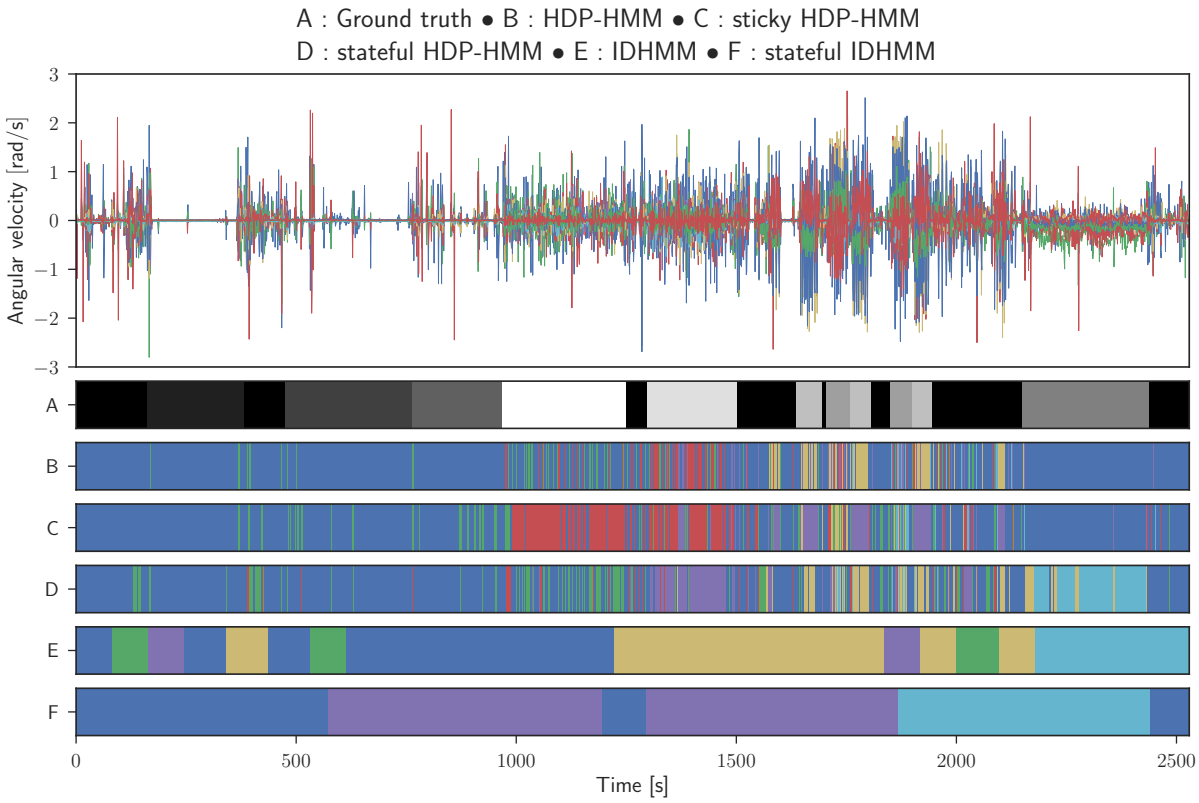


Figure B.1: From the top, the **first panel** shows the observations used for segmentation. The **second panel** shows the manual segmentation. The **following panels** depict the model segmented sequences, for the highest log-likelihoods. Inference was provided by sequential Monte Carlo.

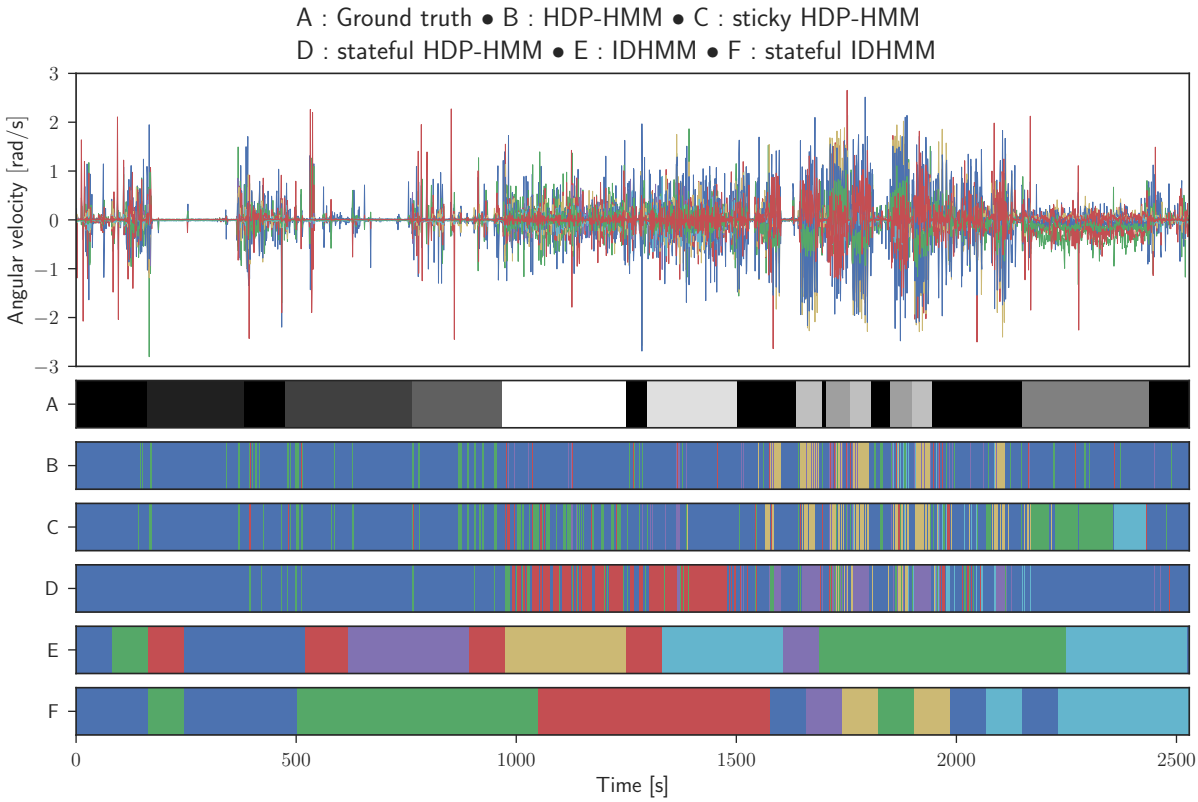


Figure B.2: From the top, the **first panel** shows the observations used for segmentation. The **second panel** shows the manual segmentation. The **following panels** depict the model segmented sequences, for the highest log-likelihoods. Inference was provided by particle independent Metropolis Hastings.

Do not go gentle into that good night,
Old age should burn and rave at close of day;
Rage, rage against the dying of the light.

Though wise men at their end know dark is right,
Because their words had forked no lightning they
Do not go gentle into that good night.

Good men, the last wave by, crying how bright
Their frail deeds might have danced in a green bay,
Rage, rage against the dying of the light.

Wild men who caught and sang the sun in flight,
And learn, too late, they grieved it on its way,
Do not go gentle into that good night.

Grave men, near death, who see with blinding sight
Blind eyes could blaze like meteors and be gay,
Rage, rage against the dying of the light.

And you, my father, there on the sad height,
Curse, bless, me now with your fierce tears, I pray.
Do not go gentle into that good night.
Rage, rage against the dying of the light.