Compositional Uncertainty in Models of Alignment

Kazlauskaite, Ieva

Award date:
2020

Awarding institution:
University of Bath

Compositional Uncertainty in Models of Alignment

Ieva Kazlauskaitė

A thesis submitted for the degree of Doctor of Engineering

University of Bath

Department of Computer Science

February 2020

Abstract

This thesis studies the problem of temporal alignment of sequences and the uncertainty propagation in models of alignment.

Temporal alignment of sequences is the task of removing the differences between the observed time series arising from the differences in their relative timing. It is a common preprocessing step in time series modelling, usually performed in isolation from the data analysis and modelling. The methods proposed in this thesis cast alignment learning in a framework where both the alignment and the data is modelled simultaneously. Specifically, we use tools from Bayesian nonparametrics to model each sequence as a composition of a monotonic warping function, that accounts for the differences in timing, and a latent function, which is an aligned version of the observed sequence. Combined with a probabilistic alignment objective, such an approach allows us to align sequences into multiple, a-priori unknown groups in an unsupervised manner. Furthermore, the use of Bayesian nonparametrics offers the benefits of principled modelling of the noisy observed sequences, explicit priors that encode our beliefs about the constituent parts of the model and the generative process of the data, and an ability to adapt to the complexity of the data.

Another feature of the probabilistic formulation that is lacking in the traditional temporal alignment models is an explicit quantification of the different kinds of uncertainties arising in the alignment problem. These include the uncertainties related to the noisy observations and the fact that the observed data may be explained in multiple different ways, all of which are plausible under our prior assumptions.

While formulating various parts of the model, we encounter and discuss some of the challenges of Bayesian modelling, most notably the need for approximate inference. We argue that variational distributions which include correlations between the hierarchical components of the model are necessary to take advantage of the potential of the model to discover the compositional structure in the data and to capture the uncertainty arising from it.

Acknowledgements

The submission of this thesis marks the end of a long journey and I would like to take this opportunity to thank those who have helped and guided me along the way.

I would like to start by thanking my supervisor Neill D.F. Campbell whose endless enthusiasm has played a central role in my choice of research topic and has in many ways shaped my way of thinking. I thank my co-supervisor Carl Henrik Ek for the encouragement and mentorship over the past three years. I am grateful to both Neill and Carl Henrik for the inspiration, energy and passion abound in our research group.

I am thankful to everyone at Frostbite Physics, especially Mike Bassett for welcoming me to the team, and to Tom Waterson for kindly supporting me and encouraging me to pursue my interests.

Thanks are due to my colleagues and everyone whom I have had the good fortune to meet and work with over the past years. I am especially grateful to Garoe Dorta for being the kindest and most thoughtful friend anyone could ever ask for, Ivan Ustyuzhaninov for the extraordinary ability to explain how things should work (in principle), Markus Kaiser and Erik Bodin for many rewarding discussions, and Chris Lewin for tolerating my lack of tact.

I thank Ieuan and Lucas for putting things into perspective, and above all, I am indebted to my parents, Vita and Gintautas – Ačiū jums!

Finally, I am grateful to my examiners, Neil Lawrence and Xi Chen for kindly offering their time to review this thesis and for the rewarding discussion during the viva.

The research leading to the results presented here has received funding from the EPSRC CDE (EP/L016540/1) grant.

"His first paper was left unfinished. Or rather, he put down only the first sentence. Actually, the beginning of the first sentence. Specifically: "As we all know...". At this juncture, the brilliantly conceived work was cut short."

– S. Dovlatov, Pushkin Hills (1983)

"- O kai baigsiu mokslus, ateisiu į zoologijos sodą ir dirbsiu tavo pavaduotoju!
- Ne, Kūlverstuk, ne, niekada! Tau negalima pas mus dirbti.
- Kodėl?
- Kodėl, kodėl... Tave suės! "

– Kūlverstukas eina į mokyklą (1983)

Contents

List of Figures

List of Tables

Nomenclature

General notation

$\mathbb{I}$	Identity matrix
$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
a	Scalar

GP notation

$k_\theta(\cdot,\cdot)$	GP covariance function, parametrised by hyper-parameters θ
$\mathbf{K}_{ij}$	Covariance matrix $\mathbf{K}$ with $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
M	Number of inducing points
$\mathbf{z}_m$	Inducing location, $\mathbf{z}_m \in \mathbb{R}^D$ ($D = 1$ in the alignment setting)
$\mathbf{Z}$	Matrix of inducing locations, $\mathbf{Z} \in \mathbb{R}^{M \times D}$
u_m	Inducing output/value/variable, $u_m \in \mathbb{R}$
$\mathbf{U}$	Vector of inducing outputs/values/variables, $\mathbf{U} \in \mathbb{R}^M$

Alignment-specific notation

In Ch. 3, 4, 5 we discuss different variants of the alignment model. In these chapters we use the following problem-specific notation:

J	Number of sequences to align
N	Number of observations in a sequence
$\mathbf{y}_j$	Observed sequence, $\mathbf{y}_j \in \mathbb{R}^N$
$\mathbf{x}$	Linearly spaced N points between -1 and 1, $\mathbf{x} \in \mathbb{R}^N$
$\mathbf{s}_j$	Vector of pseudo-observations (an aligned sequence), $\mathbf{s}_j \in \mathbb{R}^N$
$g_j(x)$	Warping function $g : \mathbb{R} \to \mathbb{R}$
$\mathbf{g}_j$	Evaluations of the warping function at $\mathbf{x}$: $\mathbf{g}_j = g_j(\mathbf{x})$

$\mathbf{Y}$	Matrix of observed sequences, $\mathbf{Y} \in \mathbb{R}^{J \times N}$
$\mathbf{X}$	Matrix of J copies of vector $\mathbf{x}$, $\mathbf{X} \in \mathbb{R}^{J \times N}$
$\mathbf{S}$	Matrix of pseudo-observations, $\mathbf{S} \in \mathbb{R}^{J \times N}$
$\mathbf{G}$	Matrix of warped inputs $\mathbf{g}_j$, $\mathbf{G} \in \mathbb{R}^{J \times N}$
Q	Number of dimensions in the GP-LVM latent space
$\mathbf{q}_j$	Vector of latent variables corresponding to the j-th sequence, $\mathbf{q}_j \in \mathbb{R}^Q$
$\mathbf{Q}$	Matrix of latent vectors $\{\mathbf{q}_j\}$, $\mathbf{Q} \in \mathbb{R}^{J \times Q}$
$\mathbf{c}_j$	Vector (one-hot) of cluster assignments in GP-DPMM model

In Ch. 6 and 7 we discuss the monotonic flow model and its applications to alignment. We use the following notation specific to these chapters:

N_S	Number of samples in Monte Carlo sampling
$S(t, \omega; \mathbf{x})$	Vector of solutions at time t of an SDE with initial values $\mathbf{x}$
$S(\mathbf{x})$	Shorthand notation for $S(T, \omega; \mathbf{x})$ for fixed values of T and ω

1

Introduction

> *"The tendency toward determinism is somehow implied in the method of retrospection itself. In retrospection we seem to perceive the logic of events, which unfold themselves in a regular order, according to a recognizable pattern, with an alleged inner necessity, so that we get the impression that it really could not have happened otherwise."*
>
> – G. Florovsky, The study of the past (1969)

In the above, Georges Florovsky, a 20^{th} century historian and theologian, refers to history and the work done by historians. He argues that when examining and explaining events that happened in the past it is easy to underestimate the uncertainty related to these events. In particular, it is tempting to assume that the decisions made throughout history were largely predetermined, ignoring the hesitations, deliberations and sheer randomness that accompany most decisions made by humans, and to underestimate the uncertainty in our knowledge of past events that stems from the limitations and the contradictions in historical data. As historians look at the events in the context of what happened afterwards, it is natural to try to create some consistent and mostly deterministic narrative for these events. Research in psychology and economics has reported a similar phenomenon [Guilbault et al., 2004], often referred to as hindsight bias [Fischhoff, 2003], that can be summarised by the observation that:

> *"A person asked to provide an argument supporting a given hypothesis usually finds this hypothesis more plausible as a result."*
>
> – D. J. Koehler, Explanation, Imagination, and Confidence in Judgment (1991)

In a similar manner, given a finite set of imperfect observations, our goal in machine learning (ML) is to infer some patterns in the observed data. In ML, it is generally accepted that we do not know what will happen in the future (*i.e.* predictions at new inputs are uncertain), however, the idea that there might be multiple explanations of the "past" (*i.e.* the data we observed could be explained in many different yet equally valid ways) might not be as intuitive. To acknowledge the possible lack of cohesion of the various parts of the model and to learn a spectrum of different explanations for the observed data, we first need to specify the prior assumptions about our ML model (as implied by the "No-free-lunch" theorem [Shalev-Shwartz and Ben-David, 2014]) and then to propagate the uncertainties through the proposed model conditioned on the observed data.

Even when the quantification of uncertainty is not crucial for the automation of a specific task, taking the uncertainties into consideration may benefit the final solution. For instance, the data, which is central to any ML solution, is often noisy or otherwise corrupted, and designing models that explicitly acknowledge the uncertainties related to the observations may reduce the need for preprocessing of the data and lead to solutions that are more robust. Furthermore, the uncertainty may help uncover additional structure in the data. More generally, the quantification of uncertainties allows us to learn more about the nature of the underlying data generating process and the alternative explanations for the observed data; this leads to models that are more informative and interpretable.

Bayesian probabilistic inference is a framework that provides a systematic way to define the assumptions about the machine learning model, perform inference, and estimate the resulting uncertainties [Murphy, 2012]. The assumptions are specified in terms of prior distributions, defining the space of possible underlying models or explanations of the data, and the likelihood function, linking each of these possible models to the observed data. The Bayes' rule reweighs the possible models specified by the prior distribution based on how consistent they are with the observations. The outcome of the inference is a posterior distribution representing the uncertainty about the most suitable model (or, more generally, about the model that offers the best explanation of the data among the models specified by the prior).

Since Bayes' rule is independent of the problem-specific model assumptions defined through the prior and the likelihood, the Bayesian framework can be applied to a wide variety of problems. Another advantage of Bayesian modelling is that the full inference results in an entire posterior distribution of the variables of interest, rather than a point estimate or a bound on certain statistics of these variables. However, exact Bayesian

inference (*i.e.* the application of the Bayes' rule) is typically tractable only for toy problems, meaning that applying such methodology on real-world data often requires sophisticated approximate inference methods. The standard set of tools for approximate inference include Laplace approximations, variational inference, expectation propagation and sampling methods [Bishop, 2006]. However, with larger data sets and more complex models, further developments in approximate inference are needed to take full advantage of the Bayesian models.

This dissertation explores the ideas in Bayesian modelling and uncertainty propagation in the context of the specific problem of temporal sequence alignment. Our goal is to propose an approach to this problem which explicitly takes into account the assumptions of the alignment problem in a systematic way, and allows us to estimate the uncertainties present in the resulting models.

In the remainder of this chapter we briefly introduce the problem of temporal sequence alignment, its significance, the industrial motivation, the constraints on the machine learning solutions for this problem and the main contributions of the work presented in this thesis.

1.1 Temporal sequence alignment

Learning from sequential data is challenging. Data might be sampled at different and uneven rates, sequences might be collected out of phase, *etc*. Consider the following scenarios: humans performing a task may take more or less time than other humans to complete parts of it, climate patterns are often cyclic though particular events take place at slightly different times in the year, the mental ability of children varies depending on their age, neuronal spike waveforms contain temporal jitter, transmitters and receivers of a signal may not be perfectly synchronised, replicated scientific experiments often vary in timing. However, most sample statistics, *e.g.* mean and variance, are designed to capture variation in amplitude rather than phase/timing. This leads to increased sample variance, blurred fundamental data structures and an inflated number of principal components needed to describe the data. Therefore, the data needs to be aligned in order to recover the patterns mentioned above.

Temporal alignment is a non-trivial task that is often performed as a preprocessing stage to modelling [Campbell and Kautz, 2014]. The goal of such a procedure is to decompose each observed sequence into two parts: one that is specific to each observation (and referred to as a temporal warp) and another that is chosen to be similar for all

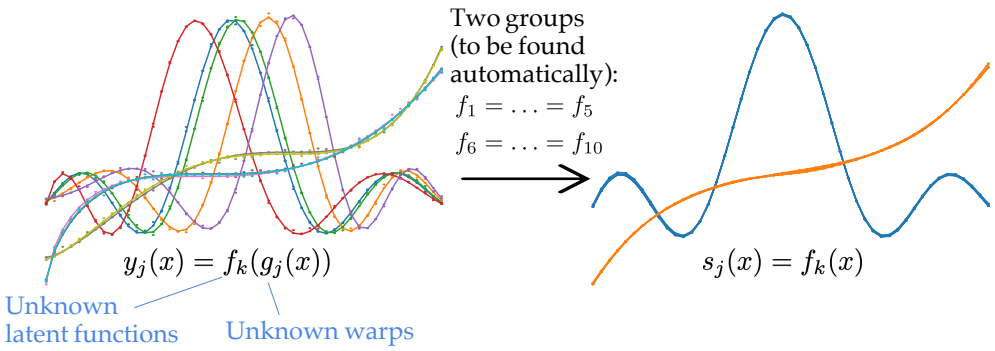


Figure 1-1: Toy example illustrating the observed data and the desired outputs.

observations, *i.e.* it shares a pattern of information across all observed sequences (and is referred to as aligned sequences or latent functions).

Typically, the temporal warps are defined to be monotonic functions, as non-monotonic ones would allow for permutation of sampled values of the observed sequences and for most applications it is logical to keep the temporal consistency of the sequences. The notion of sequence similarity traditionally comes from a local measure of pairwise similarity integrated across the sequences [Berndt and Clifford, 1994]. As noted in [Shalev-Shwartz and Ben-David, 2014], similarity is not a transitive relation, which means that if an element A is related to an element B, and B is related to an element C, then it does not imply that A is also related to C. This means that such a local measure often leads to highly non-convex optimisation problems making alignments challenging to learn. In practice, a coarse-to-fine strategy is often used to alleviate this issue [Zhou and de al Torre, 2016, Campbell and Kautz, 2014].

Further difficulties arise when the data set contains observations from several distinct functions. An illustration of this problem is shown in Fig. 1-1 where, given two types of sequences (a sinc function and a cube function) that have been warped using a set of unknown warping functions, we wish to learn that there are two types of underlying sequences and recover the warping functions that align those sequences as well as possible within the two groups. Existing alignment methods are not able to address such problems.

Methods for learning alignments can broadly be classified into two categories. The first category learns a function to warp the input domain while the second category directly learns the transformed (aligned) sequences. There are several benefits to learning a warping function explicitly as it allows us to resample the data and, by choosing an appropriate class of functions, incorporate global constraints on the alignments. These

constraints include the monotonicity and the smoothness of the warping function. A common approach to warping includes specifying a parametric function which can be challenging and often results in difficult optimisation tasks [Zhou and de al Torre, 2016].

Industrial motivation

The creation of natural-looking and stable motion of virtual characters is one of the great ambitions of research in computer graphics and related areas. The artistic approach to this problem requires intensive manual editing by an experienced animator, and, though still frequently used, is extremely costly. Therefore, significant efforts have been made from both academia and industry to develop methods that automate parts of the animation framework.

The introduction of motion capture has had a significant impact on how motion is understood and synthesised for various applications, ranging from character motion generation in entertainment to analysing one's motion for medical or sports performance reasons. However, producing large and varied data sets is expensive and requires good knowledge of the motion capture process as well as staging and acting. Consequently, recent research effort has been directed towards using motion capture data to produce predefined motions in a novel motion style [da Silva et al., 2008, Levine et al., 2012]. Such a process would allow the user to quickly and easily generate non-repetitive motions for games or other virtual environments.

The generation of content involves two parts: sequence alignment and the creation of a low-dimensional representation of motions (often referred to as a manifold of motions). In the sequence alignment step, we match the clips of motion so that the elements in all the motion sequences are in correspondence; such differences due to the timing of the action are typically referred to as temporal warps. The resulting aligned motion sequences match in terms of timing, and differ only in terms of precise characteristics of a particular motion. Consequently, the navigation of the motion manifold results in smooth changes in the style of the generated motion. The style of motion may refer to inherent traits (e.g. limping), manners that are acted (e.g. mimicking someone else's motion), or peculiarities associated with the state of the character (e.g. tiredness). Therefore, given a number of aligned motion sequences of the same motion (e.g. a right-handed golf swing) from one or multiple actors, our aim is to be able to interpolate the style of motion smoothly, and yield novel realistic-looking motion sequences.

As an additional complication, the sequences may be of different lengths and they may

be captured at different sampling rates. Furthermore, some data sets may contain sequences of different actions, for example, a swing and a putt of a golf ball in the case of golf motions; however, it is not known a-priori which sequence corresponds to which action. The overall goal is to create a generative model of the aligned sequences that can be used to produce new sequences of each of the actions that are not corrupted by the temporal warps but still contain some differences due to the different styles of motion.

While the two steps needed for content generation – the alignment and the creation of the manifold – may be performed separately, the preferred solution is one that limits the ways in which the data needs to be preprocessed. This motivates the design of a joint model that admits noisy sequences of different types, lengths and that are (potentially) captured at different sampling rates, and returns a low-dimensional representation of the aligned motions. Therefore, the goal when designing models for character animation is threefold and it includes simultaneously (1) learning the temporal warps that align the observed sequences, (2) learning what motions each sequence corresponds to, and (3) creating a generative model of the aligned sequences. To our knowledge, none of the existing approaches satisfy the aforementioned requirements.

Compositional uncertainty in alignments

An essential feature of the alignment model is its compositionality: the observed sequences are modelled as compositions of the warps (which are unique for each sequence) and the latent, group-defining functions (which are shared across multiple sequences). For example, consider the motion capture sequences of human movement; in this case the latent functions correspond to different types of movements (*e.g.* walking, running, etc.), while the warps correspond to the trial-specific characteristics of each of these sequences.

There is a specific kind of uncertainty related to compositional models, which is separate from uncertainty arising from noisy observations or finite data samples. Intuitively it can be summarised by the fact even in the limit of infinitely many noiseless observations, the explanation for the data with a composite model is not deterministic. We refer to this uncertainty as compositional uncertainty.

As a simple example, consider the identity function $y(x) = x$, and a noiseless data set $\mathcal{D} = \{(x_i, y(x_i))\}_{i=1}^{N}$ with N arbitrary large. In this case $\mathcal{D}$ can be explained by any composition $f \circ g$ of two functions, such that $f = g^{-1}$. In a similar way, in the alignment setting there are multiple possible compositions of the warp g and latent function f fitting the observed sequences; however, not all such compositions might be

consistent with the prior assumptions on f and g, and with the goal of aligning the sequences. The existing work on compositional (deep) probabilistic models typically concentrates on predictive uncertainty and rarely explores the issue of compositional uncertainty [Havasi et al., 2018]. Quantifying the compositional uncertainty in the alignment model under the prior assumptions on the warps (*e.g.* monotonicity), the latent functions (*e.g.* smoothness) and the alignment constraints (multiple observed sequences are explained, or aligned, with the same latent function) is one the main themes of the work presented in this thesis.

Evaluation

The compositional structure of the alignment models means that, in general, there are multiple possible alignments consistent with the observations. Consider, for example, the alignment in Fig. 1-1. All sinc curves have been aligned to the blue curve in the right panel but there are infinitely many other possible solutions which can be obtained by jointly changing the (monotonic) warpings and/or the fitted latent functions (in this case the sinc functions). In addition, the alignment model should automatically discover different groups of sequences and align them separately within each group (in more complex data sets the sequence-to-group assignment might also be ambiguous). These two factors imply that the alignment model cannot be evaluated by simply comparing the obtained solution to some ground truth alignment, but rather the evaluation procedure should explicitly account for different possible alignments.

Our approach to evaluating the obtained alignment solutions consists of two parts. The first part deals with the methods computing point estimates of alignments (Ch. 3 and 4). This setting is suboptimal since point estimates do not capture the full range of possible solutions. To evaluate such point solutions, we (1) check if the obtained group assignments coincide with the ground-truth ones (which we assume to be known for simplicity and concreteness), (2) compute the pairwise distances between the aligned sequences within each group, and (3) compute how much the corresponding warps differ from the identity warps. Such an evaluation procedure reflects our prior assumption that the observed sequences can be aligned using the warps which are close to the identity functions, hence we penalise the solutions violating this assumption.

In some cases we evaluate the point estimate of the alignment solution by computing the squared distances between the computed warps and some fixed ones. Such a measure assumes one of the possible alignments to be the ground-truth one and penalises the computed alignment for deviating from it. This measure is different from the one of

pairwise distances between the aligned sequences (which evaluates whether the sequences are aligned at all rather than aligned to specific sequences), and we use it for data sets with known ground-truth alignment having a specific physical meaning, in which case it makes sense to prefer one alignment solution to other possible ones.

The second part of the evaluation procedure deals with the evaluation of alignments obtained using fully probabilistic models (Ch. 6 and 7), which compute the distribution of possible alignments reflecting the compositional and the group assignment uncertainties. Uncertainty quantification in such models is an area of active research without established metrics, hence we limit ourselves to providing a qualitative assessment of our models and discussing them in relation to other existing approaches.

The results for the compositional uncertainty are presented in two parts. First, we introduce a monotonic random process and compare its performance to existing approaches where we report the quantitative results on regression tasks and a qualitative comparison of the estimates of uncertainty for the different models. Then we provide some qualitative comparisons of uncertainties in compositional models (outside of the alignment setting) for different inference schemes. Finally, we explore the compositional models of alignments with the monotonic random process as one of the constituent parts of this model, and provide a discussion of the estimates of uncertainty in the compositions.

1.2 Dissertation outline

In this thesis we use methods of Bayesian nonparametrics which encapsulate alignment and modelling within a single framework, allowing us to capture global structures in the data.

In Ch. 2, we provide a short introduction to Bayesian nonparametrics with a focus on Gaussian processes (GPs). The second part of this chapter (Sec. 2.2.1) provides a discussion of the existing approaches to the temporal alignment problem and identifies some of the limitations of the existing work.

Chapters 3 and 4 explore the utility of GP priors in the alignment setting for principled modelling of the noisy observed sequences, and formulate the alignment objective through dimensionality reduction (Sec. 3.4) and through clustering (Sec. 4.3). Such formulations include explicit priors that encode our beliefs about the constituent parts of the model. Furthermore, they provide a definition of a generative process of the (aligned) data that

can be used to generate novel content. We compare and contrast the behaviour of the two different alignment objectives on motion capture data as well as on other alignment tasks, and report the quantitative and the qualitative comparisons of the performance of our models against the existing approaches to alignment.

In Ch. 5, we examine the idea of formulating the alignment model as a joint probabilistic model (Sec. 5.1) and, as an alternative, explore the connection between the alignment model and the multi-output GPs (Sec. 5.2).

Chapters 6 and 7 extend the method discussed in Ch. 3 beyond point estimate of the solutions to the alignment problem. More specifically, Ch. 6 focuses on a nonparametric model of monotonic warps by providing an overview of the existing models of monotonic regression and proposing a new Bayesian nonparametric model based on the recent work on GP flows [Hegde et al., 2019]. We compare the performance of the proposed model to some existing approaches on a set of benchmark regression tasks, with an emphasis on the estimates of uncertainty. In Ch. 7, we investigate the uncertainties present in the alignment model and utilise the monotonic random process developed in Ch. 6 to design a more informative alignment model. We demonstrate how the propagation of uncertainty helps uncover structures in the data which are not captured by the existing models (including the approaches proposed in earlier chapters of this thesis).

Chapter 8 concludes the work presented in this thesis and describes potential directions for future work.

1.3 Publications

The work covered in this thesis was developed during an industrial placement with Electronic Arts and was inspired, partially, by problems that arise when prepossessing and utilising motion capture data for character animation and content generation in computer games. This thesis brings together a number of results obtained in collaboration with my supervisor Neill D.F. Campbell, as well as Carl Henrik Ek, Ivan Ustyuzhaninov, Markus Kaiser and Erik Bodin. The main contributions presented in this thesis appeared in the following publications and preprints (in chronological order; * denotes equal contribution):

1. I. Kazlauskaite, C. H. Ek, N. D.F. Campbell (2016). "Learning Alignments from Latent Space Structures". *Workshop on Learning in High Dimensions with Structure at the Conference on Neural Information Processing Systems (NeurIPS*

Workshop 2016).
(Early version of the work in Ch. 3)

2. I. Kazlauskaite, I. Ustyuzhaninov, C.H. Ek, N. D.F. Campbell (2018). "Sequence Alignment with Dirichlet Process Mixtures". *Workshop on Bayesian Nonparametrics at the Conference on Neural Information Processing Systems (BNP@NeurIPS 2018). arXiv:1811.10689.*
(Ch. 4)

3. I. Kazlauskaite, C. H. Ek, N. D.F. Campbell (2019). "Gaussian Process Latent Variable Alignment Learning". *The International Conference on Artificial Intelligence and Statistics (AISTATS 2019). arXiv:1803.02603.*
(Ch. 3)

4. I. Ustyuzhaninov*, I. Kazlauskaite*, C.H. Ek, N. D.F. Campbell (2020). "Monotonic Gaussian Process Flow". *The International Conference on Artificial Intelligence and Statistics (AISTATS 2020). arXiv:1905.12930.*
(Ch. 6 and parts of Ch. 7)

5. I. Ustyuzhaninov*, I. Kazlauskaite*, M. Kaiser, E. Bodin, C.H. Ek, N. D.F. Campbell (2019). "Compositional uncertainty in deep Gaussian processes". *The Conference on Uncertainty in Artificial Intelligence (UAI 2020). arXiv:1909.07698.*
(The main focus of this paper is on deep GP but it shares some of the ideas of compositional uncertainty with the work discussed in Ch. 7)

The central ideas relating to the problem of alignments, the probabilistic alignment objectives and the uncertainty propagation in the resulting models were developed in collaboration with Neill D.F. Campbell and Carl Henrik Ek. The work on monotonic random processes and uncertainty propagation in the alignment model (publications 4 and 5) was developed jointly with Ivan Ustyuzhaninov and the contribution for the resulting papers, including the derivations and the implementation, is shared between the authors. The work on compositional uncertainty (publication 5) further benefited from discussions with Markus Kaiser and Erik Bodin.

2

Preliminaries

This chapter offers a short introduction to Bayesian nonparametrics, and Gaussian processes in particular, as well as an overview of the previous work on temporal alignment of sequences and a discussion of the main building blocks of a generic model for the alignment task.

2.1 Bayesian nonparametrics

Bayesian statistics is a branch of statistics that, in contrast to frequentist statistics, relies on using probabilities to represent degrees of belief. It also has strong axiomatic foundations that rely on probability theory and the interpretation of probabilities which closely resemble the use of the concepts of probability and uncertainty in colloquial language [Barber, 2012]. At the heart of Bayesian modelling is the need to express all uncertainties related to a particular problem by means of probability distributions. More precisely, in a Bayesian setting, parameters are treated as random variables, which relates to our uncertainty about the true values of the (unknown yet fixed) parameters [Bernardo and Smith, 2007].

We cover some of the main ideas of Bayesian modelling, such as Bayesian hierarchies, Bayesian model selection and variational Bayesian inference in the context of Gaussian processes in Sec. 2.1.1 and 2.1.2. Other aspects of Bayesian inference, such as sampling techniques for intractable integrals, are introduced where applicable throughout the thesis.

Bayesian *parametrics* refers to Bayesian methods for prior and posterior distributions in models with a finite and typically low number of parameters while in Bayesian

nonparametrics the finite dimensional prior distributions are replaced with stochastic processes [Hjort et al., 2010]. Therefore, the priors in Bayesian nonparametrics express an infinite number of underlying traits. This implies that the prior beliefs can be expressed using a richer and more flexible collection of distributions than is possible in parametric modelling and the model more easily adapts to the complexity of the data. However, performing predictions at new inputs with a nonparametric method requires access to the training data while in the parametric case the information that the model learns about the structure of the data is contained in the finite set of parameters.

In practice, the field of Bayesian nonparametrics is prevailed by two stochastic processes: the Gaussian process for continuous problems and the Dirichlet process for discrete problems [Broderick, 2014]. We make extensive use of Gaussian processes throughout the work covered in this thesis, therefore, we give a short introduction to these random processes and their sparse approximations in the following sections. Dirichlet processes, which we used as nonparametric priors in a mixture model, are introduced in Ch. 4.

2.1.1 Gaussian processes

This section provides a brief introduction to GPs, mostly following the arguments of [Rasmussen and Williams, 2005]. A Gaussian process (GP) is a random process that can be considered as the infinite-dimensional generalisation to the Gaussian distribution. A GP can be specified by a mean function $m(\mathbf{x})$ and a covariance function $k_\theta(\mathbf{x}, \mathbf{x}')$. The covariance function is parametrised by a set of hyperparameters θ while the mean is often considered as constant zero. The index set of the two functions is infinite which allows GPs to be interpreted as nonparametric priors over the space of functions. Importantly, even though the process is infinite-dimensional, an instantiation of the process is finite and reduces to a Gaussian distribution by definition.

In a regression setting we observe a set of noisy samples $\mathcal{D} = \{\mathbf{x}_j, y_j\}_{j=1}^J$ of a latent function $f(\cdot)$ such that $y_j = f(\mathbf{x}_j) + \varepsilon_j$ where ε_j is i.i.d. Gaussian noise with variance σ_j^2 (which corresponds to a factorised Gaussian likelihood, $p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_j^2 \mathbb{I})$). If we place a zero-mean GP prior over the latent function such that $f(\mathbf{x}) \sim \mathcal{GP}(0, k_\theta(\mathbf{x}, \mathbf{x}'))$, then the instantiations of the function at the training data $\{f_j = f(\mathbf{x}_j)\}_{j=1}^J$ will be Gaussian by definition as $\mathbf{f} \sim \mathcal{N}(0, \mathbf{K}_\theta(\mathbf{x}, \mathbf{x}))$, where $\mathbf{f}$ are concatenations of the function instantiations.

Conditioning and marginalising Due to the self-conjugacy of the Gaussian distribution (where the prior is a multivariate Gaussian and the noise is a factorised Gaussian),

both conditioning and marginalisation can be done in closed form. In particular, consider the joint model of the observations $\mathbf{y}$ and the test values $\mathbf{f}^*$ at test locations $\mathbf{x}^*$:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_\theta(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I} & \mathbf{K}_\theta(\mathbf{x},\mathbf{x}^*) \\ \mathbf{K}_\theta(\mathbf{x}^*,\mathbf{x}) & \mathbf{K}_\theta(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix}\right). \tag{2.1}$$

The conditional distribution of the function values $\mathbf{f}^*$ given the observations $\mathbf{y}$ is:

$$\begin{aligned} \mathbf{f}^*|\mathbf{x},\mathbf{y},\mathbf{x}^* &\sim \mathcal{N}\left(\tilde{\mathbf{f}}^*,\ \tilde{\mathbf{K}}\right), \text{ where} \\ \tilde{\mathbf{f}}^* &= \mathbf{K}\left(\mathbf{x}^*,\mathbf{x}\right)\left[\mathbf{K}(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I}\right]^{-1}\mathbf{y} \\ \tilde{\mathbf{K}} &= \mathbf{K}\left(\mathbf{x}^*,\mathbf{x}^*\right) - \mathbf{K}\left(\mathbf{x}^*,\mathbf{x}\right)\left[\mathbf{K}(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I}\right]^{-1}\mathbf{K}\left(\mathbf{x},\mathbf{x}^*\right) \end{aligned} \tag{2.2}$$

which is the GP predictive posterior at inputs $\mathbf{x}^*$ given the observations $\mathbf{y}$. The marginal distribution can be recovered by finding the relevant part of the covariance matrix; for example, the marginal of $\mathbf{y}$ given $\mathbf{x}$ is:

$$\mathbf{y} \sim \mathcal{N}\left(0, \mathbf{K}_\theta(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I}\right). \tag{2.3}$$

The corresponding log marginal likelihood is:

$$\log p(\mathbf{y} \mid \mathbf{x}, \theta) = \underbrace{-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_\theta(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I}\right)^{-1}\mathbf{y}}_{\text{data fitting term}} \underbrace{-\frac{1}{2}\log|\mathbf{K}_\theta(\mathbf{x},\mathbf{x}) + \sigma_j^2\mathbb{I}|}_{\text{complexity term}} - \frac{N}{2}\log 2\pi, \tag{2.4}$$

which depends on the kernel parameters θ and typically includes additional terms that correspond to the priors over the parameters θ. We will come back to discussing the terms in this likelihood when we talk about model selection.

Covariance (kernel) function The specification of the kernel function and the priors on its parameters (sometimes referred to as the hyperpriors) allow us to impose our beliefs about the characteristics of the data, for example, our assumptions about smoothness or periodicity. In this respect, the choice for the precondition of the model is twofold: we first choose the general functional form or class of the kernel and then choose its parameters (or the priors over these parameters). One popular option is the squared exponential (SE) kernel (also called the exponentiated quadratic or the radial basis

function (RBF) kernel),

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{r^2}{2\lambda^2}\right), \tag{2.5}$$

where $r = ||\mathbf{x} - \mathbf{x}'||$ is the L^2 distance between the inputs. It depends on two parameters, the scaling parameter α and the lengthscale λ. The lengthscale determines how the correlation between two inputs changes with the distance between the inputs. The SE kernel is infinitely differentiable and it is typically used when we have strong smoothness assumptions. An alternative class of kernels that do not make such assumptions are the Matérn kernels [Rasmussen and Williams, 2005]. Typically defined using the modified Bessel function, the Matérn kernels vary greatly in terms of smoothness, for example, Matérn $\nu = 1/2$ (also called the exponential kernel),

$$k_{\nu=1/2}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{r}{2\lambda}\right), \tag{2.6}$$

is used to define the Ornstein-Uhlenbeck process that models the velocity of a particle undergoing Brownian motion, and is hence not smooth. Other popular options include Matérn $\nu = 3/2$ and Matérn $\nu = 5/2$:

$$k_{\nu=3/2}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right),$$
$$k_{\nu=5/2}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}r}{\ell}\right),$$

where the imposed smoothness increases with the value of ν. The Matérn kernels, as well as the SE kernel, are all stationary as they are defined in terms of the distance between the inputs, $r = ||\mathbf{x} - \mathbf{x}'||$, and not the locations of the inputs $\mathbf{x}$. In some practical situations, for example in the context of the alignment of temporal data, a stationary kernel might not be appropriate as the correlations between inputs depend not only on the distances between the points but also on the warping of those inputs. This naturally leads to non-stationary covariance functions defined using a warping function $g(\mathbf{x})$ so that for any given covariance function $k(\cdot, \cdot)$ we consider $k(g(\mathbf{x}), g(\mathbf{x}'))$ where $g(\cdot)$ is some non-linear mapping. There exist multiple different options for such mappings [Rasmussen and Williams, 2005] but for our applications we will consider smooth monotonic increasing functions.

Regularisation As described in [Rasmussen and Williams, 2005], there exists a relationship between the prior in the Bayesian approach and regularisation techniques

that add a penalty to the less favourable solutions (for example, to the solutions that imply a complex model) as they both encode our assumptions such as smoothness. In Bayesian setting, the prior knowledge is contained in a chosen probability distribution while standard regularisation restricts the solution space (typically defined in terms of reproducing kernel Hilbert spaces). Bayesian inference then adjusts the prior belief by learning from the data while regularisation aims to identify the optimal solution that belongs to a chosen space [Ernst et al., 2014].

Model selection In many cases, the observed data can be explained using one of infinitely many different models; this relates to the wider problem of model selection and the principle of Occam's Razor [Rasmussen and Ghahramani, 2000]. In the Bayesian setting, model selection relies on the estimation of the marginal likelihood (also called the evidence) which allows us to choose the simplest model that is able to explain the data [Rasmussen and Williams, 2005]. More generally, to generate a particular data set $\mathcal{D}$ from a specific model $\mathcal{M}$, we sample the parameters/hyperparameters θ of this model from $P(\theta)$, and the data from $P(\mathcal{D} \mid \theta)$. The simplest model generates a lot of similar data sets with high probability while the complex model can produce a lot of different data sets assigning small probability to each data set. Typically, the simplest model cannot fit the data well while the most complex model gives low model evidence as it over-fits the data. In general, the model with the lowest complexity that is able to explain the data is considered to be the optimal choice [Rasmussen and Williams, 2005].

In practice, choosing a model for a particular data set may be done at multiple different levels. At the lowest level we are concerned with placing a GP prior over functions and updating our belief about the functions once the data is observed using Bayes' rule:

$$p(f \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid f)\, p(f)}{p(\mathbf{y})}, \tag{2.7}$$

where $p(f)$ is a GP prior, $p(\mathbf{y} \mid f)$ is a Gaussian likelihood, $p(f \mid \mathbf{y})$ is the GP posterior and $p(\mathbf{y})$ is the marginal likelihood (or the model evidence). The expression for the posterior at new locations $\mathbf{x}^*$ is given in Eq. 2.2 (we call the posterior at new inputs a predictive posterior). To make predictions at test locations $\mathbf{x}^*$, we average over all possible (*i.e.* prior) functions weighted by their posterior probabilities. Meanwhile in a non-Bayesian setting, a single function is chosen using some measure.

Typically, a GP prior $p(f)$ depends on a number of parameters θ, *i.e.* $p(f \mid \theta)$, that correspond to our beliefs about the underlying generative process which are encoded in the kernel function as previously discussed; this corresponds to the second level of

model selection. Handling these parameters in a Bayesian setting involves placing priors on the parameters and averaging over their values. At this level of model selection, the marginal likelihood from the first level acts as the likelihood and the posteriors over the hyperparameters θ (*i.e.* the parameters of the GP) are given by the Bayes' rule:

$$p(\theta \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \theta)\, p(\theta)}{p(\mathbf{y})} = \frac{\int p(\mathbf{y} \mid f)\, p(f \mid \theta)\, p(\theta)\, \mathrm{d}f}{p(\mathbf{y})}. \tag{2.8}$$

Depending on the priors $p(\theta)$, a closed form solution may not be available for this posterior (this is typically the case). While it may be possible to estimate the corresponding integrals using approximation methods (such as Markov Chain Monte Carlo), it is common to rely on point estimates for the hyperparameters θ. The standard way of finding these point estimates involves maximising the likelihood of the observed data $p(\mathbf{y} \mid \theta)$ with respect to the parameter values. If no priors are available for these hyperparameters, then the estimate is called maximum likelihood (ML) while in the presence of priors on the parameters the estimate is referred to as the maximum a posteriori (MAP) estimate. The ML and the MAP estimation contradicts the standard Bayesian approach to model selection where the priors are fixed before the data is observed and the parameters are averaged over based on their posterior probabilities.

Let us return to the discussion of the first level of model selection and analyse the log marginal likelihood of a GP which is given in Eq. 2.4. While for most models the marginal likelihood (over the functions) is not available in closed form, it is analytically tractable in GP models with Gaussian noise model as discussed earlier. Maximising the log marginal likelihood of the observed data with respect to the model parameters automatically imposes the trade-off between the data fitting term and the complexity term; in other words, a complex model is able to fit the data well while a simple model has a low complexity term. Finding the optimal combination of the two gives the simplest model that is able to explain the data. In practice, the model is fitted by maximising the log marginal likelihood w.r.t. the hyperparameters θ using a gradient-based optimisation method.

In some situations, a further step in the Bayesian hierarchy may be required, where a prior is placed on the parameters of the priors of the parameters θ, *i.e.* the hyperprior is defined as $p(\theta \mid \eta)$, and there exists a prior on η. In that case, the original hyperprior may be integrated out using analytic approximations or sampling methods, and the value of η is estimated using maximum likelihood. This would correspond to the third level of model selection. Alternatively, a third level of model selection corresponds to choosing between different families of kernels with parameter η indexing such families.

One possible approach to kernel selection involves estimating the generalisation error on a test set for the different families of kernels and choosing the family with the lowest error [Rasmussen and Williams, 2005].

2.1.2 Sparse GPs and variational inference

The main limitation of GP models is their computational complexity, as the inversion of the kernel matrix $\mathbf{K}_\theta$ has a complexity of $\mathcal{O}(N^3)$, where N is the number of observations, as seen from Eq. 2.4. Most approaches that try to overcome this constraint are based on sparse approximations [Lawrence et al., 2002, Snelson and Ghahramani, 2006, Quinonero Candela and Rasmussen, 2005, Titsias, 2009] where a model is built using a smaller set of points (called the inducing points or pseudo-inputs) of size M such that $M \ll N$. As explained in [Bauer et al., 2016], the two most commonly used methods are the Fully Independent Training Conditional (FITC) [Snelson and Ghahramani, 2006] and the Variational Free Energy (VFE) [Titsias, 2009], both of which approximate the full covariance matrix with a low rank matrix. The main difference between these two approaches is that FITC approximates the model itself while VFE approximates the posterior process. This implies that VFE approximation strictly improves with the number of inducing points and avoids over-fitting, which is not necessarily the case for FITC. Furthermore, as FITC approximates the model, the predictions at new inputs are performed using this approximate model. In our work we utilize the VFE approximation, hence we proceed to give a short introduction to this approach.

Consider the regression setting as described in Sec. 2.1.1, where $\{\mathbf{x}_j, y_j\}_{j=1}^J$ correspond to observation pairs such that $y_j = f(\mathbf{x}_j) + \varepsilon_j$ and $f \sim \mathcal{GP}(0, k(\cdot,\cdot))$. Let us augment the set of observations with M pairs of inducing locations $\{\mathbf{z}_m\}_{m=1}^M$ and the corresponding inducing values $\{u_m\}_{m=1}^M, \mathbf{U} \in \mathbb{R}^M$ (which correspond to the evaluations of the function f at inducing locations $\mathbf{z}_m$). The aim of the VFE approximation is to learn the inducing locations such that the process obtained by conditioning on the inducing variables gives a good approximation to the true posterior process. The inducing points are assumed to be drawn from the same GP prior as the training values:

$$p(\mathbf{f}, \mathbf{U} \mid \mathbf{Z}, \mathbf{X}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{U} \end{bmatrix} \mid \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix}\right), \tag{2.9}$$

where $\mathbf{K}_{ff} = k(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_{fu} = k(\mathbf{X}, \mathbf{Z})$, and $\mathbf{K}_{uu} = k(\mathbf{Z}, \mathbf{Z})$. Then the results on conditional GPs from Eq. 2.2 give the conditional GP prior of the GP evaluations at

the observed points given the inducing values:

$$
\begin{aligned}
p\left(\mathbf{f} \mid \mathbf{U}, \mathbf{X}, \mathbf{Z}\right) &= \mathcal{N}\left(\mathbf{f} \mid \tilde{\boldsymbol{\mu}}, \widetilde{\mathbf{K}}\right), \text{ where} \\
\tilde{\boldsymbol{\mu}} &= \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{U}, \\
\widetilde{\mathbf{K}} &= \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}.
\end{aligned}
\tag{2.10}
$$

The distribution of the inducing values $\mathbf{U}$ follows from Eq. 2.9 by applying the marginalisation property of the Gaussian distribution (as shown in Eq. 2.3) to give:

$$
p\left(\mathbf{U} \mid \mathbf{Z}\right) = \mathcal{N}\left(\mathbf{U}|\mathbf{0}, \mathbf{K}_{uu}\right). \tag{2.11}
$$

Variational inference As the name suggests, VFE employs the variational principle to find the extreme values of a given functional. It is used in Bayesian modelling to handle intractable integrals in order to approximate the posterior of a latent random variable given observed variables, and to approximate the model evidence [Bishop, 2006]. In a general setting this is done by introducing a variational distribution over a set of latent random variables and minimising the distance between the variational distribution and the true posterior distribution using some measure of similarity of distributions. One common choice for such measure is the Kullback - Leibler divergence[1] (KL-divergence) which is defined as:

$$
D_{\mathrm{KL}}(p(x)\|q(x)) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx, \tag{2.12}
$$

where $D_{\mathrm{KL}}(p(x)\|q(x)) \geq 0$ with equality if and only if $p(x) = q(x)$. The objective function used in variational inference is then formulated in one of two alternative (yet equivalent) ways, by either minimising the KL-divergence directly, or by formulating a lower bound on the model evidence (called the evidence lower bound or ELBO). The variational approximation is chosen to have a simpler form than the original posterior distribution to make the computations tractable, yet flexible enough to allow for good approximations of the original distribution. As explained in [Bishop, 2006], it is often convenient to choose a variational distribution that factorises over the latent variables (or groups of latent variables); this is often termed the mean field approximation.

Variational inference for sparse GPs Following the framework of variational inference outlined above, in VFE we introduce a variational distribution $q(\mathbf{f}, \mathbf{U} \mid \mathbf{X}, \mathbf{Z})$ that approximates the true posterior $p(\mathbf{f}, \mathbf{U} \mid \mathbf{y}, \mathbf{X}, \mathbf{Z})$. The variational distribution

[1] Due to the assymetry of KL-divergence, it is not a distance metric.

factorises as $q(\mathbf{f}, \mathbf{U} \mid \mathbf{X}, \mathbf{Z}) = q(\mathbf{f} \mid \mathbf{U}, \mathbf{X}) q(\mathbf{U} \mid \mathbf{Z})$, where the form of the first factor in the variational distribution is chosen to be $q(\mathbf{f} \mid \mathbf{U}, \mathbf{X}) = p(\mathbf{f} \mid \mathbf{U}, \mathbf{X})$ for computational convenience as the $p(\mathbf{f} \mid \mathbf{U}, \mathbf{X})$ term cancels in the computation of the ELBO. Furthermore, assuming that the inducing variables $\mathbf{U}$ are a sufficient statistic for the function values $\mathbf{f}$, which are the noiseless observations, implies that $p(\mathbf{f} \mid \mathbf{U}, \mathbf{X}) = p(\mathbf{f} \mid \mathbf{U}, \mathbf{X}, \mathbf{Y})$. Using this approximation, the ELBO is (where we drop the dependence on $\mathbf{X}$ and $\mathbf{Z}$ to ease the notation):

$$
\begin{aligned}
\log p(\mathbf{y}) &= \log \int p(\mathbf{y}, \mathbf{f}, \mathbf{U}) \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \\
&= \log \int q(\mathbf{f}, \mathbf{U}) \frac{p(\mathbf{y}, \mathbf{f}, \mathbf{U})}{q(\mathbf{f}, \mathbf{U})} \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \\
&= \log \mathbb{E}_{q(\mathbf{f}, \mathbf{U})} \left[\frac{p(\mathbf{y}, \mathbf{f}, \mathbf{U})}{q(\mathbf{f}, \mathbf{U})} \right] \\
&\geq \int q(\mathbf{f}, \mathbf{U}) \log \frac{p(\mathbf{y}, \mathbf{f}, \mathbf{U})}{q(\mathbf{f}, \mathbf{U})} \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \quad \text{using Jensen's inequality} \\
&= \int p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) \log \frac{\cancel{p(\mathbf{f}|\mathbf{U})} p(\mathbf{U}) p(\mathbf{y}|\mathbf{f})}{\cancel{p(\mathbf{f}|\mathbf{U})} q(\mathbf{U})} \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \\
&= \int p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) \log \frac{p(\mathbf{U}) p(\mathbf{y}|\mathbf{f})}{q(\mathbf{U})} \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \\
&= \int p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) \log \frac{p(\mathbf{U})}{q(\mathbf{U})} \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} + \int p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) \log p(\mathbf{y}|\mathbf{f}) \, \mathrm{d}\mathbf{U}\mathrm{d}\mathbf{f} \\
&=: \mathcal{F}(q(\mathbf{U}), \mathbf{Z})),
\end{aligned}
\tag{2.13}
$$

which gives the bound on the model evidence in terms of the variational distribution on the inducing points, $q(\mathbf{U})$, and the inducing locations $\mathbf{Z}$. As noted in [Titsias, 2009], it is possible to find the optimal form of the variational distribution $q(\mathbf{U})$ (and hence maximise this bound) in closed form by assuming a Gaussian form for $q(\mathbf{U})$ and using the self-conjugacy of the Gaussian distributions (for detailed derivations, see, for example, [Damianou, 2015]):

$$
\begin{aligned}
q(\mathbf{U}) &= \mathcal{N}\left(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{K}}\right) \\
\tilde{\boldsymbol{\mu}} &= \sigma^{-2} \mathbf{K}_{uu} \, \boldsymbol{\Sigma} \, \mathbf{K}_{uf} \, \mathbf{y} \\
\tilde{\mathbf{K}} &= \mathbf{K}_{uu} \, \boldsymbol{\Sigma} \, \mathbf{K}_{uu}
\end{aligned}
\tag{2.14}
$$

where $\boldsymbol{\Sigma} = \left(\mathbf{K}_{uu} + \sigma^{-2} \mathbf{K}_{uf} \mathbf{K}_{fu}\right)^{-1}$. The corresponding ELBO is:

$$
\mathcal{F}(\mathbf{Z}) = \log \mathcal{N}\left(\mathbf{y} \mid 0, \sigma^2 \mathbb{I} + \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{K}_{uf}\right) - \frac{1}{2\sigma^2} \operatorname{Tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{K}_{fu}\right). \tag{2.15}
$$

Note that this no longer requires the inversion of the covariance matrix $\mathbf{K}_{ff}$ but instead depends on the inversion of $\mathbf{K}_{uu}$, which is much smaller by construction. The inference is performed by maximising the lower bound w.r.t. the inducing locations $\mathbf{Z}$ (which are the variational parameters) and the parameters of the kernel. The approximate posterior can be used to make predictions at new inputs using the result from Eq. 2.2. Since $\mathbf{Z}$ are variational parameters (*i.e.* parameters of the approximation rather than the model), including additional inducing points always tightens the bound hence improving the approximation.

2.2 Alignment model

We now turn our attention to the main application discussed in this thesis, the task of the temporal alignment of sequences. We start by reviewing the previous literature on the topic, starting with classical techniques, such as Dynamic Time Warping [Berndt and Clifford, 1994], and extending on these with more recent research. We then proceed by introducing a simplified framework which is common for many existing approaches, and which allows us to analyse the alignment task by separately considering the three constituent parts — the model of the sequences, the model of the temporal warps and the alignment objective.

2.2.1 Previous work on temporal alignment

The research on the alignment of sequences comes from multiple different communities, including machine learning, time series analysis and computer graphics. Consequently, the approaches that have been previously proposed vary greatly in terms of the main objectives and the techniques that are used to achieve the specific objective. In this section we aim to categorise and summarise the variety of different approaches.

Pairwise similarity Most approaches that try to learn alignments from data are based on the assumption of the existence of a pairwise similarity measure between the instances of each sequence. The idea is then to find an alignment of the two sequences such that the sum of these distances is minimised. The classical approach to minimise the distance between two sequences is called Dynamic Time Warping (DTW), and is based on computing an affinity matrix of size $N \times M$ where N and M are the lengths of the two sequences to be aligned [Berndt and Clifford, 1994]. The solution corresponds to the path through this matrix that leads to the minimal combined pairwise cost. The

warping path is a sequence $p = (p_1, \ldots, p_L)$ with $p_l = (n_l, m_l) \in [1, N] \times [1, M]$ for $l \in [1, L]$ satisfying the following conditions:

1. Boundary conditions: $p_1 = (1, 1), p_L = (N, M)$.
2. Monotonicity: $n_1 \leq n_2 \leq \cdots \leq n_L$, $m_1 \leq m_2 \leq \cdots \leq m_L$.
3. Step size condition: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1, L - 1]$.

The optimal solution is found by backtracking through the affinity matrix and can be estimated using Dynamic Programming [Müller, 2007]. DTW will find the optimal alignment based on a pairwise distance between each element in two sequences. Such formulation imposes a number of limitations. DTW returns an alignment but not a parametrised warping. Furthermore, it is not trivial to encode a preference towards different warps as this would be a global characteristic while DTW is a local algorithm.

Multiple sequences In its original form DTW only aligns two sequences but there have been several proposed extensions that allow it to process multiple sequences at once, most notably Procrustes dynamic time warping (PDTW), Procrustes derivative dynamic time warping (PDDTW), and Iterative Motion Warping (IMW) [Keogh and Pazzani, 2001], [Dryden and Mardia, 2016], [Hsu et al., 2005]. All of these methods are applied directly in the observation space which is a limitation when the data contains a significant amount of noise. The main algorithms that address this limitation are Canonical Time Warping (CTW) and Generalized Time Warping (GTW) [Zhou and de la Torre, 2009], [Zhou, 2012]. Both of these approaches perform feature extraction and find a subspace that maximises the linear correlation of data samples. Similarly to our approach, GTW is parametrised using monotonic warping functions. However, in all of these methods the spatial alignment and time warping are coupled. Another extension, called Generalized Canonical Time Warping (GCTW) combines canonical correlation analysis, that extracts common features from a pair of multivariate data [Zhou and de al Torre, 2016], with DTW to simultaneously align multiple sequences of multi-modal data [Zhou and de al Torre, 2016]. GCTW relies on additional heuristic energy terms and on coarse-to-fine optimisation to get the energy method to converge to a good local minimum.

Feature extraction More recently, a deep neural network architecture was employed to perform temporal alignments [Trigeorgis et al., 2016], [Trigeorgis et al., 2017]. The proposed method, called Deep Canonical Time Warping (DCTW), is based on

non-linear feature extraction and it performs competitively on large audio-visual data sets. A different method proposed by [Listgarten et al., 2005] uses continuous hidden Markov models, where the latent trace is an underlying representation of the set of observable sequences. [James et al., 2011] introduced hyperalignment that finds isometric transformations of trajectories in voxel space that result in an accurate match of the time-series data. An extension to this model was proposed by [Lorbert and Ramadge, 2012] who address the issues of scalability and feature extraction through the use of the kernel trick. As noted by the authors, the classification accuracy in these methods relies on hand-picked features.

Manifold alignment [Cui et al., 2014] propose an unsupervised manifold alignment method based on finding alignment by enforcing several constraints such as geometry matching, feature matching, geometry preservation and integer constraints. The approach shows promising results but is very computationally expensive. Another non-linear feature extraction method was proposed by [Vu et al., 2012] who construct a k-nearest neighbour graph and then perform DTW to align two sequences, limiting the approach to pairwise comparisons. These methods rely on intelligent feature selection in order to constrain the latent space to produce high quality alignments.

Implicit transformation Another approach to sequence alignment is to use an implicit transformation of the sequences. In [Cuturi et al., 2007, Cuturi, 2011] the authors propose a kernel function that is capable of mapping sequences of different lengths to an implicit feature space. Another similar approach was proposed in [Baisero et al., 2015]. By describing a range of different kernels on sequences, this method allows to learn implicit feature space mappings for sequences of different lengths and different dimensionality. These methods have been shown to work very well experimentally, however, as the alignment now is implicit, we cannot recover the warps or re-align the sequences.

Shape analysis A different line of work, often referred to as elastic registration or shape analysis, is considered in the functional data analysis literature. In [Garreau et al., 2014] the authors propose an extension to DTW by replacing the Euclidean distance with a Mahalanobis distance. By having a parametrisable distance function the authors are able to learn mappings of the distance function from a set of paired observations. [Kurtek et al., 2011] study the group-theoretic approach to warps by constructing the group of warping functions to describe the equivalence relation between signals. In

particular, the authors use the Fisher-Rao Riemannian metric and the resulting geodesic distance to align signals with random warps, scalings and translations. The work is based on using a square-root velocity function (SRVF) to transform the Fisher-Rao metric into the standard L^2 norm [Srivastava et al., 2011, Kurtek et al., 2012]. In subsequent work, [Tucker et al., 2013] proposed a generative model that combines elastic shape analysis of curves and functional principal component analysis (fPCA). Another recent extension called Elastic functional coding relies on trajectory embeddings on Riemannian manifolds and results in a manifold functional variant of PCA [Anirudh et al., 2015]. These methods are sequential in nature: first, the sequences get aligned using a proposed metric, then some statistics are used to summarise the data, and finally a manifold is build in order to generate new sequences.

Shortcomings of existing work To summarise the discussion of the previous work, we point out its main shortcomings, which we aim to address in this thesis.

Noisy observations. The discussed methods either align the observation directly in the input space (this is true for DTW, manifold alignment methods and others) or use feature representations of the observations (DCTW, implicit transformations, *etc.*). In the former case, the observations need to be prepossessed to remove the noise; otherwise, the alignment methods might not detect the common patterns in the observations or overfit to the noise by aligning the spurious similarities in the sequences arising due to the noise. In the latter case, the feature extraction procedure needs to reject the noise and ideally compute the feature of the signal only, which might be hard to guarantee and verify for complex feature extractors (*e.g.* deep neural networks).

Lack of explicit warps and models of sequences. Most of the discussed methods directly compute the correspondences between the sampled values of the observed sequences, without explicitly modelling the warping and the latent functions (g and f in Fig. 1-1). This makes it hard to impose assumptions on the warps and the aligned sequences (*e.g.* continuity or monotonicity). Moreover, the evaluation of the aligned sequences at new (unobserved) input requires interpolating between the observed values which may be performed as a separate post-processing step (*e.g.* using linear or higher order interpolation). On the contrary, explicit models of the warps and the latent functions allow us to circumvent both of these issues.

Rejecting the observational noise naturally requires making assumptions about

the underlying functions (warps and latent functions), linking this point to the necessity of noise rejection mentioned above.

Generation of new sequences. As discussed in Ch. 1, one of the motivations for this work is to generate new sequences that have a similar structure to the observed ones. For example, a new sequence corresponding to the motion (*e.g.* a golf swing or a football kick) in the observed data but performed in a different manner (*i.e.* the same latent function as for the other sequences but with a different unobserved warping). Another example is the generation of a sequence corresponding to a novel motion that is created by interpolating the observed sequence in a certain way. This requires sampling a new latent function that is different from the ones inferred from the data.

The alignment methods reviewed above can only be thought of as a prepossessing step that could be followed by generative modelling. In the following chapters we introduce a procedure for joint alignment and generative modelling, which is based on using probabilistic alignment objectives. Such models allow us to generate new sequences as well as estimate various kinds of uncertainties arising in alignment tasks.

Our work builds on some of the previously proposed themes and objectives. For example, we wish to align multiple sequences, recover the warps explicitly, and capture the low dimensional structure of the data which would allow to generate novel sequences. In contrast to the existing work outlined above, our approach is based on the ideas of Bayesian nonparametrics and offers the benefits of probabilistic learning with explicit priors that scales well with the complexity of the data, as discussed in Ch. 1 and in Sec. 2.1.

2.2.2 Parametric energy-based alignment model

The approaches described in the previous section can be seen as variants of a basic framework that is the same for most alignment tasks. Let us consider the three constituent parts of such a framework: the model of the sequences, the model of the warps and the alignment objective.

Model of sequences We start by observing that the data which we aim to align is discrete, which poses the following problems. As noted in previous work, naively matching the data points of observed sequences is sub-optimal for sequences that are

continuous in nature as many data points in one sequence will collapse to only a few data points in the other sequences [Listgarten et al., 2005]. This behaviour is illustrated in Fig. 2-1. To avoid this behaviour, we may consider interpolating the sequences linearly or with higher order interpolants (*e.g.* cubic splines). This allows us to resample the sequences in a continuous domain to produce unwarped versions of the observed data. However, this does not take into account the noise, i.e. in such a model we interpret the observations as noiseless signals. Furthermore, real-life series data often exhibits features such as rapid oscillations which are hard to capture using linear or cubic splines.

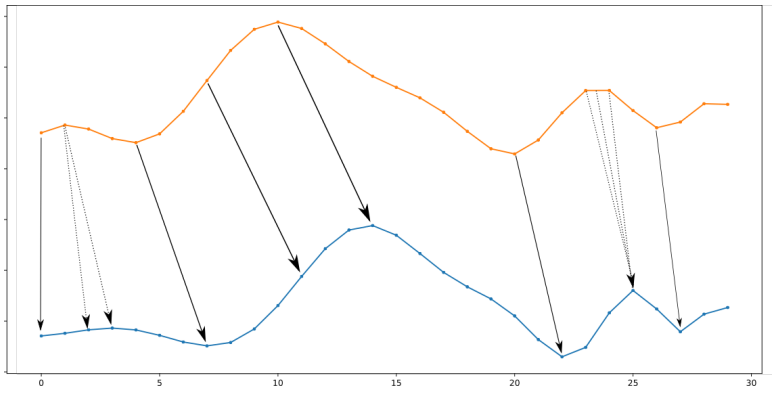


Figure 2-1: Direct matching of two sequences. The arrows show possible matching points. Some of the points in one sequence collapse to only one point in the other sequence (see the dotted arrow at around $x = 25$).

Parametric warps The second part of the framework concerns the modelling of the warpings, i.e. the transformation of the inputs $\mathbf{x}$. In order to avoid permutations of data points $\mathbf{y}$, the warping functions need to be monotonic. One simple approach considered in previous literature uses a parametric warping function defined to be a linear combination of K monotonically increasing basis functions [Zhou and de al Torre, 2016]. Given such set of basis functions, we learn a set of K weights for each input sequence. If the weights are positive and add up to 1, the resulting warping function is guaranteed to be monotonic.

More specifically, for every sequence $\mathbf{y}$ we define the corresponding unwarped sequence as $\hat{\mathbf{y}} = \mathbf{y}\left(\phi(\mathbf{w})\right)$, that is parametrised using a set of K smooth monotonically increasing basis functions, e.g. sigmoid and logarithmic functions, and corresponding weights $\mathbf{w} = \{w_k\}_{k=1}^{K}$. By imposing $\sum_{k \in K} w_k = 1, w_k \geq 0 \; \forall \; k \in K$, we ensure that the resulting mapping g is smooth and monotonic increasing thus allowing the input vectors to be

warped but not permuted. Alternatively, softmax may be used so that

$$w_i = \frac{\exp(a_i)}{\sum_{i'} \exp(a_{i'})} \tag{2.16}$$

where $\{a_i\}$ are unconstrained. The warping function is $\phi(\mathbf{w}) = \sum_{i=1}^{K} b_i w_i = \mathbf{B}\mathbf{w}$, where $\mathbf{B}$ contains the predefined basis functions b_i and $\mathbf{w}$ contains the weights corresponding to sequence $\mathbf{y}$. The alignment is performed as follows; we assume that $\mathbf{y}$ is a continuous function (as explained in the previous paragraph on the model of sequences) and evaluate $\hat{\mathbf{y}} = \mathbf{y}\left(\phi(\mathbf{w})\right)$ where the warpings $\phi(\mathbf{w})$ are real-valued, and the aligned sequence $\hat{\mathbf{y}}$ is calculated using interpolation of $\mathbf{y}$.

Alignment objective One natural formulation for the alignment problem is based on the L^2 metric in the input space which implies minimising the pairwise distances between sequences with respect to the parameters of the model (in this case, the weights of the warping functions). Such a formulation is often referred to as the energy objective. Specifically, given J observed sequences $\{\mathbf{y}_j\}_{j=1}^{J}$ and their corresponding unwarped versions $\{\mathbf{y}_j(\phi(\mathbf{w}_j))\}_{j=1}^{J}$, we are searching for the warping parameters $\tilde{\mathbf{w}}_j$, which minimise the sum of the pairwise distances between the unwarped sequences:

$$(\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_j) = \underset{\mathbf{w}_1,\ldots,\mathbf{w}_J}{\arg\min} \sum_{n=1}^{J} \sum_{m=n+1}^{J} ||\mathbf{y}_n(\phi(\mathbf{w}_n)) - \mathbf{y}_m(\phi(\mathbf{w}_m))||^2. \tag{2.17}$$

If all sequences can be aligned to the same sequence (*i.e.* there is a single underlying function, and observations are the warped versions of it), then the global minimum of Eq. 2.17 exactly corresponds to the sequences being perfectly aligned. There are infinitely many global minima (corresponding to different warpings) and we might encode a preference for one of them by imposing a prior on the coefficients $\mathbf{w}_j$ of the warpings (*e.g.* such that the warpings stay close to identity functions).

The behaviour of the energy alignment objective in the case of multiple underlying functions is more complicated. In this case the observed sequences should be split into groups corresponding to the underlying functions and aligned within these groups. However, the energy objective minimises the distances between all sequences regardless of their group assignments (which are unknown), potentially resulting in all sequences getting aligned to a single one. Yet in practice we impose priors on the warpings, which in some cases prevents all sequences getting aligned to the same one (as it might require extreme warps) and preserves the group structure of aligned sequences. We illustrate the general mechanism of group alignment with energy objective using a toy example.

Toy example: energy alignment objective

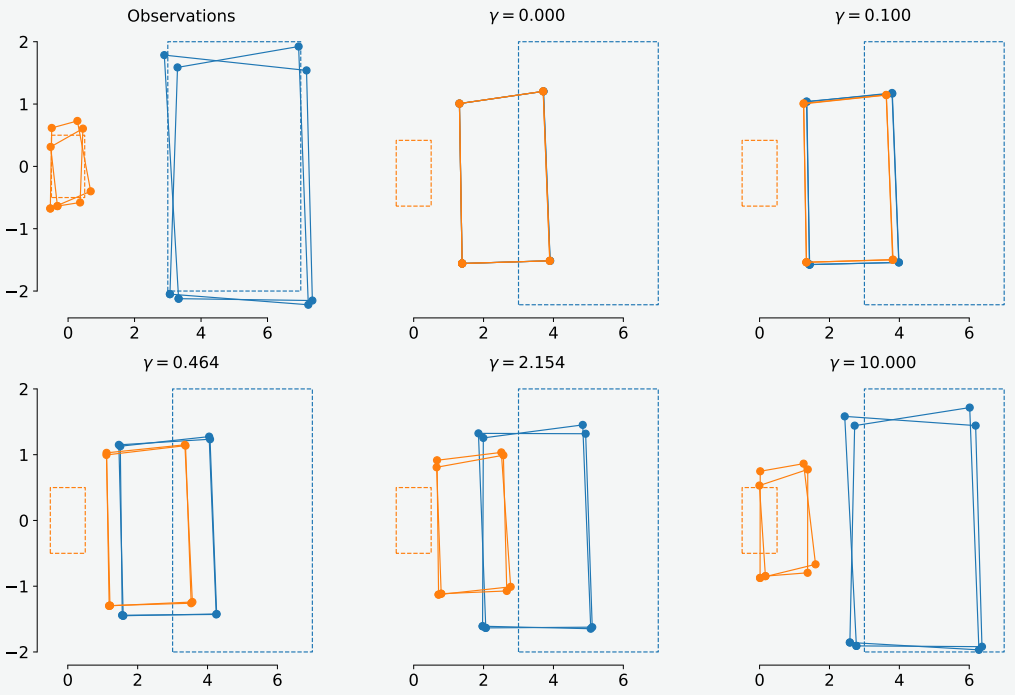


Figure 2-2: Consider the task of aligning four rectangles (shown in solid blue and solid orange in the top left figure) using the energy objective. The 5 cases show the aligned rectangles for different values of the regularisation term (*i.e.* for different values of γ). The low values of transformation prior allow all the rectangles to be aligned together, intermediate values of the prior lead to alignment within clusters (*i.e.* between the rectangles of the same colour) while high values of the prior results in no alignment with clusters and weak global alignment (*i.e.* the two groups of rectangles being pushed closer together).

To analyse the effect of the alignment objective in isolation (*i.e.* independently from the fitting of the sequences and the warping functions), consider the following simple example. Assume that we are given four quadrilaterals in $\mathbb{R}^2$ (two in orange and two in blue Fig. 2-2 (Observations)), which are noisy versions of two underlying rectangles (that play the role of sequences), shown in dashed lines. Each such quadrilateral is parametrised by 8-dim vectors consisting of coordinates of its vertices ($\mathbf{y} = (x_1, y_1, \ldots, x_4, y_4)$), and we define the aligning transformations (warpings) as independent additive transformations of the vertices: $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{w}$ with $\mathbf{w} = (w_1, w_2, \ldots, w_7, w_8)$. Given this set up, we apply the alignment objective to the four objects (the noisy rectangles) in this problem; in particular, we use the energy objective Eq. 2.17 to align these observations. Furthermore, we impose an

L^2 regularisation for the transformations to be small ($\gamma\,||\mathbf{w}||^2$). This regularisation term corresponds to the constraints on the warping functions to be close to an identity. Fig. 2-2 shows the alignment results for different values of γ.
Without the prior on the transformations (*i.e.* with $\gamma = 0$), all observations get aligned to the same sequence. However, putting a prior on the transformations allows us to shift a global optimum of the alignment objective to a point where the group structure is preserved. This optimum corresponds to the sequences within the same group being aligned (roughly $\gamma = 0.4$), though the alignment is not completely precise because there is a *global* component in the objective minimising the distance to sequences of the other group. More precisely, there is a trade-off between the quality of alignment within each group and the separation of the groups (stronger transformation priors better preserve the group structure, but deteriorate within-group alignments).

This toy example demonstrates that the energy objective can in principle be used for multi-group alignment, provided that the warpings are appropriately constrained to preserve the group structure. However, in practice finding the warping constraints allowing both good within-group alignments and good separation of the groups can be hard as previously discussed and demonstrated in Fig. 2-1. We address this problem in the following chapters by introducing probabilistic alignment objectives which overcome such limitations.

3

GP-GPLVM alignment model

The model defined in Sec. 2.2.2 offers an elementary approach to the alignment problem, and it requires compromise on every part of the model as previously explained. In this chapter we take a different approach where we encapsulate the alignment and the modelling of the sequences within a single framework. By simultaneously modelling the sequences and the alignment we capture global structure thereby circumventing the difficulties associated with an objective function based on pairwise similarity of sequences. In addition, differently from the commonly used approach of interpolating between observations, we impose GP priors on the underlying functions and the warps, which jointly regularise the solution space of the ill-posed alignment problem.

In this chapter we explore the use of a non-linear latent variable model, specifically a GP latent variable model (GP-LVM), as a probabilistic alignment objective. The choice of such model as the alignment objective is motivated by the desire to automatically infer the groups of sequences in the data set which should be aligned to the same sequence. The latent variable dimensionality reduction methods (such as GP-LVM) are well-suited for this task as they preserve structure in the observation space while imposing the preference for dissimilar data points to be placed far apart in the latent space.

The model proposed in this chapter overcomes a number of problems with the elementary approach discussed in Sec. 2.2.2, as well as the existing literature, and confers three main contributions:

1. We model the observed data directly with a generative process, rather than interpolate between observations; that allows us to reject noise in a principled manner.

2. The generative model of the data allows us to align and cluster observed sequences

in a fully unsupervised manner.

3. We use continuous, nonparametric processes to explicitly model the warping functions throughout; this allows the specification of sensible priors rather than unintuitive or heuristic choices of parametrisations.

3.1 Background

Our model consists of two main probabilistic components: Gaussian processes as priors over warpings and latent functions, as well as the Gaussian process latent variable model (GP-LVM) as the alignment objective. As discussed in Sec. 2.1.1, a GP is a random process that serves as a nonparametric prior over the space of functions [Rasmussen and Williams, 2005]. In this section we provide a detailed discussion of the construction and main features of the GP-LVM enabling us to use it as an alignment objective.

3.1.1 Gaussian process latent variable model

Following [Lawrence, 2005], we present the GP-LVM as a non-linear probabilistic formulation of the principal component analysis (PCA). The objective of both PCA and GP-LVM is to represent a high-dimensional data set $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^{J}$ that consists of J N-dimensional data points by a lower dimensional embedding $\mathbf{Q}$, where the dimension of $\mathbf{Q}$ is $Q \leq N$. In such models $\mathbf{Y}$ is the observed variable while $\mathbf{Q}$ is the latent variable. Let us assume that the observations $\mathbf{Y}$ and latent variables $\mathbf{Q}$ are related by a linear mapping $\mathbf{Y} = \mathbf{Q}\mathbf{W}^T$. Assuming the Gaussian noise model, the probability of the high-dimensional data given the mapping and the low dimensional embedding is:

$$p(\mathbf{Y} \mid \mathbf{Q}, \mathbf{W}) = \prod_{j=1}^{J} \mathcal{N}(\mathbf{y}_j \mid \mathbf{W}\mathbf{q}_j, \beta^{-1}\mathbb{I}). \tag{3.1}$$

Hence each high-dimensional data point is distributed normally with a mean equal to the linear mapping of the corresponding low-dimensional data point, and a noise equal to $\beta^{-1}\mathbb{I}$. Let the prior distribution of the latent variables be the standard normal distribution, $p(\mathbf{Q}) = \prod_{j=1}^{J} \mathcal{N}(\mathbf{0}, \mathbb{I})$. [Tipping and Bishop, 1999] have shown that such a model gives a probabilistic formulation of the PCA.

Marginalising out the latent variables leads to the following distribution:

$$p(\mathbf{Y} \mid \mathbf{W}) = \prod_{j=1}^{J} \mathcal{N}(\mathbf{y}_j \mid \mathbf{0}, \mathbf{W}\mathbf{W}^T + \beta^{-1}\mathbb{I}). \tag{3.2}$$

The corresponding log-likelihood is:

$$\begin{aligned} \log p(\mathbf{Y} \mid \mathbf{W}) = &-\frac{J}{2} \log |\mathbf{W}\mathbf{W}^T + \beta^{-1}\mathbb{I}| \\ &- \frac{1}{2}\mathrm{Tr}\left((\mathbf{W}\mathbf{W}^T + \beta^{-1}\mathbb{I})^{-1}\mathbf{Y}^T\mathbf{Y}\right) + \text{const}, \end{aligned} \tag{3.3}$$

and we can find the maximum likelihood solution with respect to the mapping $\mathbf{W}$ and the noise parameter β.

Recall that the traditional (non-probabilistic) PCA tries to find a set of orthonormal axes, called the principal axes, which under projection preserve the highest variance. When formulated as an eigenvalue problem, the principal axes of PCA correspond to the eigenvectors of the sample covariance matrix with the largest eigenvalues [Tipping and Bishop, 1999]. The argument $\mathbf{W}$ that maximises Eq. 3.3 is $\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_Q\mathbf{L}\mathbf{V}^T$, where $\mathbf{U}_Q \in \mathbb{R}^{J\times Q}$ are the first Q eigenvectors of the sample covariance matrix with corresponding eigenvalues $\{\lambda_q\}_{q=1}^{Q}$, $\mathbf{L} = (\mathbf{\Lambda}_Q - \beta^{-1}\mathbb{I})^{1/2}$ where $\mathbf{\Lambda}_Q$ is a diagonal matrix of eigenvalues, and $\mathbf{V} \in \mathbb{R}^{Q\times Q}$ is an arbitrary orthogonal matrix. In the limit $\beta \to \infty$, the maximum likelihood solution implies the PCA solution.

One limitation of the above formulation is that it requires the mapping $\mathbf{W}$ to be linear, however, in general, the high-dimensional data $\mathbf{Y}$ and the low-dimensional embedding $\mathbf{Q}$ may be related non-linearly. Unfortunately, the treatment of non-linear mappings of probability distributions is not straightforward. The usual approach to this problem is to sample from the prior distribution and to map only the sampled points; this gives a posterior-mean projection [MacKay, 1995], [Bishop et al., 1997].

The GP-LVM overcomes the issues with non-linear maps by reformulating the objective of the problem; instead of marginalising the latent variables and maximising the likelihood of $\mathbf{Y}$ with respect to the mapping, in the GP-LVM we marginalise out the mapping and maximise the likelihood with respect to the latent variables. In this dual formulation a standard normal prior distribution is placed on each dimension of the mapping $\mathbf{W}$ (alternatively, consider $\mathbf{W}$ as a matrix, and place independent Gaussian priors on each row of this matrix). Due to the choice of the conjugate prior for $p(\mathbf{Y} \mid \mathbf{Q}, \mathbf{W})$, Eq. 3.1,

the marginalisation of $\mathbf{W}$ can be performed in closed form:

$$p(\mathbf{Y} \mid \mathbf{Q}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n \mid \mathbf{0}, \mathbf{Q}\mathbf{Q}^T + \beta^{-1}\mathbb{I}). \tag{3.4}$$

Note that while in Eq. 3.2 a normal distribution is assigned to each data point (product over j), in Eq 3.4 the normal distributions correspond to the dimensions of the observed data (product over n). The log-likelihood of this dual formulation is:

$$\log(p(\mathbf{Y} \mid \mathbf{Q})) = \underbrace{-\frac{J}{2}\log|\mathbf{K}_Q|}_{\text{complexity term}} \underbrace{-\frac{1}{2}\mathrm{Tr}\left(\mathbf{K}_Q^{-1}\mathbf{Y}\mathbf{Y}^T\right)}_{\text{data fitting term}} + \underbrace{\sum_i \log \beta_i}_{\text{prior over hyperparameter}} + \text{const}, \tag{3.5}$$

where $\mathbf{K}_Q = \mathbf{Q}\mathbf{Q}^T + \beta^{-1}\mathbb{I}$. We maximise the log-likelihood w.r.t. the latent variables $\mathbf{Q}$.

[Lawrence, 2005] observed that the dual formulation of the probabilistic PCA results in a model that is closely related to Gaussian processes. Given a set of inputs, a sample from a GP prior has the following distribution:

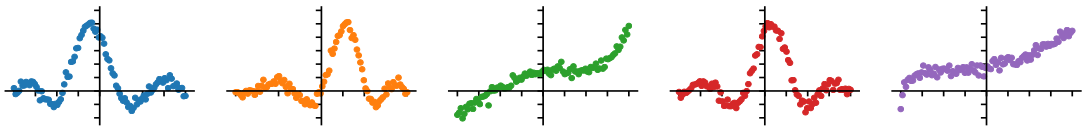
$$p(\mathbf{Y} \mid \mathbf{X}) = \mathcal{N}(\mathbf{Y} \mid \mathbf{0}, \mathbf{K}), \tag{3.6}$$

where $\mathbf{K}$ is the covariance matrix. Setting the covariance matrix to $\mathbf{K} = \mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbb{I}$ and combining the probability densities for all the dimensions of $\mathbf{Y}$, we recover the dual formulation of the probabilistic PCA. The advantage of such a formulation is that the linear kernel of probabilistic PCA may be replaced by a non-linear one by choosing $\mathbf{K}$ to be an arbitrary positive semi-definite kernel. One popular example is the radial basis function kernel (defined in Eq. 2.5), which provides a non-linear mapping from the latent space to the data space. Such a mapping is probabilistic and the uncertainty is in the mapping, as opposed to the PCA formulation.

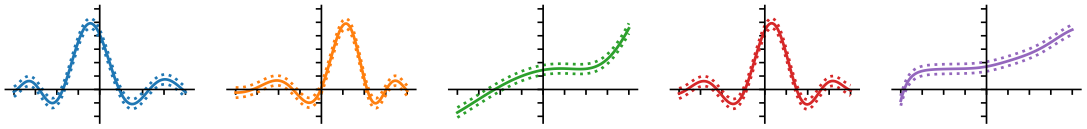
Due to its probabilistic nature, the GP-LVM makes it easy to generate new data by sampling from the latent space. Moreover, missing data and noise are both easily handled. However, the inversion of the kernel matrix implies that the complexity of the algorithm is $O(N^3)$, which makes the optimisation computationally expensive. Another drawback is that the solution for the latent variables is not unique, which results in multiple local minima.

We will come back to discussing the GP-LVM in Sec. 3.4 where we propose using it as a regulariser for the alignment task.

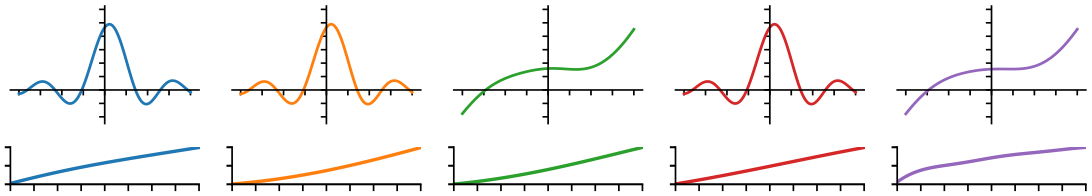
3.2 Overview of alignment task



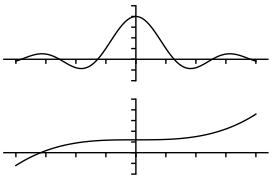
(a) The observed data $\mathbf{y}_j$, $j = 1 \dots 5$, are noisy and time warped versions of the unknown true sequences.



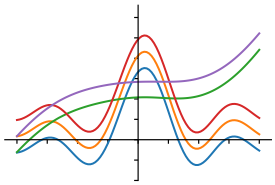
(b) Continuous functions are fitted (with uncertainty) to the noisy, time warped observations (these are $f_j(g_j(\mathbf{x}))$).



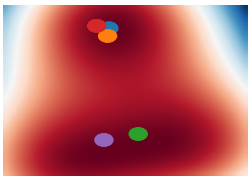
(c) These are decomposed into the aligned sequences $f_j(\mathbf{x})$ (top row) and the inferred continuous warp $g_j(\mathbf{x})$ (bottom row).



(d) The set of two true unobserved sequences for the warped observations.



(e) The inferred aligned results (shown offset in the y-axis).



(f) The inferred generative manifold clusters the two types of sequences.

Figure 3-1: Overview of our model on a toy example. We are presented with a set of noisy observations (a) that we assume to be time warped versions from a set of latent sequences (d). We fit continuous functions (b) to the observations and then decompose and cluster them into aligned versions with continuous time warps (c). This results in a generative model where the aligned sequences (e) are produced from a manifold (f) that reveals the clustering of the observations into the two latent sequences of (d).

Alignment learning is the task of recovering a set of monotonic warping functions, that have been used to corrupt a latent function, from a set of observed sequences. Fig. 3-1 provides an overview of the setting we consider. We are provided with a number of noisy time warped observations of a set of unobserved latent functions and our task is to infer both this set of sequences and the time warps that give rise to the observations.

Let us assume that we have J noisy sequence observations $\{\mathbf{y}_j\}$ (Fig. 3-1a) where each observed sequence comprises N time samples, $\mathbf{Y} = (y_{jn}) \in \mathbb{R}^{J\times N}$. We consider each sequence to be generated as a sample from a latent function $f_j(\mathbf{x})$ (Fig. 3-1b) under a monotonic warping $g_j(\mathbf{x})$ as $y_{jn} = f_j(\, g_j(x_n)) + \varepsilon_{jn}$ where the samples have been corrupted by additive Gaussian noise $\varepsilon_{jn} \sim \mathcal{N}(0, \beta_j^{-1})$. Due to the close association of sequences and temporal data, we use the word time to refer to the input domain of the sequence, however our method is general and applicable to any ordered index set.

The aligned sequences (which may be noise-corrupted), which are unobserved, are given by the corresponding functions without the time warp $s_{jn} = f_j(x_n) + \varepsilon_{jn}$, $\mathbf{S} = (s_{jn}) \in \mathbb{R}^{J\times N}$ (as illustrated in Fig. 3-1d and Fig. 3-1c). This means that we can encode our warping function as the transformation from a *known* sampling of an *unknown* aligned sequence to the *unknown* sampling of the *known* observations for each sequence. However, as described in the introduction and illustrated in Fig. 3-1d, we wish to design a model that is not restricted to the case where all the observations arise from a *single* latent function (for example, in Fig. 3-1d there are two unknown true sequences).

To account for the possible existence of multiple latent functions, we consider a generative model for the aligned sequences themselves. By specifying that the generative process be as simple as possible, we encourage the clustering of these sequences, which allows us to automatically find the smallest number of latent functions explaining the data. We encode this as the aligned sequences being generated via a smooth mapping $h(\cdot)$ from the collection of locations $\mathbf{Q} \in \mathbb{R}^{J\times Q}$ in the low dimensional space as

$$\mathbf{s}_j = h(\mathbf{q}_j) + \hat{\epsilon} \quad \text{s.t.} \quad \mathbf{s}_j = f_j(\mathbf{x}) + \epsilon_j \; \forall \; j \tag{3.7}$$

where $\hat{\epsilon} \sim \mathcal{N}\big(0, \gamma^{-1}\mathbb{I}\big)$ and $\epsilon_j \sim \mathcal{N}\Big(0, \beta_j^{-1}\mathbb{I}\Big)$. This low dimensional manifold is visualised, for our toy example, in Fig. 3-1f where the locations of the aligned sequences $\mathbf{s}_j$ are shown as coloured points matching the corresponding aligned sequences in Fig. 3-1e. We see that the two different sequences are clustered appropriately by their location in the manifold and the sequences are correctly aligned. We use a probabilistic model of the aligned sequences, which allows us to quantify uncertainty of the low-dimensional manifold representations (the heatmap in Fig. 3-1f).

Given noisy observed data, we do not impose the constraint in Eq. 3.7 exactly, but rather define both model components separately and interpret this constraint as one of the aligned sequences having high likelihood under both model components simultaneously. If the aligned sequences $\mathbf{S}$ were known, this interpretation would correspond exactly to fitting a model to the observed data by maximising the data likelihood. Since $\mathbf{S}$ are

unobserved, we refer to them as *pseudo-observations*; similarly to [Titsias, 2009], we augment the probability space with a set of pseudo-observations which are constrained by the two components of our model. We then fit the model by maximising the joint marginal likelihood of observations $\mathbf{Y}$ and pseudo-observations $\mathbf{S}$, while optimising not only w.r.t. the model parameters, but also w.r.t. the pseudo-observations $\mathbf{S}$.

It is clear from this description that the task of sequence alignment is underconstrained and there exists no ground truth solution to the alignment problem, as previously discussed in the Evaluation section in Ch. 1. Typically the observed sequences are aligned by finding the corresponding latent functions with the minimal value of pairwise distances between them. However, such pairwise distance metric can be severely misleading as it does not take into account the complexity of the corresponding warping functions, the potential of multiple underlying latent functions (*e.g.* sinc and cube in Fig. 3-1), nor the existence of multiple equally favourable solutions under this metric. This makes model selection a central issue when talking about alignments, and motivates the use of Bayesian nonparametrics in order to design models that incorporate the prior knowledge about the parts of the model and are able to adapt to the complexity of the data.

3.3 Model of observations and warps

As explained in the previous section, our alignment model consists of two components that correspond to the constraint introduced in Eq. 3.7. The first part corresponds to fitting the data that explains the observed sequences and specifies the latent functions, while the second part enforces a simple, low-dimensional structure of the aligned sequences. In this section, we will discuss the first part of this model, consequently extending and improving the basic approach described in Sec. 2.2.2.

Model over time In time series modelling generally, it is assumed that the data is generated by an unknown function and the observations contain independent measurement noise. In the context of sequence alignment, it is, furthermore, important to be able to resample the underlying functions at new input locations in order to generate warped versions of the sequences, as discussed in Sec. 2.2.2. Standard interpolation methods are not able to deal with noisy observations and missing data in a principled manner, or to incorporate prior assumptions about the structure of the data. As explained in Sec. 2.1.1, GPs address both of these issues by offering probabilistic outputs that incorporate high-level assumptions about the smoothness of the underlying functions.

Consequently, GP priors are a natural choice when it comes to modelling the sequences.

In particular, we have $\mathbf{y}_j \in \mathbb{R}^N$ as the observed sequences, and let $\mathbf{x} \in \mathbb{R}^N$ denote an observed uniform sampling of time. We introduce a random vector $G_j = g_j(\mathbf{x})$ to denote the value of the time warp function g_j evaluated at inputs $\mathbf{x}$. The realisation of this random vector is denoted $\mathbf{g}_j \sim G_j$ with $\mathbf{g}_j \in \mathbb{R}^N$. The random vectors corresponding to the latent function f_j are more involved since this function is evaluated at different locations. We denote the evaluation of f_j at the warped inputs as $F_j^G = f_j(G_j) = f_j(g_j(\mathbf{x}))$; the realisation of this random vector is denoted as $\mathbf{f}_j^G \sim F_j^G$ with $\mathbf{f}_j^G \in \mathbb{R}^N$. Similarly, we use $F_j^X = f_j(\mathbf{x})$ to denote the function evaluated at the uniform sample locations $\mathbf{x}$ with the realisation of this random vector being $\mathbf{f}_j^X \sim F_j^X$, $\mathbf{f}_j^X \in \mathbb{R}^N$. The observations $\mathbf{y}_j$ are the noise-corrupted versions of $\mathbf{f}_j^G$, and similarly, we call the noise-corrupted version of $\mathbf{f}_j^X$ pseudo-observations, since they are not observed and should be inferred.

We now define the priors over the generating and warping functions $f_j(\cdot)$ and $g_j(\cdot)$. Specifying a parametric mapping is challenging and it severely limits the possible functions we can recover. We make use of GP priors which allow us to provide significant structure to the learning problem without reducing the possible solution space. The two random variables connected with $f_j(\cdot)$ may then be jointly specified under a GP prior where the covariance, with hyperparameters θ, is evaluated at $\mathbf{g}_j$ and $\mathbf{x}$ for $\mathbf{f}_j^G$ and $\mathbf{f}_j^X$ respectively as

$$p\left(\begin{bmatrix}\mathbf{f}_j^X \\ \mathbf{f}_j^G\end{bmatrix}\middle|\, \mathbf{g}_j, \mathbf{x}, \theta_j\right) \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k_{\theta_j}(\mathbf{x},\mathbf{x}) & k_{\theta_j}(\mathbf{x},\mathbf{g}_j) \\ k_{\theta_j}(\mathbf{g}_j,\mathbf{x}) & k_{\theta_j}(\mathbf{g}_j,\mathbf{g}_j)\end{bmatrix}\right). \tag{3.8}$$

Warping functions We encode our preference for smooth warping functions $g_j(\cdot)$ by placing a GP prior with a smooth kernel function. However, the warping functions must be monotonic, while the functions distributed according to a GP prior are not necessarily monotonic. We address this issue by explicitly constraining the warpings to be monotonic by an appropriate parametrisation, and placing a GP prior on the parametrised monotonic functions to ensure their smoothness.

Specifically, we use the following approach. Without loss of generality, $\mathbf{g}_j$ are constrained to be monotonic in the range $[-1, 1]$ using a set of auxiliary variables $\mathbf{v}_j \in \mathbb{R}^N$ such that

$$[\mathbf{g}_j]_n := 2\sum_{k=1}^{n}[\operatorname{softmax}(\mathbf{v}_j)]_k - 1\ . \tag{3.9}$$

We place a GP prior with smooth covariance function on the reparametrised $\mathbf{g}_j$, and optimise the auxiliary variables $\mathbf{v}_j$ to be a MAP estimate under this prior (see further discussion in Sec. 3.5). Additionally, we place a zero-mean Gaussian prior over $\mathbf{v}_j$ to encode our preference for the warping function to be close to identity.

The warping functions are not generative in this case because they are explicitly parametrised by $\mathbf{v}_j$, and we do not have a joint generative model over the prior on $\mathbf{v}_j$ and the GP prior. We address this limitation in further chapters (see Ch. 6) by introducing a nonparametric probabilistic model of monotonic functions.

This uncertainty in the alignment problem stems from the fact that it is under-constrained and so there are generally multiple possible ways to align a group of given sequences. When choosing a model for the warpings we need to take into account the fact that the random process that generates the warpings needs to be monotonic. While there exist some methods that satisfy this constraint, for example, GPs with monotonicity information [Riihimäki and Vehtari, 2010], the propagation of uncertainty through a model that is composed of two random processes poses further questions related to exact interpretation of uncertainty in such a framework. We discuss the alternatives approaches and propose a novel Bayesian nonparametrics construction for monotonic processes based on GP flows [Hegde et al., 2019] in Ch. 6; we then discuss uncertainty in a two-layer model and the implications of using such a model for the alignment task in Ch. 7.

3.4 Alignment objective

The second component of the model is the alignment objective, which corresponds to the constraint introduced in Eq. 3.7. We would like to define an objective that aligns similar sequences to each other while keeping dissimilar sequences apart without us specifying which sequences belong together. In Sec. 2.2.2 we discussed an energy alignment objective based on minimising pairwise distances between aligned observations. Such an objective allows us to align sequences corresponding to multiple latent functions, but that is a by-product of specific regularisation of the warping functions rather than a principled construction allowing alignment in multiple groups. In this section we address this issue.

We propose using the GP-LVM (Sec. 3.1.1) as an alignment objective. Under such an objective each dimension of the aligned sequences is modelled as an output of a GP (same GP for all dimensions) at inputs in some low-dimensional latent space (denoted

by $\mathbf{Z}$ in Sec. 3.1.1). Therefore, there is a point in the latent space corresponding to each aligned sequence. The likelihood of these GPs is optimised w.r.t. the warping parameters and the latent space points corresponding to the aligned sequences. If the GP-LVM outputs (the aligned sequences in our case) were fixed, such an objective would correspond to a non-linear dimensionality reduction method finding a latent space representation of the data. However, the outputs of the GP-LVM, which are the fixed observations in a dimensionality reduction setting, are not fixed in our case, and hence we refer to them as pseudo-observations. Jointly optimising these pseudo-observations and the latent space has an aligning effect.

There are two main GP-LVM components responsible for this aligning effect: (1) a zero-mean Gaussian prior in the latent space, and (2) a stationary kernel for GPs used to map the latent space to the aligned sequences. First, let us discuss the case of only one group of sequences (*i.e.* all observations need to be aligned to a single sequence), in which the global optimum of the objective corresponds to all the sequences being aligned. Indeed, the optimal configuration under the Gaussian prior in the latent space is all points being placed at the mean, meaning that the inputs for the GPs outputting the aligned sequences are the same for all dimensions of the sequences. The highest likelihood in this case corresponds to the aligned sequences exactly coinciding in each dimension, *i.e.* being aligned. This argument depends on the priors on the warpings: typically it is beneficial to discourage extreme and potentially non-smooth warpings as they are less interpretable. However, aligning the sequences perfectly might require strong warpings with low probability under the prior. For simplicity and to illustrate the GP-LVM aligning mechanism separately from the warping prior, we assume that the warpings required to align the sequences have a high enough probability to not change the optimum of the alignment objective.

Next, let us discuss the more interesting case of multiple groups of sequences. In this case the points in the latent space cannot all be placed at zero as otherwise the outputs of GPs corresponding to different aligned sequences should be the same (since the inputs coincide), but the sequences cannot all be aligned resulting in a mismatch between the GP-LVM outputs and the aligned sequences leading to small likelihood values. The optimal latent space configuration then corresponds to the coinciding points associated with the sequences of the same group, while the points associated with different groups are pushed apart. We note that this is essentially the energy alignment applied in the latent space: given two non-coinciding points of the same group, one of them is closer to zero (*i.e.* it is more likely under the latent space prior), and moving the second one closer to the first one increases the probability of it under the prior.

We now consider the effect of the stationary GP kernel in the GP-LVM. As discussed in Sec. 2.1.1, stationary kernels depend only on the distance between the inputs, which means that the further each input is from the others, the less correlated are the GP outputs at these inputs. Such behaviour is characterised by a kernel length-scale: if the inputs are far from each other with respect to the length-scale, then their outputs are weakly correlated. Returning to the alignment argument, the points of the same group are close together in the latent space while points of different groups are far apart. If the length-scale is smaller than the distance between points of different groups, then the corresponding GP outputs at the inputs of the different groups are weakly correlated. Hence the problem reduces to warping the sequences of the same group in such a way that they have high likelihood at the coinciding inputs, which as we discussed above leads to these sequences getting aligned.

To summarise, the GP-LVM objective essentially uses energy alignment in the low-dimensional latent space, while the stationary kernel allows us to translate the aligned latent space points to the aligned sequences within each group. The crucial difference to directly aligning the sequences with the energy objective is that it acts globally, and hence it tries to bring the aligned sequences of different groups closer to each other (as much as possible given the priors on the warpings) compromising on the within-group alignment (see Fig. 2-2). Meanwhile, the kernel length-scale allows us to decorrelate the aligned sequences of different groups resulting in better within-group alignments as illustrated in the toy example below.

Toy example: GP-LVM alignment objective

We use the same toy example as in Fig. 2-2 to illustrate the alignment properties of the GP-LVM. We use a 2D latent space and GPs with squared-exponential kernel as in Eq. 2.5 with the kernel hyperparameters (scale α and length-scale λ) both fixed to 1.

Similar to Fig. 2-2, we observe that a stronger prior on the warpings results in greater separation of the groups and the aligned sequences being closer to the observations. The key difference to the energy alignment example is that the sequences within the groups are well aligned (they practically coincide). That is a consequence of the GP-LVM finite length-scale, as discussed in the main text, and illustrated by the latent space plots. The points corresponding to different groups are separated by a distance significantly larger than 1 (which is the fixed length-scale in this example) in accordance with our alignment arguments outlined above.

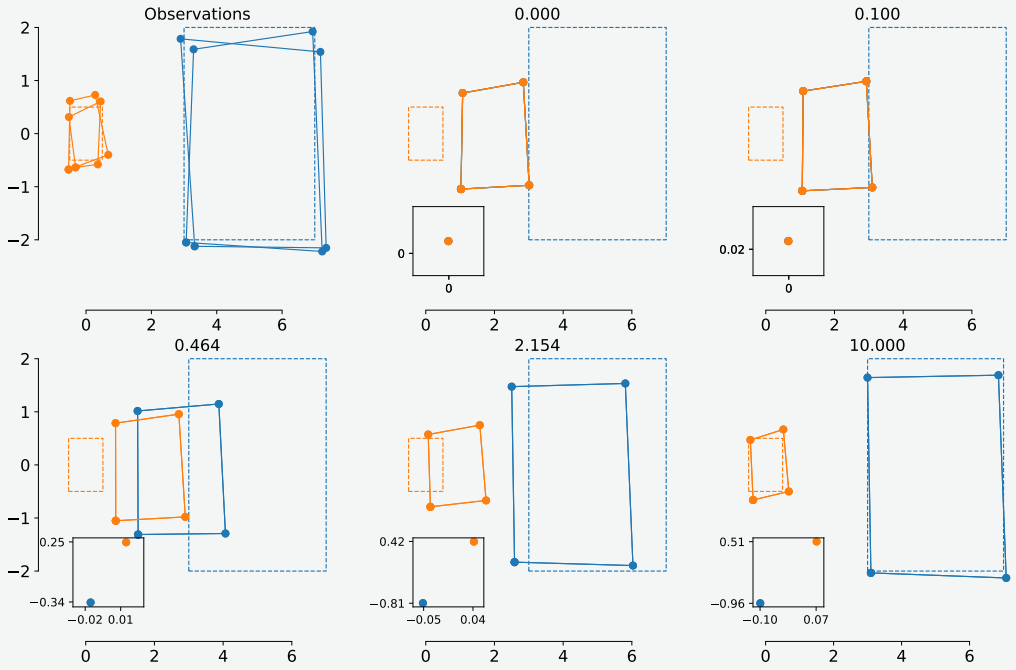


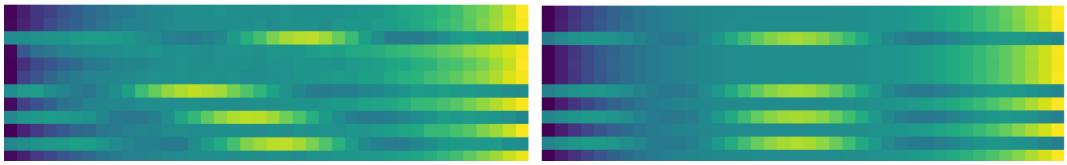
Figure 3-2: Aligned quadrilaterals for different values of the warping prior γ. Small scatter plots in lower-left corners show the 2D GP-LVM latent spaces. As the value of γ increases, the effect of global alignment (*i.e.* alignment of all four rectangles together) decreases while the alignment within clusters of the same colour remains consistent.

In the next part of this section, we formally define the GP-LVM alignment objective in our setting of temporal sequence alignment.

3.4.1 Model over sequences

We let the random variable $\mathbf{q}_j \in \mathbb{R}^Q$ be the embedded manifold (the latent space) location of the sequence. The random variable H_j denotes the output of the mapping function evaluated at $\mathbf{q}_j$ such that $H_j \sim h(\mathbf{q}_j)$ (to emphasise that the realisation of this variable is a vector in $\mathbb{R}^N$, we alternatively denote it as $\mathbf{h}_j$). To ease notation, we use bold symbols to denote the concatenation across J such that, for example, $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_j]$. We encode the preference for a smooth mapping by placing a GP prior over the mapping $h(\cdot)$ so that we have $p(\mathbf{H} \mid \mathbf{Q}, \psi_h) \sim \mathcal{N}(0, k_{\psi_h}(\mathbf{Q}, \mathbf{Q}))$ where ψ_h are the hyperparameters of the covariance kernel. The pseudo-observations are modelled by the GP-LVM by adding independent Gaussian noise to $\mathbf{H}$. Next we consider the joint distribution of the model to derive an objective which simultaneously ensures

that (i) the observed data $\mathbf{Y}$ is fitted well by the corresponding GPs f at the warped locations, (ii) the pseudo observations are fitted well by the corresponding GPs f at the fixed sampling locations, (iii) the pseudo observations are such that they exhibit a simple structure that is captured by the latent variable model. This idea is illustrated in Fig. 3-3 where we compare the observations $\mathbf{Y}$ and the pseudo-observations $\mathbf{S}$ which have a simple structure across the input dimensions (i.e. the columns of the matrices).



(a) Observed data $\mathbf{Y}$. (b) Aligned data $\mathbf{S}$ (pseudo-observations).

Figure 3-3: Observed and aligned data (pseudo-observations) in matrix form.

3.5 Alignment model

Combining the two components discussed in Sections 3.3 and 3.4, we formulate the joint objective that allows us to perform the alignment task as defined in Sec. 3.2. This involves simultaneously optimising the following two objectives:

1. For each of the J sequences we fit the GPs f_j to the observed data $\mathbf{y}_j$ and the pseudo-observations $\mathbf{s}_j$ (using $f_j(g_j(\mathbf{x}))$ and $f_j(\mathbf{x})$, respectively) by learning the hyperparameters of the GPs and the auxiliary variables $\mathbf{v}_j$ of the warpings. This includes the GP prior as defined in Eq. 3.8 and the likelihood of the observations under i.i.d. Gaussian noise with precision β_j, defined as

$$p(\mathbf{Y}|\mathbf{F}^{\mathrm{G}}) = \prod_j p(\mathbf{y}_j|\mathbf{f}_j^G) = \prod_j \mathcal{N}(\mathbf{y}_j|\mathbf{f}_j^G, \beta_j^{-1}\mathbb{I}). \tag{3.10}$$

   This leads to the corresponding marginal log-likelihood

$$\mathcal{L}_1 = \log p(\mathbf{S}, \mathbf{Y} \mid \mathbf{x}, \{\theta_j\}, \{\beta_j\}) \tag{3.11}$$

   which is available in closed form, as discussed in Sec. 2.1.1, since $\mathbf{Y}$ and $\mathbf{S}$ are the evaluations of the same latent functions at different locations.

2. We impose the alignment objective by learning a low-dimensional representation $\mathbf{Q}$ of the pseudo-observations $\mathbf{S}$. The joint distribution for the GP-LVM takes into account the GP prior over the mapping $\mathbf{H}$ with the corresponding hyperparame-

ters ψ_h, $p(\mathbf{H} \mid \mathbf{Q}, \psi_h)$, the Gaussian likelihood with precision γ factorised over the dimensions, $\prod_n p(\mathbf{S}_{:,n} \mid \mathbf{h}_n, \gamma)$, the prior over the latent variables $\mathbf{Q}$ with hyperparameters ψ_q, $p(\mathbf{Q} \mid \psi_q)$ and the hyperpriors over ψ_h and ψ_q. The corresponding log-likelihood is available in closed form (as discussed in Sec. 3.1.1):

$$\begin{aligned}\mathcal{L}_2 &= \log p(\mathbf{S} \mid \mathbf{Q}, \psi_h, \psi_z, \gamma) \\ &= \frac{N}{2} \log |\mathbf{K}_{qq}| - \frac{1}{2} \mathrm{Tr}(\mathbf{K}_{qq}^{-1} \mathbf{S}\mathbf{S}^T) \\ &+ \log(p(\mathbf{Q} \mid \psi_q)) + \log(p(\psi_h)) + \log(p(\psi_q)) + \text{const.}\end{aligned} \tag{3.12}$$

We place priors over the hyperparameters $\{\theta_j, \psi_h, \gamma, \beta_j\}$ as log-Normal distributions with zero mean and unit variance, and use a fully factorised prior over the latent variables $\mathbf{Q}$, $p(\mathbf{Q}) \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$. Furthermore, we place a zero-mean GP prior on the raw sample points $\mathbf{v}_j$ to encourage warps that are close to identity:

$$\log\, p(\{\mathbf{v}_j\}) = \sum_{j=1}^{J} \log\, \mathcal{N}(\mathbf{v}_j \mid \mathbf{0}, \mathbb{I}_N)\,. \tag{3.13}$$

We optimise $\mathcal{L}_1$ and $\mathcal{L}_2$ together w.r.t. the pseudo observations $\mathbf{S}$, the latent variables $\mathbf{Q}$ and the hyperparameters $\{\theta_j, \psi_h, \gamma, \beta_j\}$ to obtain the MAP estimates.

While we found the MAP training procedure to work well in practice in solving the alignment task, we recognise that there are multiple drawbacks to using MAP estimates more generally. Firstly, since the latent inputs are not marginalised out, the model may be sensitive to over-fitting. Secondly, the MAP objective does not provide any insight for selecting the optimal number of latent dimensions. We found that for all the applications that we considered, choosing the latent dimensionality by hand was not an issue (we typically set the latent dimensionality to 2 which provides good results and allows us to visualise the latent space easily). Alternatively, one might use cross-validation or a Bayesian GP-LVM [Damianou and Lawrence, 2013] (also called Variational GP-LVM) to estimate the number of effective dimensions for a particular task.

3.5.1 Implementation

We implement our model using the TensorFlow [Abadi et al., 2015] framework and minimise the negative log-likelihood objective using the Adam optimiser [Kingma and Ba, 2014]. By default, we used standard squared exponential covariance functions for all the Gaussian process priors. In some of the experiments, different covariance functions were

used when the data or warping functions were less smooth (*e.g.* the Matérn covariance). The complexity of our method is limited by the inversion of the covariance matrices and therefore scales with $\mathcal{O}(JN^3 + J^3)$. However, there are standard sparse approaches available to scale to longer sequences. We also implemented the sparse variational method of Titsias [Titsias, 2009] which reduces the complexity to $\mathcal{O}\big(J\,NM^2 + J^3\big)$, where M is a specified number of inducing points for the sparse approximation. This method performed well for M an order of magnitude smaller than the full N. We note that the use of a sparse approximation fits naturally with the rest of our model as it increases the smoothness of the observations, which may simplify the alignment task.

3.6 Comparison of variants of our model

In order to analyse the effect of the individual parts of the proposed approach, we consider two variants of our model, that we refer to *GP-LVM+basis* and *energy+GP*, which correspond to either replacing our model of the sequences and the warps with a parametric model introduced in Sec. 2.2.2, or replacing the alignment objective with the pairwise L^2 metric (also introduced in Sec. 2.2.2). We then compare and contrast our approach to some related previous work on warped GPs, scaled GP-LVMs and GP factor analysis.

Parametric warps Recall the parametric model of the warps defined in Sec. 2.2.2. We use a parametric re-sampling function that is a convex combination of K monotonically increasing basis functions. For each input sequence $\mathbf{y}_j$, we learn a set of weights $\mathbf{w}^{(j)} \in \mathbb{P}^K$, where the weights lie on the surface of the k^{th} order probability simplex $\mathbb{P}$; the resulting function is guaranteed to be monotonic. The task is now to find the set of weights $\{\mathbf{w}_k^j\}_{k=1}^K$ such that resampling the data according to the warping functions results in the aligned sequences. Here the pseudo-observations can be defined using the following mapping $\mathbf{s}_j = \mathbf{y}_j\left(\sum_i \mathbf{b}_i w_{j_i}\right)$ where we use linear or cubic interpolation of the values of the vector $\mathbf{y}_j$ to evaluate it at the inputs specified by the warpings $\sum_i \mathbf{b}_i w_{j_i}$. We use the same latent variable model as defined in Sec. 3.4 and refer to this model as *GP-LVM+basis*. The model can be learned using gradient descent where we optimise the alignment objective defined in Eq. 3.12 w.r.t. the latent variables and the parameters of the GP-LVM, and the weights $\mathbf{w}$. The parametric model described above, as well as some previous approaches, relies on hand-picked basis functions to define the warps. This results in poor accuracy when the set of basis functions is small and in high computational complexity when the set is large.

Energy alignment objective We demonstrate the efficacy of using the alignment GP-LVM to perform simultaneous clustering and alignment by replacing it with an energy minimisation objective that is similar to the previous literature, *e.g.* [Kurtek et al., 2011]. The latent variable model part of the objective is replaced with an energy minimisation term between each of the $\mathbf{s}_j$ pairwise:

$$E = \sum_{i=1}^{J} \sum_{j=i+1}^{J} ||\mathbf{s}_i - \mathbf{s}_j||^2. \tag{3.14}$$

In practice we can define a further latent variable $\tilde{S}$ and define the energy instead as

$$E = \sum_{i=1}^{J} ||\mathbf{s}_i - \tilde{\mathbf{s}}||^2 \tag{3.15}$$

where the variable $\tilde{\mathbf{s}}$ is directly optimised along with the parameters of the warpings (similar to the pseudo-observations in the previously defined models). Methods that rely on the standard L^2 metric in the input space are ill-posed and thus require a regularisation term. This leads to an optimisation problem that suffers from poor local minima and relies on the use of a coarse-to-fine approach where the optimisation is first performed on the data that has been substantially smoothed, gradually adding high frequency detail; in the final optimisation call the original (non-smoothed) data is used. In Sec. 3.7 we show the results of this method with the GP warping functions (*energy+GP*) and, for completeness, we also consider the energy alignment objective in conjunction with the basis function warpings as described above (*energy+basis*).

3.6.1 Further discussion of previous work

Warped GPs In [Snelson et al., 2004] and [Lázaro-Gredilla, 2012] (which in effect is a two-layer DGP [Damianou and Lawrence, 2013]) the authors construct a GP with a warped input space to account for differences in observations (e.g. inputs may vary over many orders of magnitude), and show that a warped GP finds the standard preprocessing transforms, such as the logarithm, automatically. In the case of warped GPs, the observations $\mathbf{y}_j$ are transformed using a monotonic neural-net style sum of tanh functions to produce the warped observations:

$$\mathbf{s}_j = f(\mathbf{y}_j; \Psi = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}) = \mathbf{y}_j + \sum_{i}^{I} a_i \tanh(b_i(\mathbf{y}_j + c_i)), \quad a_i, b_i \geq 0 \quad \forall i. \tag{3.16}$$

which transforms the observations using a series of smooth steps (with $\mathbf{a}$ controlling the size of the steps, $\mathbf{b}$ controlling their steepness, and $\mathbf{c}$ their position) while retaining a linear trend away from the data. Note that the resulting distribution in the observation spaces is no longer Gaussian but can be asymmetrical and multi-modal depending on the transformation function. In comparison, our approach leads to warped inputs $\{g_j(\mathbf{x})\}$; the resulting distribution is Gaussian.

Scaled GP-LVM Another example of a GP-based model that alters the observations is the Scaled GP-LVM [Grochow et al., 2004, Wang et al., 2006], where a $N \times N$ matrix $\mathbf{W}$ is used to rescale the observed data $\mathbf{y}_j$ as $\mathbf{s}_j = \mathbf{W}\mathbf{y}_j$ (which corresponds to using a warped GP discussed above with rescaling as the warping function). This is appropriate when the dimensions of the data in $\mathbf{y}_j$ have different intrinsic scales. In that case $\mathbf{W}$ is a diagonal matrix, hence the rescaling may equivalently be expressed as a GP with kernel function $k(\mathbf{x}, \mathbf{x}')/w_m^2$ for dimension m, and the data fitting term in the log-likelihood may be written as:

$$L_{\text{data fitting}} = \frac{1}{2}\sum_{n=1}^{N} w_n^2 \mathbf{y}_{:,n}^T \mathbf{K}^{-1} \mathbf{y}_{:,n}, \tag{3.17}$$

where $\mathbf{y}_{:,n}$ corresponds to the n^{th} dimension of all of the J inputs in the observation matrix $\mathbf{Y}$. Meanwhile in our model we aim to warp the observations to produce data $\mathbf{S}$ such that every dimension of this matrix contains observations that are as simple as possible in terms of the corresponding low-dimensional structure.

GP factor analysis [Duncker and Sahani, 2018] use GPs for modelling sequences of neural population spike-trains and the corresponding temporal warps. The proposed approach is an extension to GP factor analysis [Yu et al., 2009] and uses a linear combination of shared and private latent processes to encourage alignment of sequences for different trails. Unlike our work, [Duncker and Sahani, 2018] do not recover a clustering of the sequences and thus require the groups of sequences for alignment to be known a-priori.

3.7 Experiments

We now discuss the experimental evaluation of our proposed model. We use standard squared exponential covariance functions for all the GP priors, unless stated otherwise.

We show comparisons to current state-of-the-art approaches from data mining and functional data analysis communities using publicly available reference implementations[1]. The accuracy is primarily measured in terms of the warping error, *i.e.* the mean squared error (MSE) between the known true warps and the estimated warps. We also report the alignment error, the MSE between the pairs of aligned sequences; however, we note that the alignment error is easily misinterpreted since it is a local measurement. In particular, it does not capture the degenerate cases where the local maxima and minima in the input sequences are shifted to non-corresponding extrema; this is particularly true in data sets with periodic components. Other examples of degenerate behaviour are multiple dimensions collapsing to a single point and warps that rely on translating and rescaling every input in each dimension that leads to over-fitting (an example of this is IMW alignment [Zhou, 2012]). All of these cases may result in high alignment accuracy but produce poor quality aligned results.

3.7.1 Data sets with quantifiable comparisons

For this experiment, we use the data set proposed by Zhou and De la Torre [Zhou, 2012]. It consists of sequences that are generated by temporally transforming latent 2D shapes under known warping transformations that allow quantitative evaluation of the estimated warps. To better assess the quality of the alignments, we run 25 tests with a randomly selected data set size, dimensionality and temporal transformations. Our approach outperforms other methods on these data sets and produces accurate alignments irrespective of the size of the data set, dimensionality and structure of the sequences. The quantitative results are provided in Fig. 3-4 and Table 3.1. These results show that our method outperforms the baselines and the variations of our method on this task. The corresponding qualitative results are given in Fig. 3-5 and Fig. 3-6 where we show examples of the alignments and the warps.

The variant of our method that uses parametric warps (*gplvm+basis*) performs competitively on these data sets, motivating the use of a GP-LVM objective for alignments. The competitive performance of *gplvm+basis* may be explained by the fact that, unlike the methods that rely on energy minimization, *gplvm+basis* benefits from the regularization of the low dimensional representation and thus does not require any additional regularisation terms to be introduced to handle the fact that the problem is ill-posed. Furthermore, we see that our nonparametric approach to modelling the time warps improves the flexibility of the model; out of the two models that rely on energy minimisation as the

[1] See [Zhou and de la Torre, 2018] for the implementation of DTW, DDTW, IMW, CTW, GTW, and [Srivastava et al., 2018] for the implementation of SRVF.

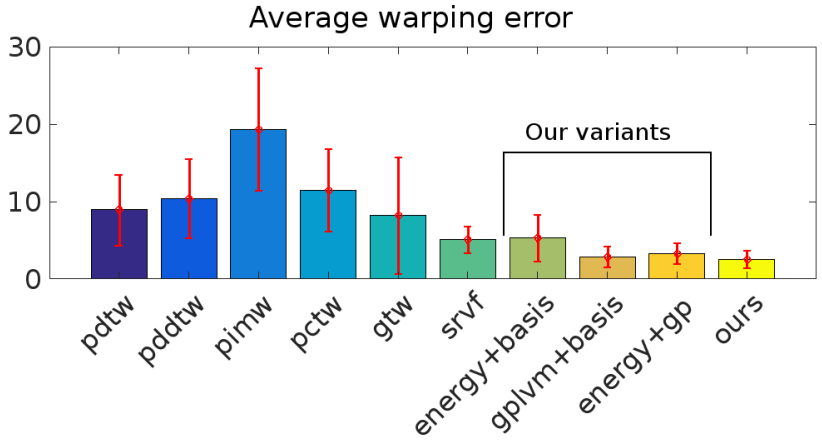


Figure 3-4: Comparison to baselines: average error on 25 data sets proposed by Zhou *et al.* [Zhou, 2012].

Dataset no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Mean
J	13	10	10	7	13	6	6	12	3	10	13	8	6	10	7	5	5	14	8	3	8	11	7	6	9	
T	258	157	107	246	169	131	92	144	138	298	146	240	204	213	157	230	247	196	248	277	141	153	83	285	178	
PDTW	9.32	14.38	5.88	12.42	10.57	10.03	2.30	6.60	2.41	16.99	9.84	13.47	4.95	16.72	4.00	6.97	14.03	6.57	15.01	1.98	4.44	5.16	5.05	13.19	10.85	8.93
PDDTW	10.55	14.67	6.80	13.86	12.18	10.30	2.58	7.35	3.95	24.59	10.68	15.45	6.57	18.60	5.54	8.35	14.91	7.70	16.38	5.37	6.11	7.43	5.44	14.02	11.17	10.42
PIMW	26.61	19.27	13.36	24.59	21.16	14.75	6.48	22.14	5.36	38.54	18.06	26.72	16.77	28.39	9.54	22.31	21.23	22.35	27.54	11.13	12.05	18.45	9.06	30.40	16.54	19.31
PCTW	12.12	15.30	9.50	18.89	15.45	11.32	2.77	10.58	2.66	17.26	9.98	24.72	8.04	17.19	6.23	10.21	16.03	10.73	15.17	7.12	5.88	5.59	6.00	16.92	10.98	11.47
GTW	6.52	9.54	6.54	6.96	7.50	3.23	4.27	10.63	0.86	33.31	2.55	21.28	5.35	4.72	3.38	23.28	8.50	10.70	2.83	3.93	2.74	2.56	5.45	15.70	2.63	8.20
SRVF	6.74	3.41	4.73	8.06	6.94	4.14	2.14	3.34	3.10	8.65	5.13	5.33	3.57	7.37	5.78	7.02	4.87	4.71	5.54	5.12	3.82	4.55	2.22	4.53	7.04	5.11
energy+basis	9.91	5.08	4.69	6.12	6.89	3.06	2.10	5.33	0.98	14.45	6.64	6.57	2.10	10.38	2.74	3.83	5.13	7.94	5.67	1.37	4.42	5.79	2.12	5.00	4.88	5.33
gplvm+basis	5.85	3.09	2.98	5.29	3.65	2.24	1.31	3.53	0.87	**1.85**	3.67	**3.68**	**1.49**	5.58	1.58	3.55	3.69	**2.13**	3.12	**1.22**	3.20	3.59	1.35	**1.72**	**1.90**	2.88
energy+gplvm	6.80	**2.45**	3.29	6.35	5.31	2.58	1.39	**3.23**	0.97	4.54	2.97	4.34	2.79	3.37	3.16	3.22	4.12	3.48	2.64	2.07	3.94	2.69	2.20	2.18	2.65	3.31
ours	**4.39**	4.55	**1.79**	**1.93**	**2.34**	**1.91**	**1.23**	4.68	**0.84**	3.49	**2.11**	4.94	3.47	**3.14**	**1.40**	**3.03**	**2.01**	2.57	**2.10**	1.48	**2.20**	**1.93**	**1.31**	2.87	2.11	**2.55**

Table 3.1: Data sets used for our evaluation, where J and T refer to the number of sequences and dimensionality. Results are presented as MSE of warpings. The summary of the results presented in this table is given in Fig. 3-4.

alignment objective, *energy+basis* and *energy+gp*, the latter one demonstrates lower warping error and significantly lower standard deviation on this data set. This result supports the premise that even though the nonparametric representation allows for any smooth monotonic warp, the probabilistic framework places sufficient structure to make the problem better regularised and to help prevent over-fitting. Due to the smoothness of the warps in both the parametrised and the nonparametric cases, we are able to avoid the degenerate cases where many consecutive elements of one sequence are aligned to a single element in the other sequences.

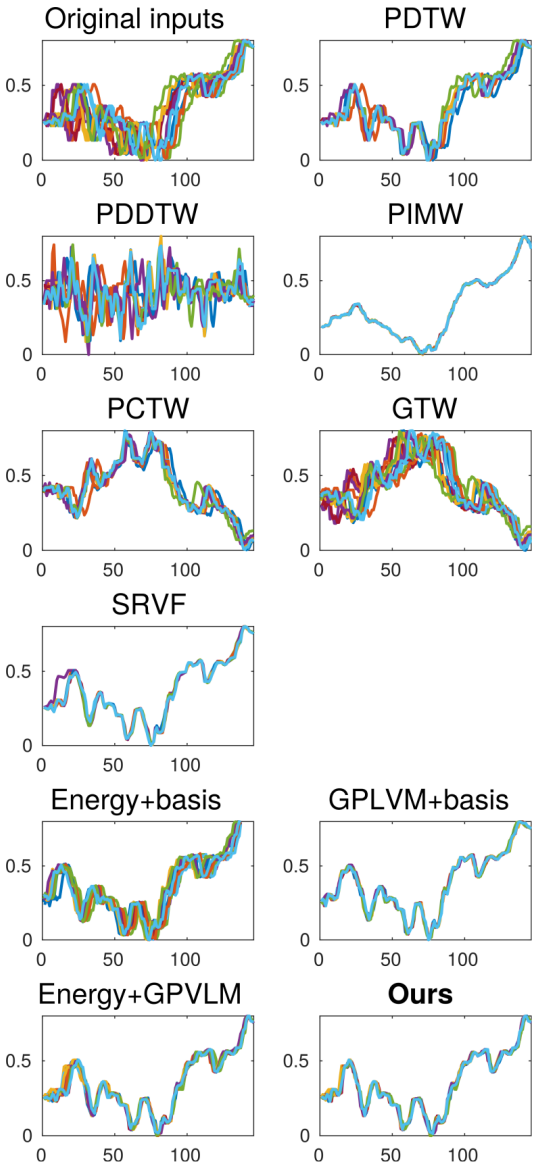


Figure 3-5: Original inputs and aligned sequences estimated by DTW, DDTW, IMW, CTW, GTW, SRVF, our approach and its three variants.

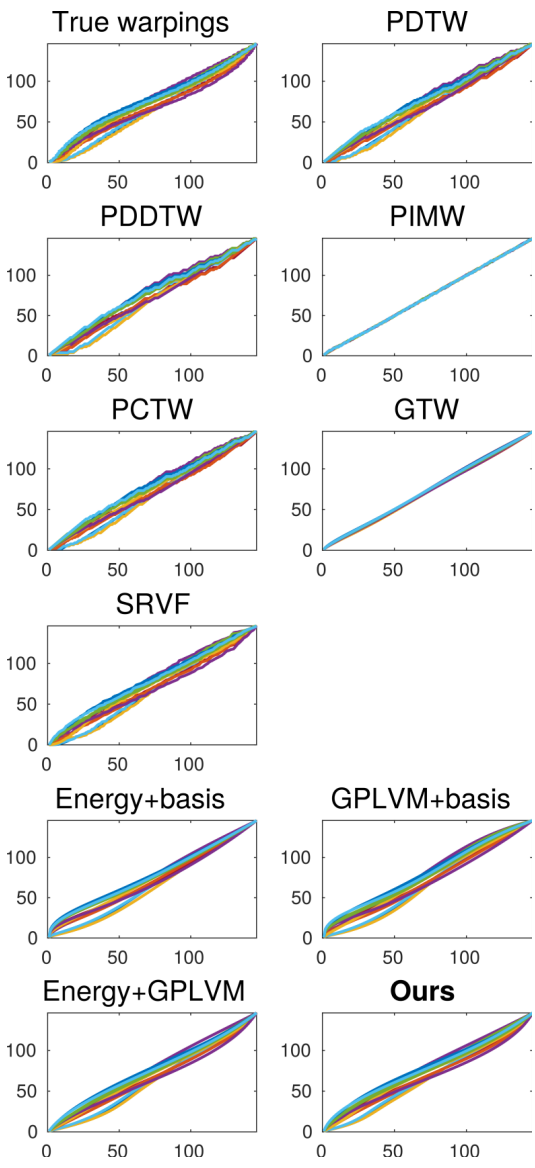


Figure 3-6: True warps and warps estimated by DTW, DDTW, IMW, CTW, GTW, SRVF, our approach and its three variants.

3.7.2 Data set for clustering

In our second experiment, we consider a data set that contains multiple clusters of sequences. This task requires the sequences to be aligned within each cluster. None of PDTW, PCTW, GTW nor the energy minimisation methods are able to perform this task as they have no knowledge of the underlying structure of the data set. The

MSE (SD)	SRVF	GP-LVM+BASIS	**OURS**
ALIGNMENT	6.4 (±1.7)	8.4 (±2.7)	**5.9** (±1.1)
WARPING	30.0 (±10.4)	**9.7** (±4.9)	**9.7** (±5.7)

Table 3.2: Quantitative comparison of alignments and warps for the best competing method on a data set with multiple true sequences (alignment and grouping task).

SRVF algorithm performs clustering by first aligning the data in terms of amplitude and phase, then performing fPCA based on the estimated summary statistics, and finally modelling the original data using joint Gaussian or nonparametric models on the fPCA representations. We compare the performance of the SRVF algorithm with our approach as well as the variant of our approach with fixed basis functions.

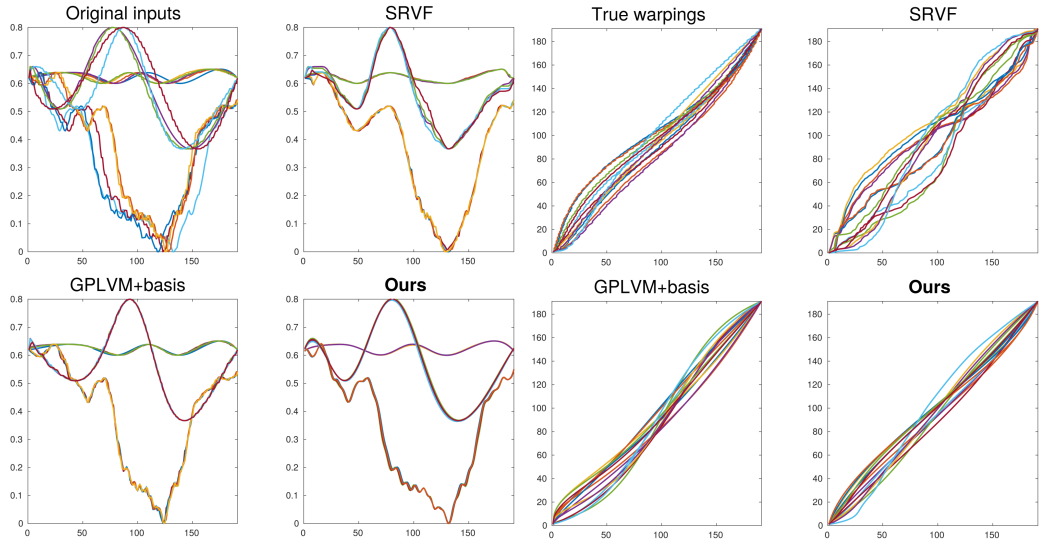


Figure 3-7: Comparison of alignment of motion capture sequences using SRVF, GP-LVM+basis and the proposed GP-LVM alignment objective. Alignment of 15 sequences that belong to 3 different clusters (left), and the corresponding warping functions (right).

We consider a data set that contains three distinct groups of functions that were generated by temporally transforming three random 2D curves. All three approaches rely on the structure of the data alone to recognise the existence of the clusters and Fig. 3-7 shows that all three methods are able to align the data within clusters.

The performance of the methods is contrasted by calculating the MSE among all pairs of sequences within each group (alignment error) and the MSE between the true warping functions and the warping functions calculated using each of the methods (warping error). For this comparison we repeat the test 25 times with randomly selected initial curves,

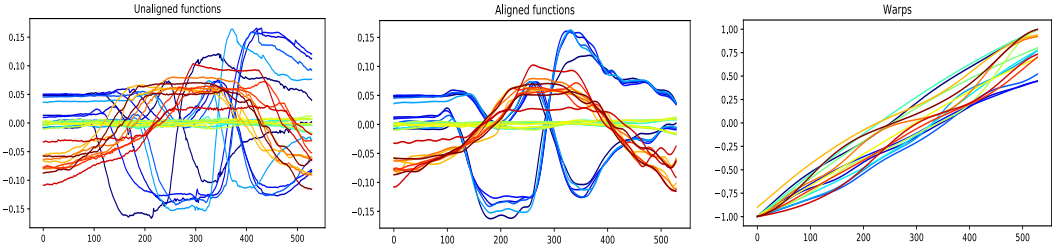
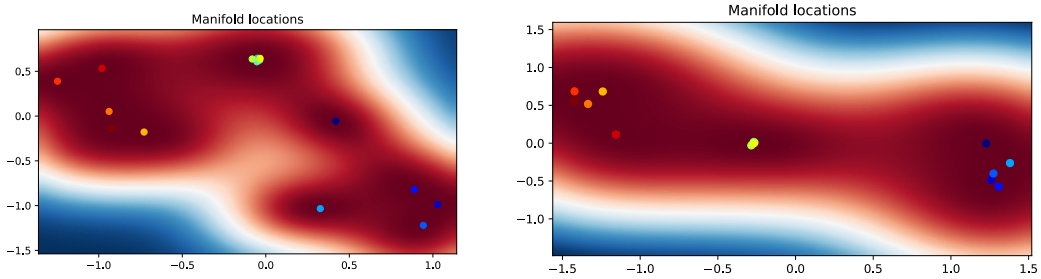


Figure 3-8: GP-LVM alignment demonstrates the preference for a simplified explanation when the model is given the ability to align the data.



(a) Without alignment.

(b) With alignment.

Figure 3-9: 2D manifolds produced without and with alignment in the GP-LVM. Using the alignments emphasizes the existence of multiple clusters of data and aligns data points within each cluster.

number of dimensions and number of sequences per group. The quantitative comparison in Table 3.2 shows that our method consistently achieves the lowest alignment errors (with lowest standard deviation (SD) on the set of data sets).

Our method, as well as the parametric variant of it, also achieves low warping errors in comparison to SRVF. This implies that we are able to reconstruct the original temporal transformations, which are smooth and close to an identity, more accurately than SRVF. This behaviour is apparent in Fig. 3-7 where the warping functions produced by our method, and the parametric version of it, resemble the true warps while SRVF estimates noticeably different warping functions. Consequently, the results for SRVF exhibit unpredictable distortions in the aligned sequences (when contrasted with the original observations of the corresponding sequences) such as the appearance of steps in the aligned sequences that are not present in the original data (see Fig. 3-7). This behaviour suggests that the algorithm has very limited knowledge of the underlying structures present in the data set. Furthermore, both the energy minimisation and the SRVF approaches require the use of the coarse-to-fine estimation in optimisation. These results further reflect the differences between the SRVF method and our approach; while SRVF is cast as an optimisation problem over a constrained domain, the domain imposed by

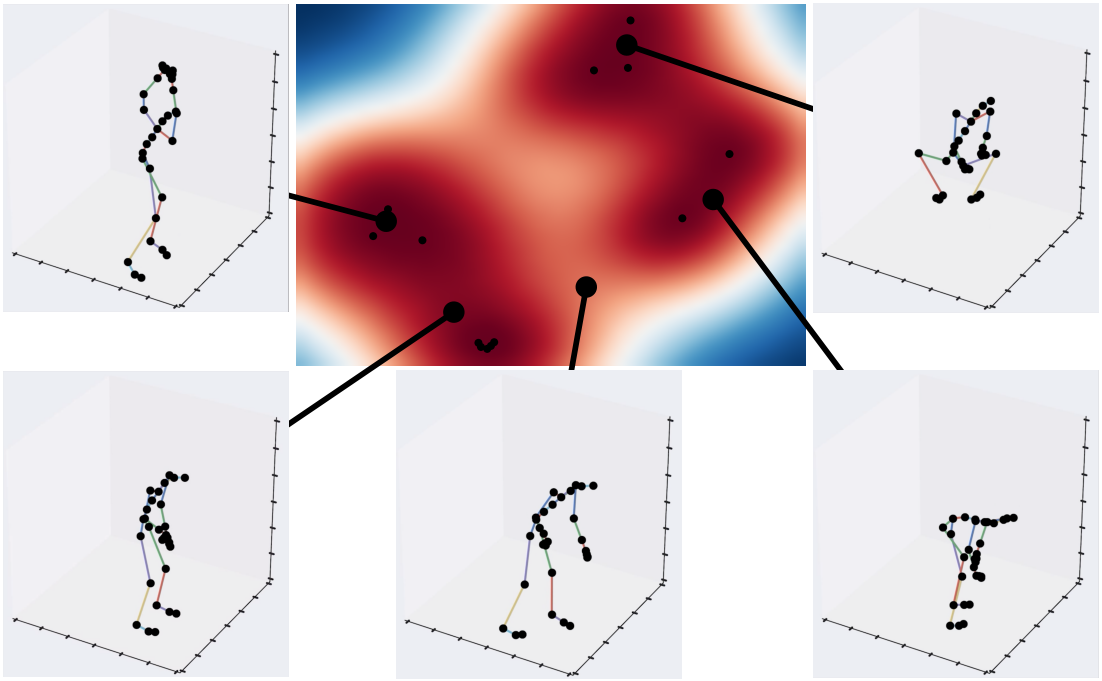


Figure 3-10: An advantage of our approach is that it not only aligns the data but is also a generative model. Here we show novel sequences generated at new locations in the manifold. The black dots indicate the embedded locations of the training sequences of golf motions of swing, putt and placing of a ball. We note that, while we have only shown still images, each manifold location describes an entire time series. The red areas in the manifold correspond to areas of high confidence (the predictions for the corresponding latent values are similar to the observed sequences) while the blue areas correspond to low confidence (the predictions for the corresponding latent values are close to the mean function). Such latent representation also allows us to produce novel clips of the various motions that interpolate between the observed clips.

our formulation is much larger and structured from the assumptions encoded in the prior. This provides a better regularisation ultimately leading to the improvement in the recovered warpings.

3.7.3 Motion capture data

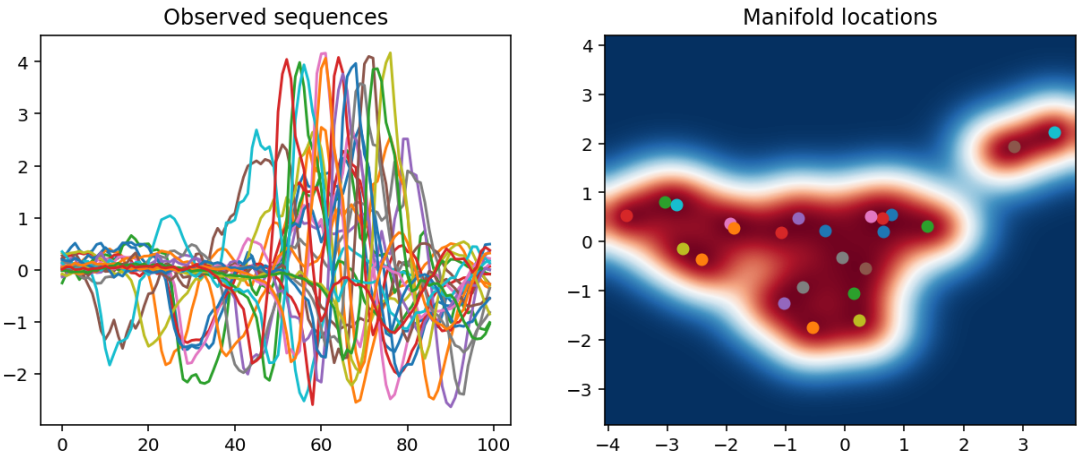
The performance of our model is evaluated on a set of motion capture sequences from the CMU database [Lab, 2016] that contains (noisy) time-series data of various types of human motion. The use of this data set is motivated by the industrial application introduced in Ch. 1. We use the motions of subject no. 64 from the CMU data set that correspond to golf related motions such as a swing, a putt, and placing of a ball. Each

input sequence corresponds to a short clip of motion and the data is represented as unit quaternion locations of the joints of the subject performing the motion. We consider five instances of the three different motions that need to be temporally aligned within the three groups. Fig. 3-8 illustrates how our model favours the simplified, *i.e.* aligned, inputs. The corresponding manifolds produced using a traditional GP-LVM (*i.e.* without alignment) and a manifold produced using our approach are shown in Fig. 3-9. Our model produces a fine alignment of the input sequences within each of the groups, and consequently the resulting two-dimensional manifold offers a good separation of the three groups.

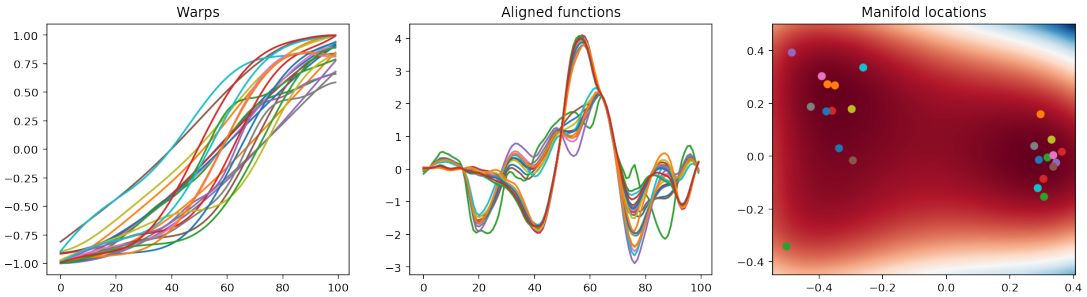
In Fig. 3-10 we provide an illustration of how the GP-LVM allows us to generate novel sequences of motions that are temporally consistent within the distinct groups of motions. New locations in the manifold encode novel motion sequences that are supported by the data, and, by allowing the model to align the data, it greatly improves the generative power as the model is capable of producing a wider range of plausible motions. Our model with implicit alignment is able to generate smoother transitions in the manifold, producing high quality predicted sequences of data.

3.7.4 Heartbeats data

We next consider a data set which contains sequences of heartbeat sounds recorded by a digital stethoscope [Bentley et al., 2011]. It is known that a normal heart sound has a clear “lub dub, lub dub" pattern which varies temporally depending on the age, health, and state of the subject [Bentley et al., 2011]. It is required to analyse this pattern for different patients to classify the recording of normal beats versus those that contain murmur or extrasystoles. Instead of using a pre-processing step with a low-pass filter to account for the noise in the high frequencies, we use a Matérn 3/2 kernel for the GPs that model the observed data; this allows us to take into count the rapid variations in the recordings while also limiting the effect of the uninformative high frequency noise. Our approach automatically aligns and clusters the recordings of the heart sounds into two groups corresponding to two types of wave forms. Fig. 3-11 illustrates how simultaneous fitting and alignment allows us to correctly discover and cluster the two types of heartbeats.



(a) The clustering of the unaligned observed sequences does not reveal the two types of heartbeats.



(b) Accounting for the alignment of sequences allows us to discover automatically the two different types of heartbeats.

Figure 3-11: Alignment of heartbeats data [Bentley et al., 2011].

3.7.5 iPhone motion data

This data set contains aerobic actions recorded using the Inertial Measurement Unit (IMU) on a smartphone [McCall et al., 2012]. The IMU includes a 3D accelerometer, a gyroscope, and a magnetometer and records samples at 60 Hz, consequently, the data contains high frequency variations. As in [Tucker et al., 2013], for our experiment we take the accelerometer data in the x-direction for the jumping actions from subject 3, and, in particular, we look at 5 sequences each of which contains 400 frames. Unlike previous methods, which require the data to be smoothed first [Tucker et al., 2013]; our framework allows us to take into account the prior belief about the data set in a principled way. By replacing the smooth RBF kernels for modelling the data with a Matérn 1/2 kernel and taking into account the periodic nature of the actions by also including an additive periodic kernel, we are able to model the data without the need for preprocessing. Furthermore, by removing the smoothing prior from the warping

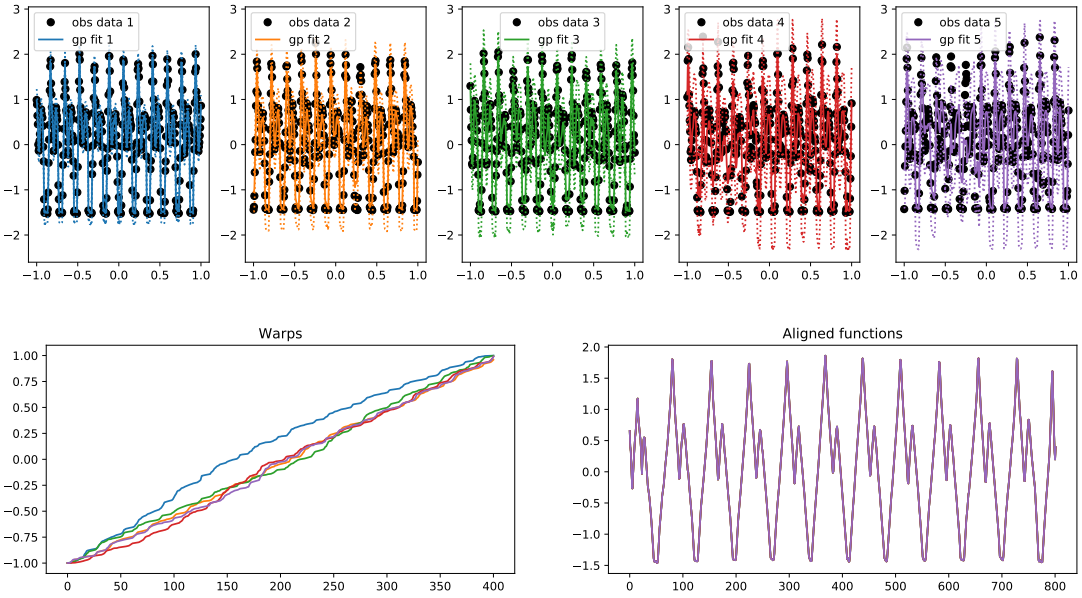


Figure 3-12: The top row shows the observed data that contains high frequency oscillations, and the GPs fitted to this data. The bottom row shows the corresponding warps (left) and the aligned functions. In this example, no smoothness prior is placed on the warping function resulting in warps that are piecewise linear.

functions, we allow the warps to be more flexible improving the alignment accuracy. The qualitative alignment results for the iPhone motion data are shown in Fig. 3-12.

3.7.6 Shift task

A common task in functional data alignment is that of estimating uniform translations of the time axis. One particular problem described by Marron *et al.* is that of aligning nuclear magnetic resonance (NMR) spectrum corresponding to different chemical components (e.g. ethanol) for a set of wines [Marron et al., 2015]. It is known that pH differences in wines induce a shift in values of the components and impedes their identification [Larsen et al., 2006]. As shown in [Marron et al., 2015], the alignment may be achieved using uniform shifts and minimizing the loss that requires sequences to be proportional to each other. Our approach allows us to perform the task of NMR spectrum alignment, and we are able to demonstrate a separation in the phase between the red wines and the white and rosé, see Fig. 3-13.

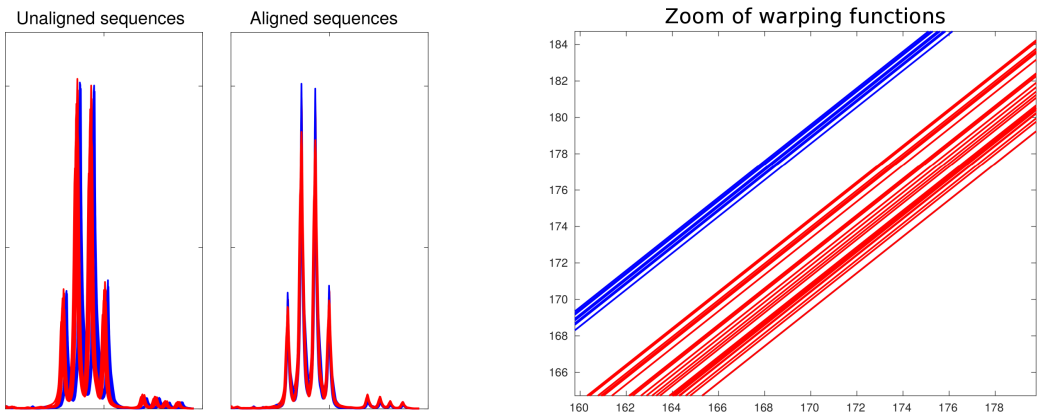


Figure 3-13: Alignment of NMR spectrum data [Marron et al., 2015]. The zoom of the warping functions show the separation of the white/rosé wines (shown in blue) and red wines (shown in red).

3.8 Discussion

We have presented a model that is able to implicitly align inputs that contain temporal variations. Our approach models the observed data directly producing a generative model of the functions rather than interpolating between observations. In addition, using a GP-LVM for alignment builds a generative model that has the benefit of simultaneous clustering and alignment of the input sequences. Furthermore, we proposed a continuous, nonparametric explicit model of the time warping functions that removes issues such as quantisation artefacts and the need for ad-hoc pre-processing. We demonstrated that the proposed approaches perform competitively on alignment tasks, and outperform the existing methods on the task of simultaneous alignment and clustering.

When designing the model described in this section, we make a number of modelling and implementation choices that influence the behaviour of the model and allow us to achieve the results reported earlier. In the following section we discuss our findings related to the concept of pseudo-observations.

3.8.1 Implementation of pseudo-observations

Recall that in the proposed model the aligned sequences get optimised directly and we refer to these sequences as pseudo-observations, denoted by $\mathbf{S}$. The idea of noisy pseudo-observations that are treated identically to the real observations has been discussed previously by [Louizos and Welling, 2016] in the context of efficient posterior sampling.

The authors assume noisy pseudo-observations with i.i.d. Gaussian noise and explain that this helps avoid numerical instabilities during optimization.

In our implementation, the pseudo-observations are defined as a Tensorflow variable $\mathbf{S}$ that corresponds to the evenly sampled values of the latent functions at locations $\mathbf{x}$ (throughout this section we ignore the index j for the sequences, and assume that the data has been concatenated for all J):

$$\mathbf{S} = [f_1(\mathbf{x}), \ldots, f_J(\mathbf{x})]^T. \tag{3.18}$$

The variable $\mathbf{S}$ has shape $J \times N$ with each row corresponding to N sampled values of the latent function f_j at $\mathbf{x}$. Note that the "=" sign in Eq. 3.18 should be understood as a constraint rather than as an equality sign; $\mathbf{S}$ is a directly optimised variable, and we wish to find the value of $\mathbf{S}$ as well as the latent functions $f_1, \ldots, f_J$ such that Eq. 3.18 holds.

To find the value of $\mathbf{S}$, we optimise the alignment objective discussed in Sec. 3.4. It consists of the data fitting term in Eq. 3.11, which imposes the constraint in Eq. 3.18, and the alignment objective in Eq. 3.12, which imposes the constraint that the latent functions should be similar to each other (*i.e.* the rows of $\mathbf{S}$ should be similar to each other).

An alternative approach entails understanding $\mathbf{S}$ in Eq. 3.18 as the posterior predictive mean of $\{f_j\}_{j=1}^J$ at $\mathbf{x}$, in which case we do not directly optimise $\mathbf{S}$. We fit the functions $\{f_j\}$ to the observed data $\mathbf{Y}$ at the warped sampling locations $g(\mathbf{x})$ using the Eq. 3.10, and use the predictive posterior means in Eq. 3.18 as the aligned sequences in the GP-LVM alignment objective in Eq. 3.12. This model will be discussed in Sec. 5.1.2. In this case there are fewer variables that need to be optimised (as $\mathbf{S}$ is no longer a variable itself) but changing the values of $\mathbf{S}$ becomes much harder as it can only be done indirectly by changing the fitted functions $\{f_j\}$ (which depend on the kernel parameters of the corresponding GP and the warped input locations $g(\mathbf{x})$).

4

GP-DPMM alignment models

In this section we build on the approach presented in Sec. 3 where the models of the individual sequences and the alignment across sequences are cast within a single framework. Similarly to the model proposed in Sec. 3, we use GPs to model the data and the warping functions, which allows us to reject the observation noise in a principled manner and imposes a smoothness constraint on the warping functions. In contrast, we consider mixture models (MMs) as models for alignment. Using a MM as the alignment objective results in an alignment model which performs clustering explicitly as opposed to the implicit clustering offered by the GP-LVM. Furthermore, casting the problem in a Bayesian setting allows us to take into account our uncertainty about the distribution of the data within the clusters, and, more importantly, the uncertainty about the effective number of clusters in the data, which in our case corresponds to the number of underlying functions explaining the observed sequences. Consequently, we explore the use of a nonparametric Bayesian MM based on Dirichlet processes in order to automatically find the optimal number of clusters.

We start this chapter by introducing the mixture models generally, and more specifically, the Bayesian MM and the Dirichlet process MM. We then describe how these can be utilised as alignment objectives in our model. We then discuss the practicalities and the implications of designing an alignment objective based on MMs in Sec. 4.3. Finally, we compare the performance of such an alignment model on synthetic data as well as real-world tasks in Sec 4.5.

4.1 Background

In this chapter we make extensive use of the ideas and specific constructions of mixture models, a general technique that has been used extensively in a pattern recognition setting for at least a half a century, in both its finite [Duda and Hart, 1973] and infinite [Antoniak, 1969, Ferguson, 1973] forms.

4.1.1 Mixture models

As explained in [Murphy, 2012], a mixture model is a case of a discrete latent variable model where the latent variables are integer labels, $c \in \{1, \ldots, K\}$, such that each observation $\mathbf{x}$ is associated with one and only one latent variable (*i.e.*, each observation $\mathbf{x}$ is assigned to only one of the K available clusters as indicated by c). Alternatively, $\mathbf{c} = \{c_1, \ldots, c_K\}$ is a vector such that $c_k \in \{0, 1\}$ and $\sum_k c_k = 1$, hence $\mathbf{c}$ is a vector of zeros with one non-zero value equal to 1 at location k in the vector (*i.e.* $\mathbf{c}$ is one-hot). The latent variables $\mathbf{c}$ are specified using a set of weights $\{\pi_k\}$ associated to each of the K components such that $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0$. This specifies the proportion of the data in each of the clusters, and $p(c_k = 1) = \pi_k$. Alternatively, the (marginal) distribution of the latent cluster assignments $\mathbf{c}$ can be written as:

$$p(\mathbf{c}) = \prod_{k=1}^{K} {\pi_k}^{c_k}, \tag{4.1}$$

which is a multinomial distribution where, due to the constraints on $\mathbf{c}$, the normalisation constant is equal to 1.

The conditional distribution of the observation $\mathbf{x}$ given the cluster assignment $\mathbf{c}$, $p(\mathbf{x} \mid c_k = 1)$, corresponds to the distribution of the data in the k^{th} cluster and is referred to as the k^{th} (mixture) component distribution[1]. The component distribution may take a variety of different forms, and it typically depends on a set of parameters $\{\boldsymbol{\theta}_k\}$. For our applications we will always assume that the component distributions are multivariate Gaussians where $\boldsymbol{\theta}_k$ corresponds to the mean $\boldsymbol{\mu}_k$ vector and the precision $\boldsymbol{\Lambda}_k$ matrix. Then, given a multivariate normal component distribution, the conditional

[1] Some authors, *e.g.* [Murphy, 2012], refer to this distribution as the base distribution, however, in the Bayesian setting it is common to refer to the priors on the mixture component distributions as the base distributions [Blei and Jordan, 2005]; we shall use the latter naming convention.

distribution of the observations given the cluster assignments is:

$$p(\mathbf{x} \mid \mathbf{c}) = \prod_{k=1}^{K} \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{c_k} . \tag{4.2}$$

The joint distribution of the observations $\mathbf{x}$ and the latent variable $\mathbf{c}$ follows from the graphical model in Fig. 4-1a, $p(\mathbf{x}, \mathbf{c}) = p(\mathbf{c})p(\mathbf{x} \mid \mathbf{c})$. Summing over the latent variables gives the marginal distribution of the observations:

$$p(\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{c})p(\mathbf{x} \mid \mathbf{c}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right) \tag{4.3}$$

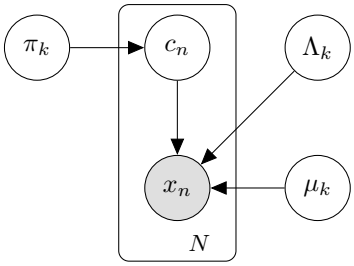
which is a convex combination of the multivariate Gaussian component distributions due to the constraints on the mixing proportions $\{\pi_k\}$, and so it is itself a probability distribution, and can be used as such. For example, a Gaussian mixture model can be used to model continuous data that is distributed as a sum of Gaussian bell curves. In a Bayesian setting, prior distributions are placed on the weights π_k and the parameters of the component distributions, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$. A popular choice is to use a Dirichlet prior on the weights, and conjugate priors for the component distributions (for example, given that the component distribution is a multivariate Gaussian with unknown mean and precision, one might use a normal-Wishart prior). A graphical model for the Bayesian Gaussian mixture model with conjugate priors is shown in Fig. 4-1b.

Clustering The construction of mixture models outlined above justifies their use for clustering. Clustering is a task of partitioning of data into groups (also called clusters) in a way that assigns similar data points to the same cluster and dissimilar data points to different clusters using some measure of similarity [Shalev-Shwartz and Ben-David, 2014]. This definition motivates the use of mixture models for clustering: a mixture model as outlined above partitions the data using the latent variable $\mathbf{c}$ in a way that the data within each cluster follows the corresponding component distribution.

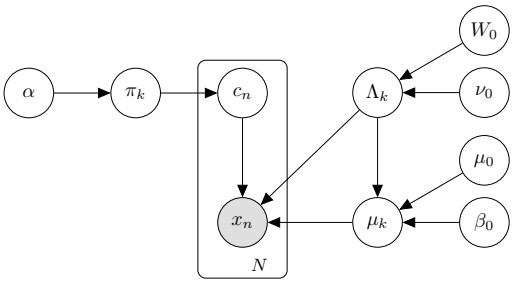
Once the mixture model is fitted, then the posterior probability that observation $\mathbf{x}$ belongs to cluster k follows from Bayes' rule:

$$p\left(c_k = 1 \mid \mathbf{x}, \boldsymbol{\theta}\right) = \frac{p\left(c_k = 1 \mid \boldsymbol{\theta}\right) p\left(\mathbf{x} \mid c_k = 1, \boldsymbol{\theta}\right)}{\sum_{k'=1}^{K} p\left(c_{k'} = 1 \mid \boldsymbol{\theta}\right) p\left(\mathbf{x} \mid c_{k'} = 1, \boldsymbol{\theta}\right)}. \tag{4.4}$$

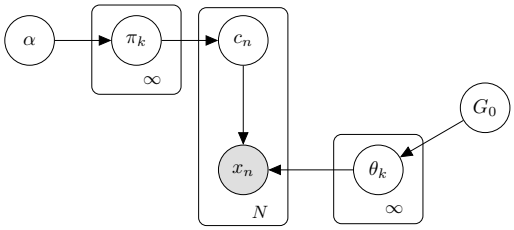
The argument that maximises this posterior distribution with respect to the cluster assignments gives a point estimate of the cluster assignment for a given data point $\mathbf{x}$.



(a) Graphical model of a Gaussian mixture model.



(b) Graphical model of a Bayesian Gaussian mixture model with normal-Wishart prior on the means $\boldsymbol{\mu}_k$ and the precision $\boldsymbol{\Lambda}_k$ and a Dirichlet prior on π_k.



(c) Graphical model of a Dirichlet process Gaussian mixture model with G_0 (e.g. normal-Wishart) prior on the parameters $\boldsymbol{\theta}_k$ (e.g. mean $\boldsymbol{\mu}_k$ and precision $\boldsymbol{\Lambda}_k$), and a Dirichlet process prior on π_k.

Figure 4-1: Graphical models of Gaussian mixture models.

4.1.2 Inference in mixture models

Learning in mixture models involves estimating the weights $\{\pi_k\}$ (also called the mixing coefficients) and the parameters of the component distributions (or the parameters of the corresponding priors). What makes learning these parameters hard is that the cluster assignments are unknown and so they need to be estimated from the data just like the parameters of the component distributions. A mixture model with K components has $K!$ equivalent ways of assigning the data to clusters which is often referred to as the problem of identifiability [Bishop, 2006]. In practice, this means that the marginal log likelihood $p(\mathbf{x})$ includes a summation in the logarithm, and while it is possible to find the derivatives of the conditional log likelihood with respect to the parameters in closed form, the interplay between the cluster assignments and the fitting of the base distributions interferes with finding closed form solutions for the parameters.

Expectation - Maximisation The same interplay motives the use of an Expectation - Maximisation (EM) algorithm [Dempster et al., 1977] which utilises the closed form ex-

pressions for the distributions of the mixing coefficients and the component distributions. The EM algorithm consists of iterating over two steps: the first involves computing the expectations of the latent variables using $p(c_k = 1 \mid \mathbf{x})$ while the second step computes the maximum likelihood parameters given the expected values of the latent variables from the first step for each cluster component k.

Variational inference An alternative approach to learning the parameters of a mixture model involves using variational inference. As discussed in Sec. 2.1.2, variational inference is a technique that allows us to approximate intractable integrals and, in the context of Bayesian modelling, it is used for two main purposes: the approximation of the posterior distribution of the latent variables and the estimation of the lower bound on the model evidence. We refer the reader to, for example, [Bishop, 2006] for a detailed derivation of the variational approximation for a Bayesian Gaussian mixture model (with conjugate priors), and here we only comment on the form of the variational distributions and the resulting lower bound on the evidence. Let us assume that the variational distribution, which is used to approximate the posterior probability of the latent variables $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$ and the model parameters $\{\pi, \boldsymbol{\theta}\}$ (suppressing the cluster index k), factorises as follows:

$$q(\mathbf{C}, \pi, \boldsymbol{\theta}) = q(\mathbf{C})q(\pi, \boldsymbol{\theta}) \tag{4.5}$$

where $\boldsymbol{\theta}$ denotes the parameters of the component distribution. This factorisation implies independence between the distribution of the latent variables $\mathbf{C}$ and the parameters π and $\boldsymbol{\theta}$[2]. Then the general form of the lower bound on the marginal likelihood $p(\mathbf{X}), \mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, for continuous variables $\pi, \boldsymbol{\theta}$ and a discrete variable $\mathbf{C}$ is:

$$\begin{aligned}\mathcal{L}_q(\mathbf{X}, \boldsymbol{\theta}, \mathbf{C}) &= \sum_{\mathbf{C}} \iint q(\mathbf{C}, \pi, \boldsymbol{\theta}) \log \left(\frac{p(\mathbf{X}, \mathbf{C}, \pi, \boldsymbol{\theta})}{q(\mathbf{C}, \pi, \boldsymbol{\theta})} \right) \mathrm{d}\pi \mathrm{d}\boldsymbol{\theta} \\ &= \mathbb{E}_q[\log p(\mathbf{X} \mid \mathbf{C}, \boldsymbol{\theta})] + \mathbb{E}_q[\log p(\mathbf{C} \mid \pi)] + \mathbb{E}_q[\log p(\pi)] + \mathbb{E}_q[\log p(\boldsymbol{\theta})] \\ &\quad - \mathbb{E}_q[\log q(\mathbf{C})] - \mathbb{E}_q[\log q(\pi)] - \mathbb{E}_q[\log q(\boldsymbol{\theta})]\end{aligned} \tag{4.6}$$

where the terms in this bound are available in closed form given conjugate priors $p(\pi)$ and $p(\boldsymbol{\theta})$. As previously mentioned (and illustrated in Fig. 4-1b), the prior for the distribution of the mixing coefficients, $p(\pi)$, is a Dirichlet distribution, $\mathrm{Dir}(\pi \mid \alpha)$, which is a conjugate prior to the multinomial distribution of the cluster assignments $\mathbf{c}$. The parameter α,

[2]Note that a further factorisation is implied by the independence assumptions in the model, that is, $q(\mathbf{C})q(\pi, \boldsymbol{\theta}) = q(\mathbf{C}) \prod_{k=1}^{K} q(\pi_k)q(\boldsymbol{\theta}_k)$. However, this factorisation is not a part of the assumption that leads to a tractable lower bound, which follows from the independence between $q(\mathbf{C})$ and $q(\pi, \boldsymbol{\theta})$.

called the concentration or scaling parameter, is chosen to be the same for all components and it can be interpreted as the effective prior number of observations associated with each component of the mixture [Bishop, 2006]. Alternatively, a nonparametric version of a Dirichlet distribution, a Dirichlet process, can be used to model the mixing coefficients. We give an introduction to Dirichlet processes and the associated nonparametric mixture model in the next section.

4.1.3 Dirichlet process

A Dirichlet process is a distribution over probability measures $G : \Theta \to \mathbf{R}^+$ such that $G(\theta) \geq 0$ and $\int_\theta G(\theta)d\theta = 1$ [Ferguson, 1973]. Alternatively, a Dirichlet process can be seen as the infinite-dimensional generalisation of the Dirichlet distribution. A sample from a Dirichlet process is a distribution $G \sim \mathcal{DP}(\alpha, G_0)$, where the real number α is the concentration or scaling parameter, and G_0 is called a base measure or a base distribution. Note that the base distribution is the expectation of the DP and the parameter α can be interpreted as the inverse variance of this process: for large α, the variance is small and the DP concentrates more of its mass around the mean implied by the base distribution [Teh, 2010]. As further explained in [Teh, 2010], two alternative names to parameter α are the strength parameter, referring to the strength of the prior when using the DP as a nonparametric prior over distributions, and the mass parameter, as the strength of this prior can be measured in terms of the sample size (or mass). A sample from the Dirichlet process can be thought of as an infinite weighted sum of delta functions at locations sampled from the base distribution G_0.

Stick breaking process A constructive definition of a DP is as follows: assume there exists a stick of length 1, and draw a sample from a beta distribution, $\beta_1 \sim \text{Beta}(1, \alpha)$ where α is the concentration parameter. Then break the β_1 part of the stick, and denote the piece on the left of the break as π_1. Then sample a value $\beta_2 \sim \text{Beta}(1, \alpha)$, and (assuming the remainder is of length 1) break the remainder of the stick at the point $\pi_2 := \beta_2$. This iterative procedure results in an infinite set of (mixture) weights, $\pi = \{\pi_k\}_{k=1}^{\infty}$ which add up to one and thus correspond to probabilities. This process can formally be written as:

$$\begin{aligned} \beta_k &\sim \text{Beta}(1, \alpha), \\ \pi_k &= \beta_k \prod_{l=1}^{k-1}(1 - \beta_l) = \beta_k(1 - \sum_{l=1}^{k-1} \pi_l), \end{aligned} \tag{4.7}$$

and denoted as $\pi \sim SBP(\alpha)$. Then define a distribution

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta}) \tag{4.8}$$

where δ denotes the indicator function and the values $\boldsymbol{\theta}_k$ are sampled from a base distribution G_0. The resulting distribution G is a sample from a DP, $G \sim \mathcal{DP}(\alpha, G_0)$, and it is discrete with probability one.

In our applications we use the stick breaking construction of the DP, however, to highlight two important properties of the DP we discuss a different formulation that allows us to easily draw samples from a DP.

Chinese restaurant process A Chinese restaurant process is an alternative formulation of a DP in which the analogy is as follows: the tables at a restaurant correspond to clusters, and the customers are the observations. Each new customer chooses to sit at an existing table with probability proportional to the number of people sitting at the table already, N_k, and otherwise chooses to sit at a new table k^* (i.e. open a new cluster). A draw from the base measure G_0 is associated with every table. The resulting distribution over the sample space is a random sample from a Dirichlet process.

Such generative process highlights two important properties of a DP:

1. The *rich get richer* property encourages re-sampling of the same values (using the Chinese restaurant analogy, a new customer is more likely to join a table with more customers).

2. Even though the base distribution is defined over a continuous sample space, with a nonzero probability two samples from a DP will have the same value (the tables in the Chinese restaurant process get assigned a value from a continuous base distribution, and this exact value can be associated with more than one customer as multiple customers are allowed to join the same table).

4.1.4 Dirichlet process mixture models

Due to the above mentioned properties (in particular, the stick breaking construction), a DP can be used as a prior for the parameters of a mixture model that allows for a flexible number of clusters.

Assume that the observations $\{\mathbf{s}_j\}_{j=1}^J$ are clustered, and the data points in each cluster are distributed according to some component distribution $p(\mathbf{s}_j \mid \boldsymbol{\theta}_{c_j})$ (*e.g.* a multivariate Gaussian) given the cluster assignment c_j. The parameters $\boldsymbol{\theta}_j$ (for instance, the mean and the precision of a multivariate Gaussian distribution) are i.i.d. according to the base distribution $G_0(\lambda)$ (*e.g.* a normal-Wishart) with some hyperparameter(s) λ. The mixing coefficients π are generated using the stick breaking construction with concentration parameter α; for small values of α only a few weights are significantly above 0, meaning that the data $\mathbf{s}_j$ are sampled from one of a few mixture distributions with parameters $\boldsymbol{\theta}_j$. The more data we see, the more likely it is that the parameters $\boldsymbol{\theta}_{c_j}$ are equal to one of the $\boldsymbol{\theta}_j$. An explicit construction of the DPMM based on a stick breaking representation is:

$$\begin{aligned} &\pi \sim SBP(\alpha), \\ &\boldsymbol{\theta}_i \sim G_0(\lambda), \quad i = 1, 2, \ldots, \\ &c_j \sim \text{Mult}(\pi), \quad j = 1, 2, \ldots, J, \\ &\mathbf{s}_j \sim p(\mathbf{s}_j \mid \boldsymbol{\theta}_{c_j}). \end{aligned} \tag{4.9}$$

Alternatively, the DPMM is written a follows:

$$\begin{aligned} &G \sim \mathcal{DP}(\alpha, G_0(\lambda)), \\ &\boldsymbol{\theta}_i \sim G, \\ &\mathbf{s}_j \sim p(\mathbf{s}_j \mid \boldsymbol{\theta}_{c_j}), \end{aligned} \tag{4.10}$$

where G is a sample from the Dirichlet process. The graphical model of a DPMM is given in Fig. 4-1c.

Typically inference in DPMMs is performed using a collapsed Gibbs sampler, which can be very slow. Alternatively, variational inference is used in a similar vein as described in Sec. 4.1.2. In particular, since the data likelihood is not available in closed form, it is approximated using a variational lower bound [Blei and Jordan, 2005]. The posterior distributions over the parameters β_i, $\boldsymbol{\theta}_i$ and c_j are approximated with factorised beta, Gaussian and multinomial variational distributions respectively:

$$q(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{c}) = \prod_{t=1}^{T-1} q_{\gamma_t}(\beta_t) \prod_{t=1}^{T} q_{\tau_t}(\boldsymbol{\theta}_t) \prod_{j=1}^{J} q_{\phi_j}(c_j), \tag{4.11}$$

where T is the maximal number of clusters in the mixture (*i.e.* the infinite mixture model is truncated for the variational approximation). This approximation allows us to

obtain a lower bound on the data likelihood:

$$\log p(\boldsymbol{s}_j) \geq \mathcal{L}_q(\boldsymbol{s}_j, \boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{c}) := \mathbb{E}_q[\log p(\boldsymbol{s}_j, \boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{c})] - \mathbb{E}_q[\log q(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{c})], \tag{4.12}$$

where both terms are analytically tractable, and we refer the reader to [Blei and Jordan, 2005] for their exact expressions.

We now return to the alignment model and after a short summary of the model of sequences and warps in Sec. 4.2 we introduce the alignment objective based on the MMs in Sec. 4.3.

4.2 Recap of model over sequences and warps

We start by providing a brief recap of the relevant parts of the methodology from Sec. 3.3. Let us assume that we are given J noisy observed sequences $\{\mathbf{y}_j\}_{j=1}^J$ where each sequence comprises of N time samples $\mathbf{y}_j := (y_{j,1}, \ldots, y_{j,N}) \in \mathbb{R}^N$. We consider each sequence to be generated by sampling a latent function $f_j(\cdot)$ at points $\{g_j(x_n)\}_{n=1}^N$, *i.e.* $y_{j,n} = f_j(\, g_j(x_n)) + \varepsilon_j$, where $\mathbf{x} := (x_1, \ldots, x_N) \in \mathbb{R}^N$ are known evenly-spaced inputs (same for all sequences), $\{g_j(\cdot)\}_{j=1}^J$ are the warping functions (different for different sequences) and $\varepsilon_j \sim \mathcal{N}(0, \beta_j^{-1})$ is i.i.d. Gaussian noise.

We are interested in the case where the number of distinct latent functions $f_j(\cdot)$ is smaller than J, *i.e.* some of the observed sequences are generated from the same underlying functions. In this setting, the observed sequences were *misaligned* by applying different time warpings $g_j(\cdot)$ to the input locations. We treat the *aligned* sequences $\mathbf{s}_j := (f_j(x_1), \ldots, f_j(x_N)) \in \mathbb{R}^N$ as if they were observed, and hence we refer to $\{\mathbf{s}_j\}$ as pseudo-observations, and we wish to infer these aligned sequences $\mathbf{s}_j := (f_j(x_1), \ldots, f_j(x_N)) \in \mathbb{R}^N$ from the observations $\mathbf{y}_j$. The warping functions are modelled using the parametrisation defined in Sec. 3.3.

We now introduce the alignment objective based on the mixture models.

4.3 Alignment objective

Instead of modelling each sequence $\mathbf{y}_j$ in isolation, we wish to encourage the model to use the smallest possible number of distinct latent functions $\{f_j(\cdot)\}$ to explain the data. One way to introduce such a constraint is to add a regularisation term that encourages

clustering of aligned sequences $\{\mathbf{s}_j\}$ into a small number of clusters. Indeed, the subset $\{\mathbf{s}_{c_j}\}$ of the aligned sequences that belong to the same cluster would be similar to each other, meaning that the corresponding latent functions $\{f_{c_j}(\cdot)\}$ are also similar when evaluated at $\mathbf{x}$. Meanwhile, the GP prior on $\{f_{c_j}(\cdot)\}$ enforces that the functions are smooth and nearby values are correlated.

We propose clustering $\{\mathbf{s}_j\}$ using an alignment objective based on mixture models which were introduced in Sec. 4.1.1. In the context of our alignment application, we assume that the number of mixture components (*i.e.* the number of underlying functions that generated the observations) and the distribution of the data within each cluster are unknown and need to be inferred from the data.

In practice, the standard Gaussian mixture model defined at the beginning of Sec. 4.1.1 suffers from singularities when learning the maximum likelihood solution (without any priors) [Bishop, 2006]. This occurs when one of the components coincides with a specific data point, collapsing the variance of that component to zero. This problem can be circumvented by using a Bayesian mixture model where the parameters of the components (the mean and the variance in the case of a mixture of Gaussians) are themselves random variables with associated prior distributions. Furthermore, the mixture weights are also considered to be random variables, and the prior on these weights (e.g. a Dirichlet) can be used to impose our belief about the potential number of components present in the data.

In a Bayesian mixture model, the number of clusters K is set in advance and model comparison is performed by estimating the marginal likelihood of the data for a different number of clusters and choosing K with the highest likelihood. An alternative to this involves defining nonparametric priors which do not require the number of clusters to be specified a priori, and reflect greater initial uncertainty, in this case, associated with the unknown number of clusters [Hjort et al., 2010]. Therefore, we also consider the use of a DP mixture model where the DP provides a nonparametric prior over the number of clusters. As a nonparametric mixture model, DPMM allows us to automatically infer the number of clusters (*i.e.* distinct aligned sequences) from the data. The locations of the mixture components and the mixing weights are distributed according to a sample from a Dirichlet process and depending on the parameters of the process, such a prior seeks to explain the data using only a few mixture components, corresponding to aligning the underlying latent functions.

The alignment effect is a result of the choice of the base distribution of the mixture components and the corresponding priors. A natural option for the base distribution

for the alignment application is a zero-mean spherical Gaussian (*i.e.* a multivariate Gaussian with diagonal covariance function) imposing a strong local pattern and hence limiting the variability of the data points within each cluster. Meanwhile, the cluster assignment takes place globally. In the case of the BMM and the DPMM, the prior over the mixing proportions incorporates our belief about the potential number of distinct groups in the data set.

Toy example: BMM alignment objective

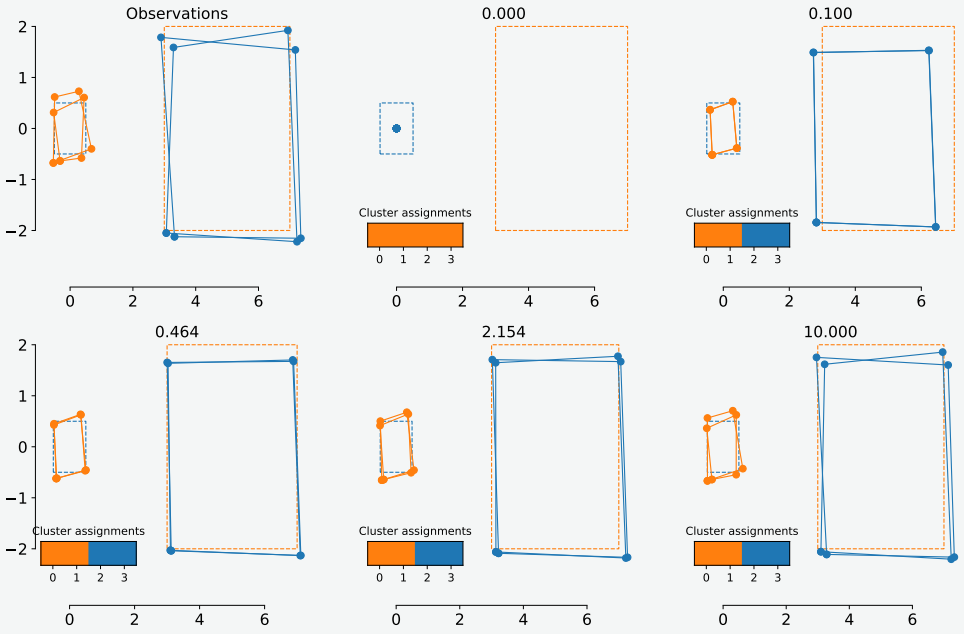


Figure 4-2: Aligned quadrilaterals for different values of the warping prior γ using the BMM alignment objective. The coloured squares at the bottom of each figure show colour-coded cluster assignments. Unlike in the analogous examples in Ch. 2 and Ch. 3, here the local alignment objective dominates; the rectangles get aligned to two clusters which are independent given the cluster assignment. Within the clusters, the rectangles are well aligned to each other for small and intermediate values of γ, while for large values of γ the cost of aligning the squares is lower than the regularisation term that penalises transformations, hence the alignment within the groups diminishes.

Let us compare the behaviour of the mixture model alignment objective to the two previously considered cases (the energy objective in Fig. 2-2 and the GP-LVM objective in Fig. 3-2). As previously, given four quadrilaterals, we define the aligning transformations (warpings) as independent additive transformations of the vertices and impose an L^2 prior for the transformations to be small ($\gamma\,||w||^2$).

Fig. 4-2 shows the alignment of the quadrilaterals using BMM with $K = 10$ and K-means initialisation.

Without the prior on the warpings (*i.e.* with $\gamma = 0$), all observations collapse to a single point (and hence a single cluster) at $(0, 0)$ in accord with the prior on the clusters (with are zero-mean Gaussians). However, including a cost on the transformations (*i.e.* setting $\gamma > 0$) results in a BMM with two clusters each of which contains two squares. Within each of the two clusters the prior on the cluster distributions encourages the squares to be aligned. For small values of γ the alignment is very accurate within each cluster, and it deteriorates with increasing cost of transformations. Note that the observations are conditionally independent given the cluster assignments. This illustrates the property of *local* alignments that distinguishes the MM alignment objective from the GP-LVM objective which uses the length-scale as a parameter to balance the local and the global behaviour, and the energy objective which considers the alignment task globally.

4.4 Alignment model

We want the observed data $\{\mathbf{y}_j\}$ and aligned sequences $\{\mathbf{s}_j\}$ to be modelled by the GPs (by $f(g(\mathbf{x}))$ and by $f(\mathbf{x})$ respectively), and the aligned sequences $\{\mathbf{s}_j\}$ to be regularised by clustering them into groups. Therefore, we simultaneously maximise the GP data likelihood given in Eq. 3.11 and the lower bound $\mathcal{L}_q(\mathbf{s}_j)$ on the MM likelihood given in Eq. 4.6.

The GP likelihood in Eq. 3.11 includes the evaluation of the warping GP, $\mathbf{g}_j$, which we cannot integrate out analytically. Thus, following the argument from Sec. 3.3, we obtain a point estimate by including the likelihood of $\mathbf{g}_j$ in the objective, and directly optimising $\{\mathbf{v}_j\}$, which parametrise $\mathbf{g}_j$, as defined in Eq. 3.9.

Now consider the lower bound on the BMM likelihood, given in Eq. 4.6. As previously discussed, we use a Dirichlet prior on the weights π, and conjugate priors for the multivariate Gaussian distribution with unknown mean $\boldsymbol{\mu}_k$ and precision $\boldsymbol{\Lambda}_k$, specifically,

normal-Wishart priors:

$$
\begin{aligned}
p(\mathbf{Z} \mid \pi) &= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi^{c_{nk}}, \\
p(\pi) &= \mathrm{Dir}(\pi \mid \alpha), \\
p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k \mid \mathbf{m}_0, (\beta_0 \boldsymbol{\Lambda})^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k \mid \mathbf{W}_0, \nu_0),
\end{aligned}
\tag{4.13}
$$

where the prior over the parameters of the base distribution, $p(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ factorises over the clusters, and there is a dependence between the distribution of the mean and the precision (this is also illustrated in the graphical model in Fig. 4-1b). A normal-Wishart prior implies that the component distributions $p(\mathbf{s}_j \mid \boldsymbol{\mu}, \boldsymbol{\Lambda})$ have a non-diagonal covariance (this means that the component distributions are not spherical Gaussian). For our application, imposing a diagonal covariance might be beneficial in terms of encouraging alignment of data within clusters. In particular, in the alignment model the covariance between sequences should be explained using the warpings $\mathbf{g}_j$ as well as possible, rather than by constructing more complex component distributions. Therefore, we also consider the case where the precision Λ is constrained to be a diagonal matrix $\sigma^{-1}\mathbb{I}$ (and hence the covariance is $\sigma\mathbb{I}$). Then the conjugate prior for the means of the mixture components is a zero-mean multivariate Gaussian, $\boldsymbol{\mu}_k \sim \mathcal{N}(\mathbf{0}, \rho\mathbb{I})$ for some $\rho > 0$ and $\mathbf{m}_0 = \mathbf{0}$.

As discussed in Sec. 4.1.2, assuming that the variational distribution factorises as:

$$
q(\mathbf{C}, \pi, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{C})\, q(\pi, \boldsymbol{\mu}, \boldsymbol{\Lambda}),
\tag{4.14}
$$

the variational posteriors are tractable and are conjugate to the priors, *i.e.* a multinomial for $\mathbf{C}$, a Dirichlet for π and a normal-Wishart with an induced factorisation over the clusters for $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$. We refer the reader to [Bishop, 2006] for the exact expressions of these distributions, and the evidence lower bound $\mathcal{L}_q(\mathbf{s}_j)$.

In the case of the DPMM, the prior on the weights π is implied by the stick breaking process with parameter α and the prior on the cluster assignments is a multinomial with parameter π (*i.e.* the same as in the BMM case). The prior on the parameters of the base distribution, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, can be chosen as in the BMM case (see Eq. 4.13). Then the inference is performed by maximising the evidence lower bound $\mathcal{L}_q(\mathbf{s}_j)$ as described in Sec. 4.1.4. Note that while the variational distribution is truncated to T clusters (or equivalently, T sticks in the SBP), there are no such constraints on the full model, and hence variational inference provides an approximation to the full (infinite) model [Blei and Jordan, 2005]. For the full expressions of the terms in the lower bound

of the DPMM likelihood, see, for example, [Bietti and Chizat, 2014].

Then the optimisation objective for the alignment task is:

$$\mathcal{J}(\mathbf{s}, G, \boldsymbol{z}) = \sum_{j=1}^{J} \left[p\left(\begin{bmatrix} \mathbf{s}_j \\ \mathbf{y}_j \end{bmatrix} \middle| \mathbf{g}_j, \mathbf{x}, \theta_j \right) + \mathcal{L}_q(\mathbf{s}_j) + p(\mathbf{g}_j \mid \mathbf{x}, \omega) \right], \tag{4.15}$$

where $\mathcal{L}_q(\mathbf{s}_j)$ denotes the evidence lower bound for either the BMM or the DPMM. We maximise this objective w.r.t. the pseudo-observations of the aligned sequences $\{\mathbf{s}_j\}$, the MM variational parameters, the auxiliary variables of the warping functions $\{\mathbf{v}_j\}$, and the hyperparameters of the GPs and the MM. Among the hyperparameters of the MM are the scaling parameter α, the parameters of the base distribution, and the parameters of the mixture components. We found that for the alignment application (especially beyond the toy examples) the model with the full normal-Wishart prior is inferior to the constrained model where the base distribution is a zero-mean multivariate Gaussian while the mixture components are modelled using a multivariate Gaussians with a diagonal-covariance with the same variance for all components. Furthermore, we constrain the model by making an assumption that the variance of the mixture components is equal to $1/\beta$, *i.e.* the estimated noise in the GP fits, which we assume to be the same for all sequences. This implies that the variance in each cluster should correspond to the noise in the observations and (aligned) sequences whose variability cannot be explained by this noise model should be assigned to different clusters.

4.5 Experiments

First, we consider the utility of MMs as an alignment objective on a toy data set, and then test it on a more complicated real-world data set of heart beat recordings.

4.5.1 Synthetic data set

We consider a set of 10 sequences generated using $\text{sinc}(\mathbf{x})$ and $\mathbf{x}^3$ functions, where $\mathbf{x}$ is a linearly spaced vector of values in the interval $[-1, 1]$, and warped using randomly generated monotonically increasing warping functions. To analyse and compare the performance of different approaches, we compute the following metrics:

(1) the mean (median) alignment error as the sum of means (medians) of pairwise distances between observations within each group of sequences in the N-dimensional

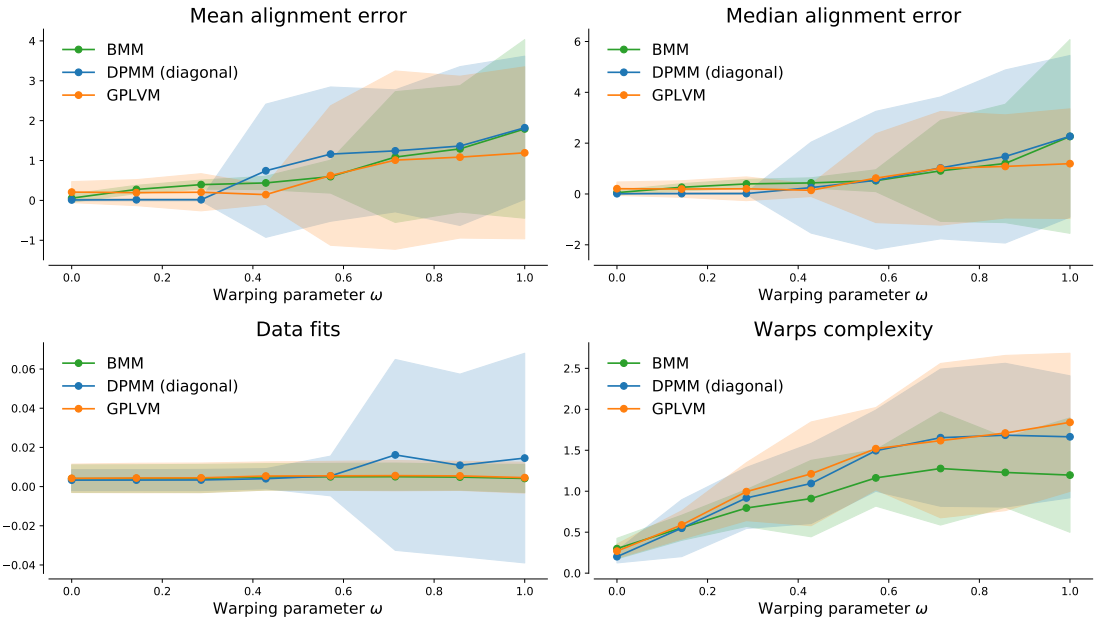


Figure 4-3: Comparison of alignment error, data fit and warp complexity for alignment objectives based on a GP-LVM, a BMM and a DPMM. Solid lines show mean values across 15 trials and the solid region corresponds to 2 standard deviations.

space,

(2) the data fit as the standard deviation of the estimated observational noise ($\sqrt{1/\beta}$), and

(3) the warping complexity as the sum of the absolute values of differences between components of $\mathbf{v}_j$, which corresponds to the total variation of variables that define the warps.

The warpings are parameterised with a single parameter ω, where $\omega = 0$ corresponds to no warping and the warps get progressively larger as ω increases.

Fig. 4-3 provides a quantitative comparison between the alignment objectives based on MMs and on the GP-LVM in terms of the three criteria outlined above as a function of the warping parameter ω. We report the results for a BMM with conjugate priors, which include a full covariance matrix for the mixture distributions, and for a DPMM with a diagonal covariance matrix for the mixture distributions. In this setting where only two clusters are present, the difference between a BMM and a DPMM is not significant, and it is likely to be due to the differences in the inference procedure rather than in the model assumptions. The GP-LVM objective leads to slightly lower mean and median

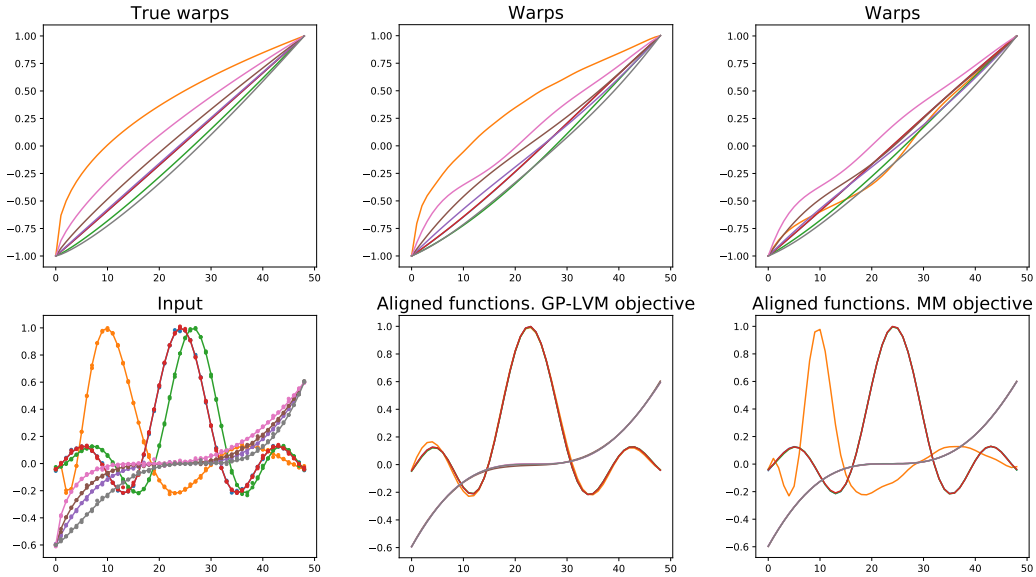


Figure 4-4: An example of the behaviour of the models with the GP-LVM and the BMM alignment objectives for the warping parameter ω of 0.8 in Fig. 4-3.

alignment error, especially for larger warps. This may be explained by the fact that the MMs tend to separate highly warped inputs into new clusters, as illustrated in Fig. 4-4. If a sequence is an outlier due to a large warp, the MM tends to create a new cluster for it, while the GP-LVM favours the solution that recovers the two groups of sequences, which sometimes leads to higher alignment error within the groups. Here we report the alignment error for the two groups, hence in the case where the MM implies more than 2 clusters, the total alignment error is much higher than it would be if the additional cluster was taken into consideration when computing the error.

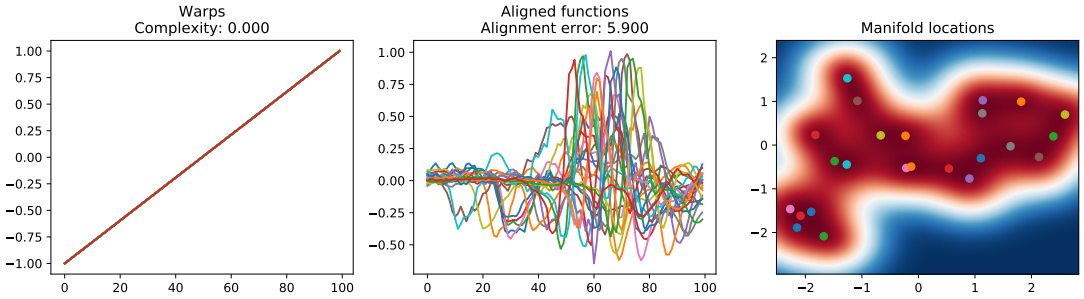
The example given in Fig. 4-4 is an illustration of the typical behaviour of the models with the two types of alignment objectives. However, as discussed in Sec. 3.4 and Sec. 4.3, this behaviour is controlled by the parameters of the corresponding alignment objectives. For example, in the GP-LVM case, initialising or fixing the kernel lengthscale to a large value leads to weaker correlations of sequences, which in turn implies that the sequences with large warps no longer get aligned to the corresponding groups; in that case the result for the GP-LVM objective would look similar to the result achieved by the MM objective in Fig. 4-4. Similarly, in MMs, adjusting the concentration parameter and the variance of the mixture components leads to a different number of clusters, and potentially, different alignment within the clusters (*i.e.* higher mixture variance would weaken the effect of such an alignment objective).

4.5.2 Heartbeats data

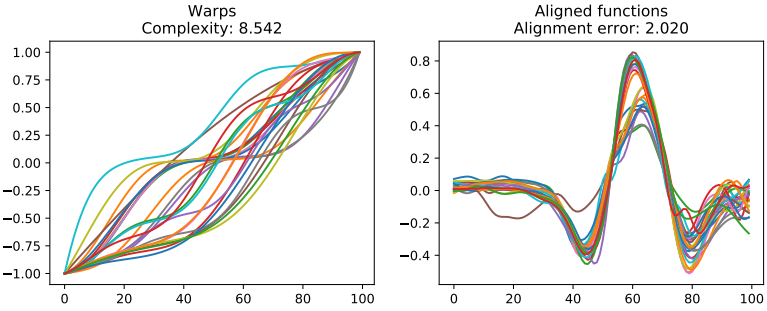
We now consider a data set of heart beat recordings that vary temporally depending on the age, health, and state of the subject [Bentley et al., 2011] (this data set was previously introduced in Sec. 3.7.4). The data set contains 24 beat recordings of two types, where each recording is 100 frames long. While standard approaches rely on signal filtering as a pre-processing step [Bentley et al., 2011], GPs with a Matérn 3/2 kernel are used to model the sequences over time; this takes into account the rapid variations in the recordings while also limiting the effect of the uninformative high frequency noise. For this experiment we use the DPMM with a constrained distribution over the mixture components (a Gaussian with a diagonal covariance with the variance equal to the observational noise β^{-1} estimated when fitting the GPs over time) and a constrained base distribution (zero-mean Gaussian over the means of the mixture distributions).

Fig. 4-5 gives a comparison of the alignment and the clustering of a set of heartbeats for the three alignment objectives: the energy objective (Sec. 2.2.2), the GP-LVM and the DPMM. The observational noise values β^{-1}, estimated by fitting the GPs to the sequences (and assumed to be the same for all sequences), are 0.08, 0.041 and 0.03 for the energy, the GP-LVM and the DPMM objectives, respectively, while the observational noise estimated without using an alignment objective is 0.02 (which we refer to as the true noise). While all three objectives estimate a higher value for the noise than the true noise, the latter two objectives allow the model to stay more faithful to the observed data without significantly compromising on the quality of the alignment. Meanwhile, the energy objective estimates the noise to be higher in order to allow for alignment to a single cluster, even though such a model is not supported by the observations. Both the GP-LVM and the DPMM infer that there are two types of recordings and align them within the two groups. In the DPMM, the clustering is explicit while the GP-LVM learns a latent space where the two groups are located far from one another relative to the distances of the latent points within the two groups. On the contrary, the energy objective relies on the flexibility of the warps and compromises on the data fits in order to align all the sequences together. Using the alignment objective defined in terms of a MM, the model estimates warping functions which are closer to an identity function and also fits the observations better as explained above, therefore, it might be the preferred model, even though the alignment error is higher than for the GP-LVM objective.

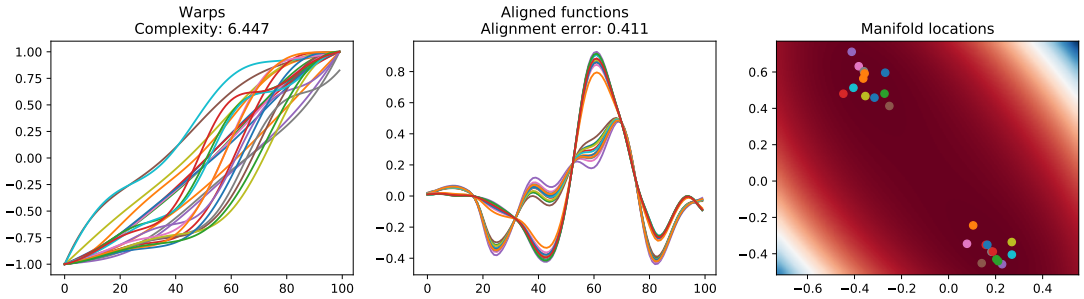
In practice, the DPMM is sensitive to the initialisation of the latent variables. A possible initialisation that can be used for the MM is K-means clustering, however, it requires the number of clusters to be picked a priori, while for our applications we assume the number



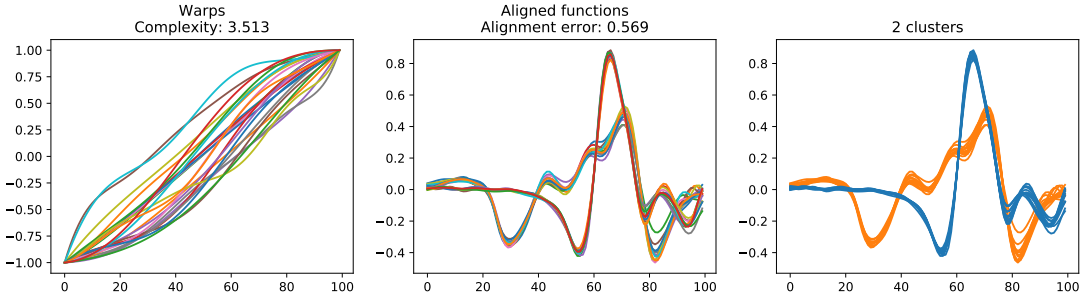
(a) Input sequences. The corresponding GP-LVM manifold does not uncover the two different types of heartbeats.



(b) Energy alignment.



(c) GP-LVM alignment.



(d) DPMM alignment.

Figure 4-5: Alignment of heartbeats data [Bentley et al., 2011].

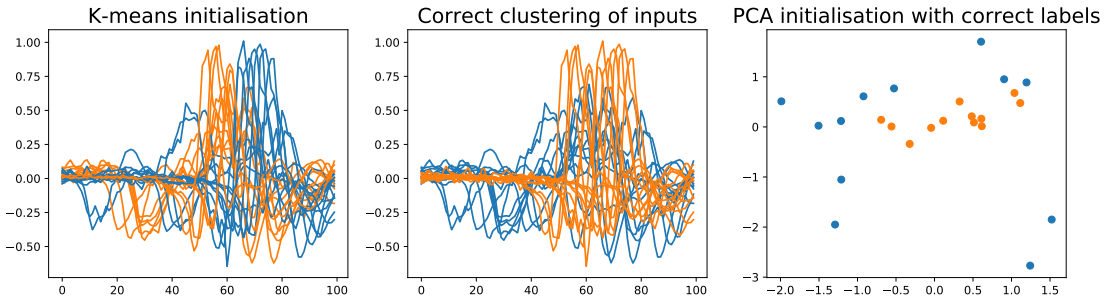


Figure 4-6: Comparison of initialisations from the MMs and the GP-LVM on the heartbeats data from Sec. 4.5.2. An illustration of K-means clustering (using 2 clusters) of the heartbeats data is shown on the left, while the correct clustering is shown in the middle. In this case, the K-means algorithm separates the sequences based on the horizontal shift (the sequences with the peak before approx. $t = 70$ are assigned to one cluster, which the sequences with the peak beyond $t = 70$ get assigned to a different cluster). Meanwhile, the clustering that corresponds to the temporally aligned sequences is by definition invariant to such shifts. Based on this observation (and the fact that K-means clustering requires the number of clusters to be chosen by hand), it is more appropriate to initialise the MM with random cluster assignments. The GP-LVM is initialised using PCA (the figure on the right), which does not discover the two temporally aligned clusters that we hope to find in this data (shown in orange and blue in the figure).

of clusters to be unknown. Meanwhile, the latent space of the GP-LVM is typically initialised using PCA. As illustrated in Fig. 4-6, the initialisation can be misleading for both the GP-LVM and the MM, even when the correct number of clusters is known beforehand. In practice, we found that the PCA initialisation for the GP-LVM works at least as well as a random initialisation (and typically better). Choosing the initialisation for the DPMM is much harder, both in terms of initialising the latent variables and the parameters of the latent distributions, as discussed in [Blei and Jordan, 2005], and it might require multiple runs with random restarts in order to find a solution that provides good fits to the observations and good alignment within clusters.

4.5.3 Rigid alignment

The alignment problem is defined using a space of transformations that are applied to the observations to bring them into correspondence. The transformations used throughout this thesis are defined to be one dimensional monotonic warpings (except for the motion capture example where the extension to higher dimensions was trivial as the same warping was applied to every dimension of the inputs). Extending the concept of monotonicity beyond the one dimensional case is not trivial and there exists no standard

definition for multivariate monotonicity [Clarke et al., 1993, Dyke et al., 2013]. Hence when considering alignments in higher dimensions it is necessary to define additional constraints, such as preservation of area/volume in two/three dimensions. In this section we move away from the task of aligning 1-dimensional temporal sequences and consider a different yet related task of aligning 3-dimensional meshes.

So far in this work we have discussed the task of aligning discrete signals, which are assumed to be noisy evaluations of continuous functions at warped inputs. A more general setting is as follows: given a set of sequences (discrete signals) $\{\mathbf{y}_j\}_{i=1}^J$, $\mathbf{y}_j \in \mathbb{R}^D$, we want to find transformations $s_j : \mathbb{R}^D \to \mathbb{R}^D$, such that the transformed sequences $\{s_j(\mathbf{y}_j)\}_{i=1}^J$ are aligned to each other according to some metric. Clearly, there is a trivial solution to this task with s_j mapping every sequence to the first one. Therefore, we need to impose certain assumptions on $\mathbf{y}_j$ and s_j for the alignment to have a meaningful interpretation. One such set of assumptions states that $\mathbf{y}_j$ are samples of a continuous function, and s_j are the warps of the input (*i.e.* the argument). Another option is to assume that $\mathbf{y}_j$ are certain physical objects, and s_j are a certain class of transformations of these objects. We discuss an example of the latter setup in this section.

The problem can be described as follows: assume that there exist a set of 3-dimensional meshes that correspond to one underlying structure (e.g. a cube) but that have been transformed by rotating and translating the object in space. If the set of transformed shapes have a temporal structure (for example, they correspond to a set of meshes from a 3-dimensional motion capture sequence) then we refer to this set as a sequence. The task is then to find the rotations and translations that superimpose the set of objects with respect to each other. A possible real-life application includes the task of rigid stabilisation of meshes (in particular, meshes of animate objects), which refers to the removal of the rigid movements for each frame in a sequence of motion. Head stabilisation is a common task in computer graphics where it is used as a pre-processing step for motion capture data before it is passed on to an artist who may want to generate novel meshes (e.g. heads with different expressions) that are not affected by rigid movements of the head.

More specifically, we assume that for each mesh $\mathbf{Y}_j \in \mathbb{R}^{D\times 3}$ in a sequence there exists a transformation consisting of a rotation and a translation such that transforming the mesh removes the rigid motion associated with the movement of the object. In our framework this is done as follows. We define the rotations and the translations for each input $\mathbf{Y}_j$ as latent variables and add a regulariser to the transformed meshes $\mathbf{Y}_j^*$.

Modelling transformations A rigid motion can be decomposed into a rotation around a given axis and a translation by a vector. Suppose we have a vector $\mathbf{v} \in \mathbb{R}^3$, its rotation can be represented by a matrix multiplication $\mathbf{v}_{\text{rot}} = M(\mathbf{v} - \mathbf{a}) + \mathbf{a}$ with $M \in \text{SO}(3)$, i.e. an orthogonal matrix with determinant 1, and we introduce an additional parameter $\mathbf{a} \in \mathbb{R}^3$ for the point lying on the rotation axis to allow rotations around arbitrary axes. Finally adding a translation $\mathbf{t} \in \mathbb{R}^3$, we obtain a model for the aligning transformation $s(\mathbf{v})$:

$$s(\mathbf{v}) = \mathbf{v}_{\text{rot}} + \mathbf{t} = M(\mathbf{v} - \mathbf{a}) + \mathbf{a} + \mathbf{t}. \tag{4.16}$$

Such a decomposition induces 9 parameters for each s_j: 3 parameters of the rotation matrix (two for the axis of rotations, and one for the angle), and 3 parameters for each of $\mathbf{a}$ and $\mathbf{t}$. The alignment task then consists of finding the parameters of s_j such that $s_j(\mathbf{Y}_j)$ are as similar as possible given the allowed transformations.

Rotations can alternatively be modelled not as orthogonal matrices, but rather as unit quaternions. Such a quaternion can be represented as

$$\mathbf{q} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2}, \tag{4.17}$$

where the unit vector $\mathbf{u} = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$ is the axis of rotation, and θ is the angle of rotation around this axis. Applying a rotation to an arbitrary vector $\mathbf{v}$ corresponds to a quaternion conjugation (where $\mathbf{v}$ is considered as a quaternion without a real part): $\mathbf{v}_{\text{rot}} = \mathbf{q}\mathbf{v}\mathbf{q}^{-1}$. There are no conceptual differences between representing rotations as matrices or as quaternions, however, the latter allows for simple linear interpolations between rotations and for an efficient numerical implementation using cross-products for computing quaternion conjugations. Quaternions are hence more conventional in applications within computer graphics [Shoemake, 1985].

Alignment objective Having defined the transformations s_j (each parametrised by a parameter vector θ_j), we choose an optimisation objective $\mathcal{L}(\Theta)$, $\Theta = (\theta_1, \ldots, \theta_J)$, such that the minimum of such an objective corresponds to transformations aligning original meshes $\mathbf{Y}_j$. As discussed in Sec. 2.2.2, a possible objective could be the minimisation of the sum of Euclidean distances between the transformed meshes, i.e.

$$\mathcal{L}(\Theta) = \sum_{j=1}^{J} \sum_{k=i+1}^{J} ||s_j(\mathbf{Y}_j) - s_j(\mathbf{Y}_k)||, \tag{4.18}$$

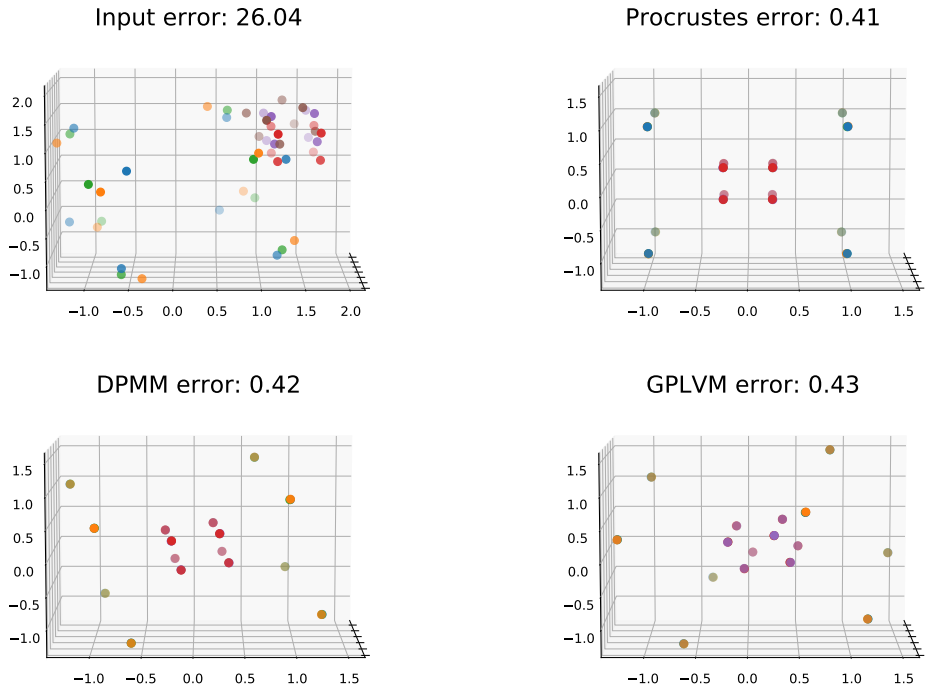


Figure 4-7: Comparison of alignment objectives for alignment of 3D meshes of cubes using rigid transformations.

which resembles the loss functions used in classical shape alignment methods, such as Procrustes analysis [Gower, 1975]. As an alternative, we study the GPLVM and the DPMM as regularisers for these transformations.

Consider a data set that consists of 3-dimensional meshes of 6 cubes that were generated by applying a random rotation to either a unit cube centered at the origin or a smaller cube (with side length equal to 0.5) centered at $(1.25, 1.25, 1.25)$. Furthermore, the data is corrupted by i.i.d. Gaussian noise, $\epsilon \sim \mathcal{N}(0, 0.01)$. Fig. 4-7 illustrates the alignment of the cubes using the three different objectives: the Procrustes analysis, the GP-LVM and the DPMM. There are two groups of data corresponding to the two input cubes, and all three objectives uncover the two groups and align the inputs within the groups. This is to be expected in the case of the GP-LVM and the DPMM objectives (as previously discussed) as these approaches have the flexibility to capture the two groups in their latent representations (the latent space for the GP-LVM and the latent label assignments and clustering for the DPMM). However, the Procrustes analysis has no latent representation and it still recovers the two underlying cubes and achieves equivalent alignment error (calculated as a sum of pairwise distances between meshes within the same cluster) as the other two alignment objectives. This happens because

the transformations used in this problem are rigid, and hence very constrained: it is impossible to align the two cubes using only rotations and translations (it is only possible if an additional transformation - scaling - is added to the aligning transformations). As a result, the optimal solution using the Procrustes analysis is the following. Assume we are given two groups with three elements in each, $[1, 2, 3]$ and $[4, 5, 6]$. Consider pairwise alignment where all the meshes are aligned to 1: then 1, 2 and 3 are aligned well since they correspond to the same underlying mesh, and the other three meshes, $[4, 5, 6]$, get aligned to a reference (in this case 1) as well as possible, so they are all transformed to match 1 as well as possible, which results in them being aligned to each other as well. It is worth noting, conversely, that the alignment error for the two alternative alignment objectives, the GP-LVM and the DPMM, is equivalent to the error achieved by Procrustes analysis, which in this case is the optimal alignment error.

4.6 Discussion

The work discussed in this chapter examines the potential of using a mixture model objective as an alignment objective when the number of underlying functions that generated the observations, is unknown. This approach offers a probabilistic model of alignments that is able to implicitly align inputs that contain temporal variations. The proposed approach builds on the previous work discussed in Ch. 3, and the main difference between the two approaches can be summarised as follows. The MM performs explicit clustering in the data space (and automatically estimates the number of clusters), while the GP-LVM performs dimensionality reduction and looks for a set of sequences that exhibit a simple structure in the latent low dimensional space (and provides an explicit latent space). While the experimental results suggest that both models perform well on real and synthetic data sets, they display different qualitative behaviour. The GP-LVM aligns the sequences constrained to them staying on a low-dimensional structure; in comparison, the MM does not have this global constraint and aligns sequences in each estimated cluster independently from the other clusters.

One of the difficulties with MMs is the inherent ambiguity of clustering [Ben-David, 2018]. Clustering typically refers to assigning similar data points to the same cluster while keeping dissimilar data points in separate clusters [Shalev-Shwartz and Ben-David, 2014]; however, similarity in the broad sense is not a transitive relation, unlike cluster assignment. Furthermore, there is no universal metric for clustering as different applications may require the same data to be partitioned in different ways [Ben-David, 2018]. Consequently, any practical clustering method should be designed with the

application in mind. Same is true in the context of alignments, where a given data set may be partitioned in a multitude of different ways, and the MM priors help guide the algorithm towards a solution that is likely given our prior belief about the structure of the data set.

Another difficulty with the MM-based methods (that at least partially stems from the ambiguity of clustering) is related to the fragility of the inference schemes for MMs. For example, the ML estimate computed using EM requires an iterative procedure which is hard and expensive to optimise jointly with the GPs fitting the data. Meanwhile, the variational inference in BMM is sensitive to the initialisations of the parameters and to the choice of the hyperparameters in the variational approximation of the lower bound on the likelihood. Therefore, a successful application of this method on new data sets relies on the choice of priors which constrain the problem enough to find consistent solutions. The additional flexibility offered by the DP prior over a finite distribution over clusters does not seem significant for the applications considered in this work. However, a regularisation objective based on a DPMM might be applicable to other tasks where it is important to keep the model very flexible and to allow the complexity of the model to increase as more data becomes available.

In practice, the models that use the GP-LVM objective are easier to train and they are more consistent when applied to different data sets. This is at least partially due to the ML estimates used in the GP-LVM objective rather than variational approximations obtained by maximising the lower bound on the evidence used in the MMs. In addition, it might be the case that the GP-LVM recovers better from poor initialisations than a MM due to (1) the continuous nature of the latent representation in a GP-LVM in contrast to the discrete cluster assignments in a MM, and (2) the fact that initialising the variational parameters in the MMs is problematic [Blei and Jordan, 2005] while the inference scheme we use for the GP-LVM does not have the additional parameters. In future work we propose developing a model which makes use of the global low-dimensional structure of the GP-LVM and the unconstrained alignment within clusters of the MMs.

Finally, evaluating the quality of the model given the data is not always straightforward, as discussed in Sec. 4.5.2. Relying on the alignment error alone might lead to degenerate solutions where the sequences are all aligned to a single cluster by compromising on the quality of the data fitting, and/or by choosing extreme warping functions. Note that in applications that are by design very constrained, for example when the transformations (equivalent to the warpings) are rigid as explained in Sec. 4.5.3, all of the discussed alignment objectives perform equally well in terms of uncovering the structure in the data and aligning the observations.

5

Further discussion of alignment model

The models proposed in Ch. 3 and 4 can be split into two constituent parts – the model of the observations (*i.e.* fitting the GPs with warped inputs to the observations) and the alignment objective – where the two parts are designed separately (but optimised jointly). For instance, the alignment objective can be formulated as an L^2 norm or as a probabilistic objective, such as a GP-LVM or a DPMM, irrespective of the approach used to model the observations. Such a formulation corresponds to using the (probabilistic) alignment objective as a prior (or a regulariser) over the possible solutions rather than as a generative model of the observed data.

Given the probabilistic nature of the two parts of our alignment model, it is reasonable to ask whether this model can be viewed as a joint probabilistic model rather than as a probabilistic model and a probabilistic regulariser that are jointly optimised. Formulating a joint probabilistic model implies constraining the types of interactions between the variables, especially in terms of the independence structure that is appropriate given one's knowledge of the data and the task.

In this chapter we first revisit the models formulated in Ch. 3 and 4 and offer multiple different perspectives on these models. We examine the independence structure imposed by the alignment objectives proposed in the previous chapters. Furthermore, we consider the meaning and effect of the different noise terms and visualise the dependence structure in the models using directed acyclic graphs (DAGs). In doing so, we begin a more general discussion of the uncertainties in the alignment model.

We then discuss an alternative formulation of the alignment model that builds on

the ideas of matrix-normal distributions and multi-output GPs [Álvarez et al., 2012]. A multi-output GP jointly models the correlations between the sequences as well as the correlations between the features of the data making this framework particularly appealing for the alignment task. We introduce a simple extension to the multi-output GPs that uses the correlations between the features of the data to uncover the underlying (aligned) sequences. We illustrate the behaviour of the simple model on toy data, and outline some directions for future work.

We end this chapter with a discussion of the property of stationarity of random processes and the implications of using a GP with a stationary covariance function on non-stationary time-series data. We also briefly outline the utility of the monotonic warps in modelling non-stationary data with a stationary GP.

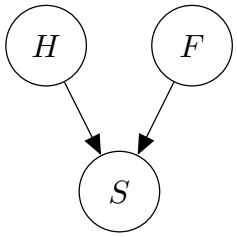
5.1 GP-GPLVM as a joint probabilistic model

Let us return to the alignment model proposed in Ch. 3. Recall that the objective in this model consists of two parts (as defined in Sec. 3.5) :

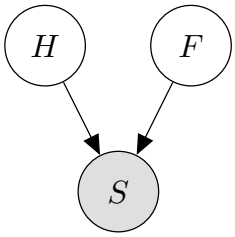
$$\begin{aligned} \mathcal{L}_1 &= \sum_j \log p(\mathbf{s}_j, \mathbf{y}_j \mid X, \theta_j, \beta_j), \\ \mathcal{L}_2 &= \log p(\mathbf{S} \mid \mathbf{Q}, \psi_h, \psi_z, \gamma) \end{aligned} \tag{5.1}$$

where $\mathcal{L}_1$ corresponds to fitting the observed data at warped inputs and $\mathcal{L}_2$ imposes a constraint on the model that favours warps such that the resulting pseudo-observations $\mathbf{S}$ at evenly sampled inputs X are aligned within groups. Consequently, the pseudo-observations $\mathbf{S}$ are regularised using both objectives.

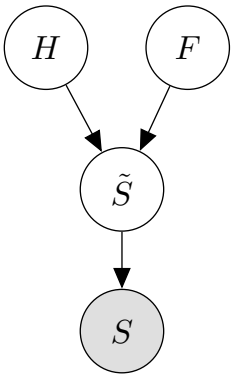
Encoding the resulting dependence structure poses the question of whether $\mathbf{S}$ should be treated as latent or as observed. An illustration of the implications of conditioning on an observed variable versus the marginalisation of a latent variable is given in Fig. 5-1. If $\mathbf{S}$ are treated as a latent variable then integrating it out breaks the dependence structure between the two parts of the model, which would correspond to fitting each observed sequence separately without imposing any alignment constraints. If, on the other hand, $\mathbf{S}$ is treated as observed, then conditioning on it keeps the dependence between the two parts of the model. This further motivates the term *pseudo-observations* used to describe $\mathbf{S}$ — it implies that the aligned sequences are treated as observed even though they are not observed per se. In practice, as explained in the previous chapters, the values of $\mathbf{S}$ are directly optimised, making it possible to treat them as observed.



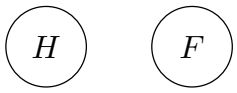
(a) Marginalising over S makes H and F independent.



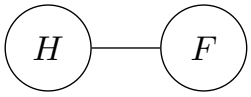
(b) Conditioning on S makes H and F dependent.



(c) Conditioning on S makes H and F dependent.



(d) H and F are (unconditionally) independent: $p(H, F) = p(H)p(F)$.



(e) In general, $p(H, F \mid S) \neq p(H \mid S)p(F \mid S)$.

Figure 5-1: Marginalising and conditioning in a DAG. The effect of marginalising and conditioning is shown in the corresponding figures below the three DAGs. Marginalising a latent variable S as shown in Fig. (a) results in the independence structure as shown in Fig. (d). Meanwhile, conditioning on an observed variable S (with or without an intermediate latent variable $\tilde{S}$, see Fig. (b) and (c)) can encode conditional structure as shown in (e). Figure adapted from [Barber, 2012].

With this in mind, the joint distribution of the observations $\mathbf{Y}$ and all the random variables (ignoring the hyperparameters and noise terms for clarity) may be written as:

$$p(\mathbf{Y}, \mathbf{S}, \mathbf{F}^{\mathrm{X}}, \mathbf{F}^{\mathrm{G}}, \mathbf{G}, \mathbf{H}, \mathbf{Q} \mid \mathbf{x}) = p(\mathbf{Y} \mid \mathbf{F}^{\mathrm{G}})\, p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}}) \\ p(\mathbf{H} \mid \mathbf{Q})\, p(\mathbf{F}^{\mathrm{X}}, \mathbf{F}^{\mathrm{G}} \mid \mathbf{G}, \mathbf{x})\, p(\mathbf{G} \mid \mathbf{x})\, p(\mathbf{Q}). \quad (5.2)$$

The terms $p(\mathbf{H}|\mathbf{Q})$, $p(\mathbf{F}^{\mathrm{X}}, \mathbf{F}^{\mathrm{G}}|\mathbf{G}, \mathbf{x})$ and $p(\mathbf{G}|\mathbf{x})$ are the GP priors defined in Sec. 3.5, and $p(\mathbf{Q}) \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ is the latent prior. The distributions $p(\mathbf{F}^{\mathrm{X}}, \mathbf{F}^{\mathrm{G}} \mid \mathbf{G}, \mathbf{x})$ and $p(\mathbf{G} \mid \mathbf{x})$ define the J observed sequences and hence they factorise fully over J. The likelihood of the observations under i.i.d. Gaussian noise with precision β_j is $p(\mathbf{Y} \mid \mathbf{F}^{\mathrm{G}}) = \prod_j p(\mathbf{y}_j \mid \mathbf{f}_j^G) = \prod_j \mathcal{N}(\mathbf{y}_j \mid \mathbf{f}_j^G, \beta_j^{-1}\mathbb{I})$.

Treating the pseudo-observations as observed variables, we are interested in interpreting the objective defined in Eq. 5.1 as a likelihood term $p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}})$ in the joint distribution. In particular, the pseudo-observations $\mathbf{S}$ should have a high likelihood under the two

parts of the model, the GPs of the sequences and the GP-LVM. Consider the likelihood $p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}})$ defined as an equal mixture of the two terms hinted in Eq. 5.1:

$$\begin{aligned} p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}}) = \frac{1}{2} \mathcal{N}\left(\mathrm{vec}(\mathbf{S}) \mid \mathrm{vec}(\mathbf{H}), \mathbb{I}_J \otimes (\gamma^{-1}\mathbb{I}_N)\right) \\ + \frac{1}{2} \mathcal{N}\left(\mathrm{vec}(\mathbf{S}) \mid \mathrm{vec}(\mathbf{F}^X), (\beta^{-1}\mathbb{I}_J) \otimes \mathbb{I}_N\right) \end{aligned} \tag{5.3}$$

where $\mathrm{vec}(\cdot)$ refers to the vectorisation of the input and β^{-1} can potentially be different for each $j = 1, \ldots, J$. Both components of this mixture explain the same observations, namely $\mathbf{S}$, and the likelihood is the highest when the means of the two components are the same.

However, as part of the alignment model, the two components are heavily constrained. The first term in Eq. 5.3 is defined in terms of a GP-LVM with priors on both the latent space and the mappings from the latent space to the observed space. Meanwhile the second term in Eq. 5.3 is constrained by the prior on the functions f but also by the observations $\mathbf{Y}$ that are fitted using these same functions. Moreover, the two terms exhibit a different independence structure over the inputs $\mathbf{S}$: the GP-LVM factorises over the features of the inputs (as discussed in Sec. 3.1.1) while the fitting of functions f factorises over the J sequences. To illustrate this, the likelihood may be rewritten as:

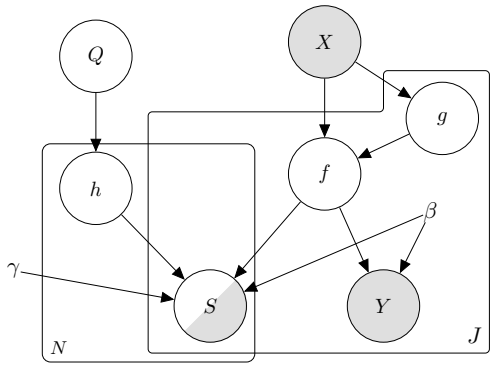
$$p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}}) = \frac{1}{2}\left(\prod_n \mathcal{N}(\mathbf{S}_{:,n} \mid \mathbf{h}_n, \gamma^{-1}\mathbb{I}_J) + \prod_j \mathcal{N}(\mathbf{S}_{j,:} \mid \mathbf{f}_j^X, \beta_j^{-1}\mathbb{I}_N)\right) \tag{5.4}$$

where $\mathbf{S}_{j,:}$ refers to the rows and $\mathbf{S}_{:,n}$ refers to the columns of $\mathbf{S}$.

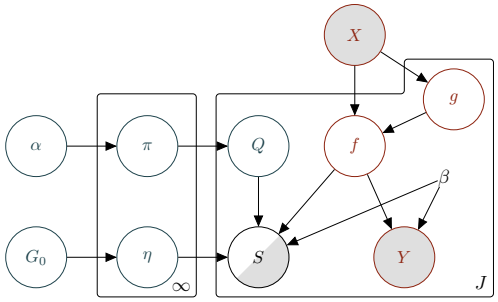
To take advantage of this factorisation when finding the MAP estimate, we note that $\log(\frac{1}{2}\, a + \frac{1}{2}\, b) \geq \frac{1}{2}\log(a) + \frac{1}{2}\log(b)$, which in this case implies that:

$$\begin{aligned} \log p(\mathbf{S} \mid \mathbf{H}, \mathbf{F}^{\mathrm{X}}) &= \log\left(\frac{1}{2}\prod_n \mathcal{N}(\mathbf{S}_{:,n} \mid \mathbf{h}_n, \gamma^{-1}\mathbb{I}_J) + \frac{1}{2}\prod_j \mathcal{N}(\mathbf{S}_{j,:} \mid \mathbf{f}_j^X, \beta_j^{-1}\mathbb{I}_N)\right) \\ &\geq \frac{1}{2}\left(\sum_n \log \mathcal{N}(\mathbf{S}_{:,n} \mid \mathbf{h}_n, \gamma^{-1}\mathbb{I}_J) + \sum_j \log \mathcal{N}(\mathbf{S}_{j,:} \mid \mathbf{f}_j^X, \beta_j^{-1}\mathbb{I}_N)\right) \end{aligned} \tag{5.5}$$

Using this likelihood, we can integrate out $\mathbf{F}^{\mathrm{X}}, \mathbf{F}^{\mathrm{G}}$ and $\mathbf{H}$ from Eq. 5.2 to obtain $p(\mathbf{Y}, \mathbf{S} \mid \mathbf{G})$, which is a sum of two terms due to the mixture in the definition of the likelihood in Eq. 5.5. Learning in this model then corresponds exactly to the learning procedure defined in Sec. 3.5. In particular, we optimise $\log p(\mathbf{Y}, \mathbf{S} \mid \mathbf{G})$ which includes



(a) Illustration of a DAG for the GP-GPLVM alignment model.



(b) Illustration of a DAG for the GP-DPMM alignment model.

Figure 5-2: Graphical models for the GP-GPLVM and the GP-DPMM alignment models. The observed quantities are shown in grey.

the two terms derived in Eq. 5.5 along with the remaining terms that correspond to the likelihood of the observations $\mathbf{Y}$ and the (hyper) priors w.r.t. the pseudo observations $\mathbf{S}$, the latent variables $\mathbf{Q}$ and the hyperparameters of the model to obtain the MAP estimates. The resulting lower bound on the marginal log-likelihood now corresponds directly to the objective in the original alignment model, which is maximised when the following objectives are simultaneously optimised: (1) the observed data $\{\mathbf{y}_j\}$ is fitted well by the corresponding GPs f_j at the warped locations, (2) the pseudo-observations $\{\mathbf{s}_j\}$ are fitted well by the corresponding GPs f_j at the fixed sampling locations, (3) the pseudo-observations are such that they exhibit a simple structure in the low-dimensional latent space.

5.1.1 Graphical models and observed residuals

Given the joint distribution as defined in Eq. 5.2, we may consider a corresponding graphical model, shown in Fig. 5-2a (in this discussion we consider the GP-GPLVM alignment model though similar arguments hold for the GP-DPMM model; the corresponding graphical model is given in Fig. 5-2b). Here the inputs $\mathbf{x}$ and the outputs $\mathbf{Y}$ are observed (and are shown as shaded nodes) while the pseudo-observations $\mathbf{S}$ are not observed but they are treated as observations, hence they are partially-shaded in the graphical model.

One of the difficulties with such a graphical model of alignments stems from the noise associated with the pseudo-observations. Specifically, there is a noise term associated with the fitting of each of the functions $\{f_j\}$ as well as a noise term associated with the

fitting of the GP-LVM to the pseudo-observations. This makes it impossible to reason about the noise associated with each of the aligned sequences and the generative process defined by such a model remains ambiguous. This is illustrated in Fig. 5-2a where the pseudo-observations $\mathbf{S}$ are dependent on the two noise terms, γ and β_j.

To further explore this issue, we discuss an alternative presentation of the model inspired by the hierarchical models in [Park and Choi, 2010]. In this formulation, we introduce a noisy residual term which is defined as the difference between the predictions of the two parts of our model (the fitting of the $\{f_j\}$ and the GP-LVM), and is chosen to be zero by definition.

Let us define the inputs to the GP-LVM as an observed residual term $\mathbf{R} = \mathbf{0}^{N\times J}$. As in the original model, there exists a latent variable $\mathbf{Q} \in \mathbb{R}^{N\times Q}$, with $Q < N$, such that $\mathbf{Q}$ is related to the predictions $\mathbf{R}$ through smooth probabilistic mappings $h_n, n = 1, \ldots, N$, as follows:

$$\begin{aligned} \mathbf{r}_n &= h_n(\mathbf{Q}) + \epsilon_h, \\ h_n \sim \mathcal{GP}(\,\cdot\, \mid 0, \mathbf{K}_{\psi_q}), &\quad \epsilon_h \sim \mathcal{N}(\mathbf{0}, \gamma^{-1}\mathbb{I}_N). \end{aligned} \tag{5.6}$$

As previously, there exist N independent functions h_n, one for each dimension of the input data, all of which are conditioned on $\mathbf{Q}$ and share the same covariance and mean functions.

The posterior processes associated with each function f_j are independent and defined as $f_j^* \mid \mathbf{x}, \mathbf{g}_j \sim \mathcal{GP}(\tilde{m}_{\theta_j}, \tilde{k}_{\theta_j})$ where the mean and the covariance functions are as follows:

$$\begin{aligned} \tilde{m}_{\theta_j} &= k_{\theta_j}(\mathbf{x}, \mathbf{g}_j)^T (k_{\theta_j}(\mathbf{g}_j, \mathbf{g}_j))^{-1} \mathbf{y}_j, \\ \tilde{k}_{\theta_j} &= k_{\theta_j}(\mathbf{x}, \mathbf{x}) - k_{\theta_j}(\mathbf{x}, \mathbf{g}_j)^T (k_{\theta_j}(\mathbf{g}_j, \mathbf{g}_j))^{-1} k_{\theta_j}(\mathbf{x}, \mathbf{g}_j) \end{aligned} \tag{5.7}$$

with GP kernel parameters θ_j. We would like to use the predictive mean functions of the $\{f_j\}$ as constant mean functions of the GP priors in the GP-LVM similarly to the hierarchical GP models. See Fig. 5-3 for the corresponding graphical model.

Each function h_n (with a corresponding realisation $\mathbf{h}_n$) is now modelled by an independent GP with a non-zero mean that corresponds to the realisations of $\{f_j\}$ at inputs $\mathbf{x}$ (denoted by $\mathbf{f}_n^*$), and a covariance function k_{ψ_q} that depends on the latent inputs $\mathbf{Q}$ and is the same for all h_n, $n = 1, \ldots, N$:

$$h_n \mid \mathbf{f}_n^*, \mathbf{Q} \sim \mathcal{N}(\mathbf{f}_n^*, \mathbf{K}_{\psi_q}). \tag{5.8}$$

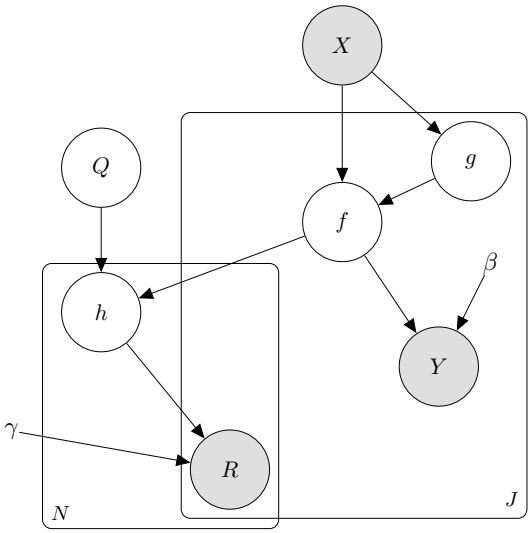


Figure 5-3: Illustration of a DAG for the GP-GPLVM alignment model using residuals. The observed quantities are shown in grey.

While h_n are non-zero mean GPs, in practice, we consider the corresponding zero-mean GPs, i.e. $\eta_n := \mathbf{h}_n - \mathbf{f}_n^*$ is $\eta_n \,|\, \mathbf{f}_n^*, \mathbf{Q} \sim \mathcal{N}(0, \mathbf{K}_{\psi_q})$, and $\mathbf{f}_n^*$ are independent of $\mathbf{Q}$. Then the latent space $\mathbf{Q}$ is related to the noisy observations $\mathbf{R}$ as $r_{j,n} = h_n(\mathbf{q}_j) + \epsilon_\gamma$ or equivalently, $r_{j,n} - \mathbf{f}_n^*(\mathbf{x}_j) = \eta_n(\mathbf{q}_j) + \epsilon_\gamma$, where ϵ_γ is Gaussian noise, $p(\epsilon_\gamma) = \mathcal{N}(0, \gamma^{-1})$.

This interpretation includes some notable difference from the previous formulation. The explicit definition of the residual term $\mathbf{R}$ as an observed variable makes it possible to avoid defining the pseudo-observations $\mathbf{S}$. Furthermore, unlike the pseudo-observations, the residuals depend only on the noise term associated with the mappings of the GP-LVM, ϵ_γ (and not on the observation noise parameter ϵ_β). In this case, the aligned sequences correspond to the predictive means of $\{f_j\}$ evaluated at the fixed input locations $\mathbf{x}$.

In the next section we examine some of the practical implications of the aforementioned modelling assumptions, in particular, the definition of the aligned sequences as the means of the GPs as an alternative to the directly optimised pseudo-observations. At the same time, we study the two noise terms present in the alignment model, and start a broader discussion on the meaning and the value of the uncertainty propagation in the model of alignments.

5.1.2 Shared noise terms

The model of alignments as introduced in Ch. 3 and further discussed in Sec. 5.1 has two noise terms, one associated with the fitting of the GPs to the data (i.e. the fitting

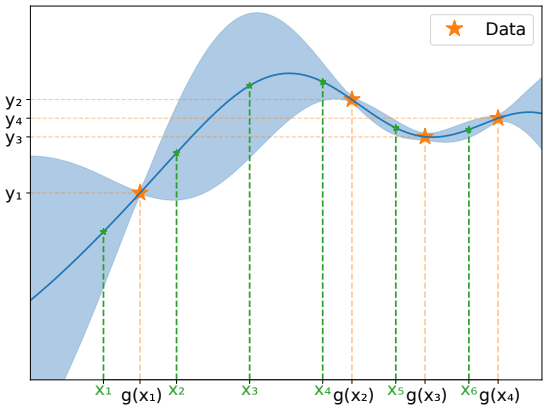


Figure 5-4: Illustration of uncertainty at fixed input locations $\mathbf{x}$ given a composition $f(g(\mathbf{x}))$ of functions (or, equivalently, a function f fitted to warped inputs $g(\mathbf{x})$) fitted to the observations $\mathbf{y}$. The blue curve is the mean of the GP fitted to 4 data points (orange). The blue region shows 2 SD confidence region.

of functions $\{f_j\}$), parameterised by β_j, and a separate noise term associated with the inputs to the GP-LVM, parameterised by γ. The latter one refers only to the noise on the pseudo-observations $\mathbf{S}$ (or the residual term $\mathbf{R}$ in the alternative formulation of the model) as the GP-LVM does not observe the data $\mathbf{Y}$ directly. Therefore, treating this term as observational noise is somewhat misleading. Alternatively, we may note that the two noise terms are intrinsically related as part of a joint model, suggesting that they should be interpreted jointly. Furthermore, this raises the question of what the uncertainty associated with the alignment procedure is. While the noise terms ϵ_β explain the (irreducible) uncertainty due to the observation/measurement error associated with the observed data $\mathbf{Y}$, we might also be interested in the model uncertainty, *i.e.* the uncertainty that arises from the fact that there are multiple different explanations for the observed data all of which are consistent with our prior beliefs about the characteristics of the warps and the observations. In the case of alignments, one might, for instance, be interested in the quality of the warps (*i.e.* the plausibility of the specific warps) and the confidence in the fits (*i.e.* the fitting of the $\{f_j\}$ to the observations) given the set of observed data and the prior beliefs. To make progress on these questions, we consider how the uncertainty could be propagated between the two parts of the model.

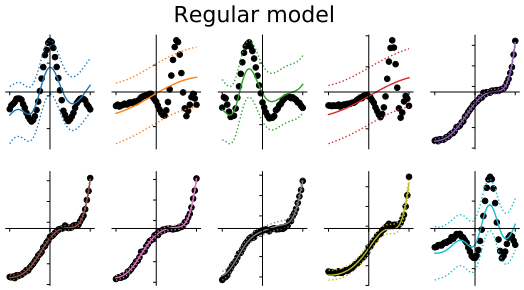
Let us assume that instead of the directly optimised pseudo-observations, the GP-LVM observes the means of the predictive posteriors of the functions $\{f_j\}$ at $\mathbf{x}$ that model the observed sequences. The covariances of the corresponding predictive posteriors take into account the uncertainty that arises from (1) the observational/measurement noise and (2) the uncertainty related to the non-uniform sampling of the latent functions

$\{f_j\}$ introduced by the warping of the inputs. For example, part (2) implies that the covariance between any two elements in f_j depends on the distances between the inputs $\mathbf{x}$ and $g_j(\mathbf{x})$'s (*i.e.* the error bars for f_j will vary depending on how close the positions in the fixed sampling rate $\mathbf{x}$ are to the positions $g_j(\mathbf{x})$ at which we have observed $\mathbf{y}_j$). This is illustrated in Fig. 5-4. Consequently, the predictive covariances may be used to propagate the uncertainty that is associated with the compositions $f_j(g_j(\mathbf{x}))$.

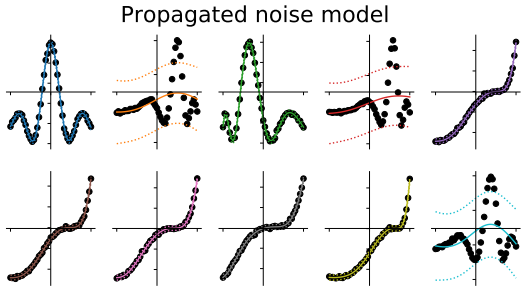
In the GP-GPLVM alignment model, the covariance of the GP mappings in the GP-LVM is defined as $\mathbf{K}_{\psi_q}$ where $\mathbf{K}_{\psi_q} = k(\mathbf{q}, \mathbf{q}')$ encodes the relations amongst the latent inputs, as defined in Sec. 3.4.1. In the original model, the observations $\mathbf{S}$ are considered to be noisy, therefore, the prior on these noisy observations becomes $\mathbf{K}_{\psi_h} + \gamma\mathbb{I}_N$. However, the uncertainty about the aligned sequences is encoded in the predictive posteriors of $\{f_j\}$ and thus we consider replacing the noise term $\gamma\mathbb{I}_N$ with a more informative term that stems from the modelling of the observations.

Let us define $\mathbf{\Sigma}_n$ as the matrix of the standard deviations of the predictive posterior associated with each term $\{f_{j,n}\}$ at fixed inputs $\mathbf{x}$ (these correspond to the standard deviations at the fixed (green) locations in Fig. 5-4). This determines the uncertainty associated with the fitting of the $f_j(g_j(\mathbf{x}))$ to the observed data $\mathbf{y}_j$ for all $j = 1, \ldots, J$. For each of the N (independent and identical) GP priors in the GP-LVM, the diagonal of $\mathbf{\Sigma}_n$ is the n^{th} column of the matrix that contains the standard deviations. Therefore, each of the N GPs in the GP-LVM, h_n, now has a covariance matrix $\mathbf{K}_n = \mathbf{K}_{\psi_h} + \mathbf{\Sigma}_n$. The term $\mathbf{\Sigma}_n$ does not depend on the latent variables $\mathbf{Q}$, so it remains constant in the GP-LVM. However, this implies a different noise term for each feature in the GP-LVM (*i.e.* $\mathbf{\Sigma}_n$ is different for each feature). This leads to N different GPs in the GP-LVM and is not efficient in comparison to the original formulation.

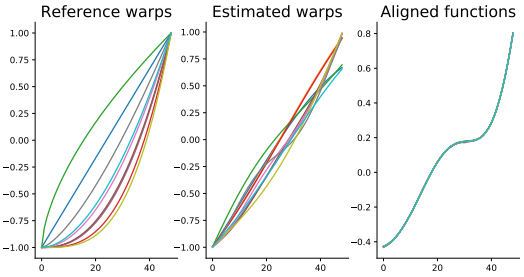
Moreover, in this setting the $\mathbf{\Sigma}_n$ is only known for the J observed sequences. Therefore, from the perspective of the GP-LVM, if we were to sample a new location $\mathbf{q}^*$ from the latent space of the GP-LVM, we would need an estimate of $\mathbf{\Sigma}_n$ at $\mathbf{q}^*$ to be able to calculate the predictive mean and the predictive variance at this point. In particular, instead of using independent Gaussian noise, we are considering a model where the noise depends on $\mathbf{Q}$, i.e. input-dependent noise. One way to estimate this term is using GP interpolation (noiseless GP regression) on the observed noise $\mathbf{\Sigma}_n$. This is equivalent to introducing a new process $\tilde{h}$ which models the predictive variance of the functions f_j using the latent inputs $\mathbf{Q}$ from the original process h (which in turn models the predictive mean of the functions f_j). This model is closely related to the heteroscedastic GP regression as defined by [Kersting et al., 2007].



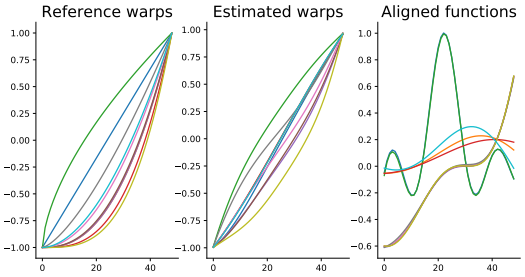
(a) Data fits (regular model).



(b) Data fits (propagated noise model).



(c) Warps and alignments (regular model).



(d) Warps and alignments (propagated noise model).

Figure 5-5: (a, b): The GP fits to the same data for the regular and propagated noise models; (c, d): estimated warps and aligned functions for the regular and propagated noise models.

The advantage of sharing the predictive variance of f_j with the GP-LVM is that it allows us to take into account the quality of the data fits while aligning the sequences. Intuitively, the predictive means of the functions $\{f_j\}$ can look perfectly aligned even if the true underlying sequences are different if (1) some of the functions $\{f_j\}$ do not fit the data well (ignoring the evidence that the underlying sequences cannot be aligned), or (2) some of the observed sequences are estimated to be heavily warped (leading to high uncertainty in some parts of the domain, as illustrated in Fig. 5-4). Both (1) and (2) account for an increase in predictive variance of the $\{f_j\}$ but the GP-LVM cannot tell if the alignment is plausible unless we propagate this information to the GP-LVM.

This effect is especially noticeable for the heavily-warped observed data, which is illustrated in Fig. 5-5. The panels (a) and (b) show the observed sequences, and the predictive means of $\{f_j\}$ for the regular and the propagated noise models. Panels (c) and (d) show the estimated warps and the aligned sequences. The data fits are poor for both cases, however, for the propagated noise model that does not lead to the sequences being over-aligned, rather on the contrary, it aligns the sequences with good fits, and ignores the poorly fitted ones (the estimated warps for these cases are close to identity

functions). The regular model aligns the predictive means of the $\{f_j\}$ regardless of the quality of the fitting.

While the predictive variance implicitly captures some of the uncertainty about the warps, the model of the warps is not itself probabilistic and thus we cannot reason about the uncertainty related to the possible warps. In particular, since we estimate f_j and g_j simultaneously, it is often possible to fix f_j to the same function f for all $j = 1, \ldots, J$, and find corresponding g_j, such that $f_j \circ g_j$ fits the observed sequence $\mathbf{y}_j$ well. To address this issue, the warps need to be defined as a stochastic process that is constrained to be monotonic.

We will further explore the research on monotonic random processes in Ch. 6, and we will return to the discussion of uncertainty propagation in Ch. 7 where we further explore the uncertainty related to the warpings and the issue of compositional uncertainty in hierarchical models.

5.2 Matrix distributions, multi-output GPs and multi-task learning

The main objective in the alignment model is to impose correlations between the (unknown) aligned sequences. In the proposed models of alignments this is achieved by placing a GP-LVM or a DPMM as a prior on the set of aligned sequences. This construction bears a similarity to the constructions imposed by matrix distributions and those used in multi-output learning where the goal is to find confounding factors among the sequences that correlate the rows of the data matrix, as well as the columns (*i.e.* each output dimension is modelled by a GP, and our goal is to correlate the different dimensions).

Matrix normal distribution One way to introduce these correlations is to consider a matrix normal distribution [Dutilleul, 1999], defined as:

$$\begin{aligned} \mathbf{Y} &\sim \mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V}), \\ p(\mathbf{Y}|\mathbf{M}, \mathbf{U}, \mathbf{V}) &= \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\mathbf{V}^{-1}(\mathbf{Y}-\mathbf{M})^T\mathbf{U}^{-1}(\mathbf{Y}-\mathbf{M})\right]\right)}{(2\pi)^{np/2}|\mathbf{V}|^{n/2}|\mathbf{U}|^{p/2}} \end{aligned} \tag{5.9}$$

where $\mathbf{Y} \in \mathbb{R}^{J\times N}$ is a matrix-values random variable, $\mathbf{M} \in \mathbb{R}^{J\times N}$ is the location parameter, and $\mathbf{U} \in \mathbb{R}^{J\times J}$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$ are scale parameters (positive-definite

matrices). Such distribution is equivalent to a multivariate normal distribution if and only if the covariance structure can be captured using a Kronecker product:

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{U} \otimes \mathbf{V}). \tag{5.10}$$

Therefore, $\mathbf{M}$ is the mean of the observations $\mathbf{Y}$ while $\mathbf{U}$ and $\mathbf{V}$ capture the correlations among the J sequences and the N features. This framework naturally generalises to models based on GPs and is typically termed multi-output GPs (MOGPs) [Bilionis et al., 2013, Álvarez et al., 2012]. Some typical scenarios for multi-output regression involve observing data from multiple tasks and transferring the knowledge between tasks [Bonilla et al., 2008], observing multiple trials of the same experiment and inferring missing data in some of the trials using the information from the other trials [Álvarez et al., 2012], and multi-fidelity learning using cheap measurements as a proxy for expensive ones [Liu et al., 2018]. As more concrete examples, consider the following: given data $\mathbf{Y} \in \mathbb{R}^{J \times N}$:

1. [Bonilla et al., 2008] place a GP prior over each sequence (task) and include a free-form matrix C that captures the covariances among the tasks,

2. [Stegle et al., 2011] use dimensionality reduction (for example, a GP-LVM) to account for the confounding covariances between the samples.

5.2.1 Alignment using MOGPs

It is then natural to ask if alignment can be performed using the MOGPs. Such a model would be fully generative allowing us to sample jointly from the latent space and GPs that explain the data, avoiding the issues discussed in Sec. 5.1.

Alignment in single group Given some observed unaligned data $\mathbf{Y} \in \mathbb{R}^{J \times N}$, assume that there is only one underlying function f for all the observed sequences $\{\mathbf{y}_j\}$. Then all of the sequences should be modelled using the same function f, which is equivalent to fitting a GP to a data set that contains all the inputs $\mathbf{Y}$ without differentiating which $\mathbf{y}_j$ comes from which sequence (*i.e.* every input y_{jn} is correlated with every other). In general, we know that the data set $\{\mathbf{x}_j, \mathbf{y}_j\}$ for all J cannot be modelled using one f due to the different warps in each $\{\mathbf{y}_j\}$. Therefore, we introduce the warping functions $\{g_j\}$ for each sequence such that $\mathbf{y}_j = f(g_j(\mathbf{x})) + \epsilon_\beta$, where the functions g_j are constrained to be monotonic as in the previous chapters using the parameterisation defined in Eq. 3.9. Since we allow for this extra flexibility in modelling the data, *i.e.* we model $\mathbf{y}_j = f(g_j(\mathbf{x})) + \epsilon_\beta$ for all J rather than $\mathbf{y}_j = f(\mathbf{x}) + \epsilon_\beta$ for all J, the warping functions

can be used to alter the data set in a way that all the data $\mathbf{Y}$ for this joint data set (with $J \times N$ data points) can be explained with a single GP with a stationary kernel. Using the terms from the geostatistics literature, this turns the isotopic data (where each output has the same set of inputs) into heterotopic data (where each output may be associated with a different set of inputs) [Álvarez et al., 2012]. Assuming $f \sim \mathcal{GP}(0, k_\theta(\cdot,\cdot))$ and every observed sequence is a noisy evaluation of f at the corresponding warped inputs, the marginal likelihood is as follows:

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}\left(\mathbf{0}^{J\times N}, \underbrace{\begin{bmatrix} \mathbf{K}_\theta\left(g_1(\mathbf{x}), g_1(\mathbf{x})\right) + \beta^2\mathbb{I} & \dots & \mathbf{K}_\theta(g_1(\mathbf{x}), g_J(\mathbf{x})) \\ \vdots & \ddots & \vdots \\ \mathbf{K}_\theta(g_J(\mathbf{x}), g_1(\mathbf{x})) & \dots & \mathbf{K}_\theta(g_J(\mathbf{x}), g_J(\mathbf{x})) + \beta^2\mathbb{I}) \end{bmatrix}}_{=:K_\theta^G}\right) \tag{5.11}$$

where the ij^{th} (with $i \neq j$) element in the covariance matrix is $\mathbf{K}_\theta(g_i(\mathbf{x}), g_j(\mathbf{x}))$, while the diagonal elements include the noise term. Depending on the data set, the noise parameter β can be defined to be different for each of the J sequences. Training the model involves finding the MAP estimates of the parameters θ of the (stationary) kernel $\mathbf{K}_\theta(\cdot,\cdot)$. We note that as an alternative to MAP estimates, the model can be trained using variational inference [Álvarez et al., 2010]. The aligned version of the sequences can be recovered by evaluating the predictive posterior of f at fixed evenly spaced inputs $\mathbf{x}$. Fig. 5-6 shows the alignment results for this MOGP model. More specifically, the MOGP model is able to align the three toy sequences that were generated by evaluating the same function ($\text{sinc}(\cdot)$) at differently warped inputs.

Alignment in multiple groups Now assume that there are multiple underlying functions $\{f_k\}$ that generated the observations $\mathbf{Y}$ and we do not know the group assignments for each of the sequences $\mathbf{y}_j$. If, for example, PCA or a GMM allowed us to assign these sequences into groups correctly, then we could use the above argument (as described for a single group) to align sequences within multiple groups (either by fitting K separate GPs for each of the K groups, or by modelling all the sequences jointly but explicitly identifying which sequences should correlate with which other sequences).

Unfortunately, as previously discussed (see, for example, Fig. 4-6), PCA or a GMM do not provide reliable clustering as the clustering needs to be based on an invariance

to temporal warps (*i.e.* the clusters are such that the sequences in each cluster can be temporally aligned). Therefore, we need a procedure that allows us to automatically find the smallest number of groups such that the sequences in each group share the underlying function f_k; this directly corresponds to defining an alignment objective as we did in the previous chapters.

Confounders As in the standard MOGP formulation (see Eq. 5.10), we can utilise the Kronecker product structure of the covariance matrix to introduce correlations between the sequences, resulting in:

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}\left(\mathbf{0}^{J\times N}, \begin{bmatrix} c_{11}\mathbf{K}_\theta\left(g_1(\mathbf{x}), g_1(\mathbf{x})\right) + \beta^2\mathbb{I} & \dots & c_{1J}\mathbf{K}_\theta(g_1(\mathbf{x}), g_J(\mathbf{x})) \\ \vdots & \ddots & \vdots \\ c_{J1}\mathbf{K}_\theta(g_J(\mathbf{x}), g_1(\mathbf{x})) & \dots & c_{JJ}\mathbf{K}_\theta(g_J(\mathbf{x}), g_J(\mathbf{x})) + \beta^2\mathbb{I}) \end{bmatrix}\right) \tag{5.12}$$

where $\mathbf{C} \in \mathbb{R}^{J\times J}$ is symmetric positive definite. Matrix $\mathbf{C}$ is the matrix of correlations (or similarities) between the latent functions for different observations and it encodes the group assignments; for example, if two latent functions are highly correlated, they are assigned to the same group. For the alignment task we expect this matrix to be approximately binary, masking the spatial covariances $\mathbf{K}_\theta(g_i(\mathbf{x}), g_j(\mathbf{x}))$ in a way that only the values of the sequences of the same group (*i.e.* those that share the same latent functions) are correlated.

The correlation matrix $\mathbf{C}$ could be optimised directly while ensuring that the matrix remains positive definite (for example, formulating $\mathbf{C}$ in terms of a triangular matrix formed using the Cholesky decomposition). Imposing correlations between sequences within the groups leads to a higher likelihood when compared to the case with no correlations between sequences, which corresponds to an objective to align sequences into few groups.

Correlations using a latent space Similarly to the model proposed by [Stegle et al., 2011], and in a vein of the work in Ch. 3, the correlations may be introduced using dimensionality reduction. For instance, the correlations $\mathbf{C}$ can be learnt using a formulation similar to a GP-LVM by creating an explicit latent space representation. In particular, let us define $\mathbf{C}$ such that $c_{ij} = k_\theta(\mathbf{q}_i, \mathbf{q}_j)$ where $\mathbf{q}_i \in \mathbb{R}^Q$ are the latent place locations and $k_\theta(\cdot,\cdot)$ is the GP kernel in the GP-LVM. Then the term $k_\theta(\mathbf{q}_i, \mathbf{q}_j)$ tells us if the sequences $\mathbf{y}_i$ and $\mathbf{y}_j$ should be correlated; if $k_\theta(\mathbf{q}_i, \mathbf{q}_j) = 0$ then the effect of the

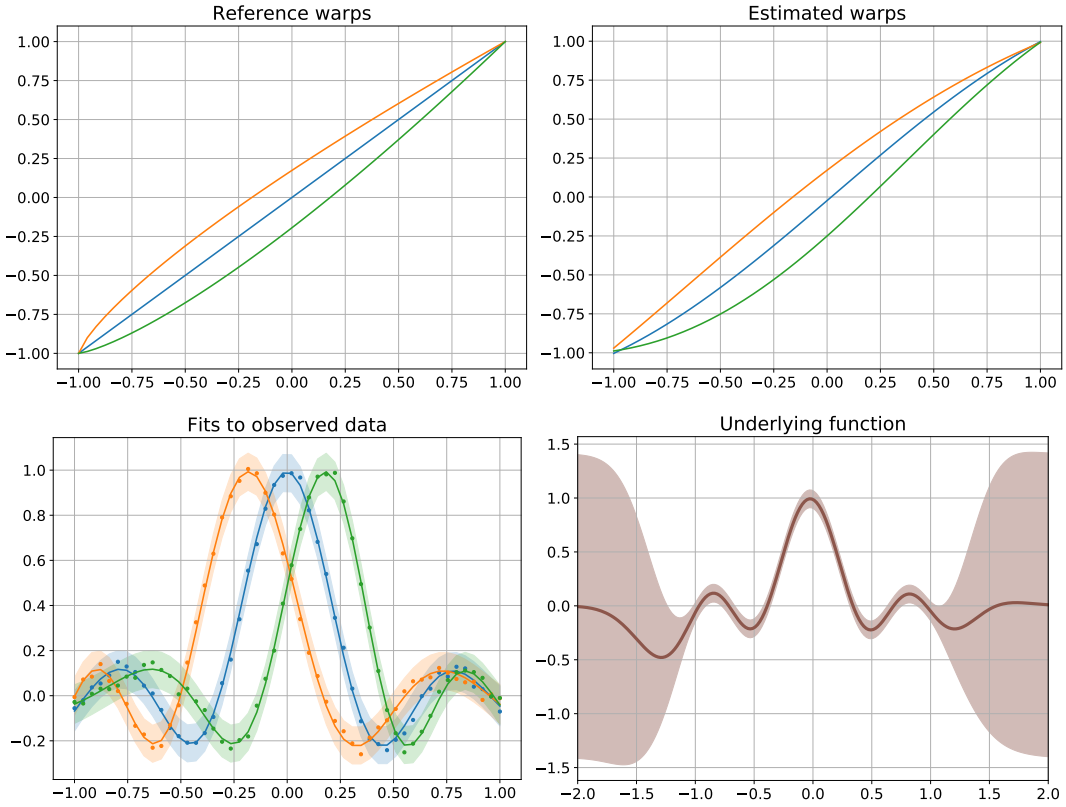


Figure 5-6: Alignment of toy sequences within one group using MOGP. The true warps and the estimated warps are shown in the top row. The bottom row shows the fit to the observed (noisy) data (left) and the predictive mean and the region of 2 SD away from the mean of the underlying latent function at fixed inputs $\mathbf{x}$ (right).

term $\mathbf{K}_\theta(g_i(\mathbf{x}), g_j(\mathbf{x}))$ is ignored.

Consider the GP-LVM with a SE kernel and a latent space prior $p(\mathbf{Q}) \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$, and fit a MOGP model (as defined in Eq. 5.12). As previously, we can find the MAP estimates to the parameters of this model, including the parameters of the two respective kernels and the latent space locations $\mathbf{Q}$. Fig. 5-7 illustrates alignment when the data comes from two distinct groups.

One of the caveats of this formulation is that the kernel $\mathbf{K}_\theta$ is the same for all of the sequences. This is a reasonable choice when there is only one group of sequences as that corresponds to an assumption that there is only one underlying function f that generated all of the observed sequences. However, given multiple groups of sequences, the characteristics of the sequences within each group might be different as they were generated using a different underlying function f_k for each of the K groups. The artifacts due to using the same kernel can be seen in Fig. 5-7 (e), where the lengthscale of the SE

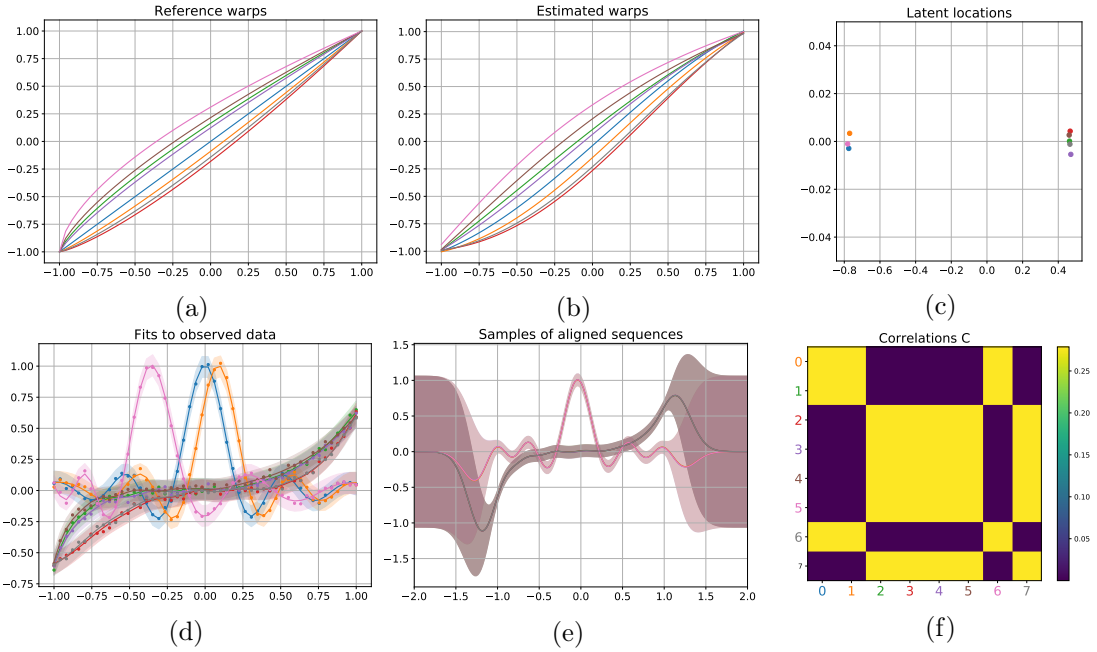


Figure 5-7: Alignment of toy sequences within multiple groups using MOGP. The latent locations $\mathbf{q}_i$ for the sequences within the two groups are separated in two clusters (Fig. (c)). This is equivalently evident from the structure of the resulting correlation matrix $\mathbf{C}$.

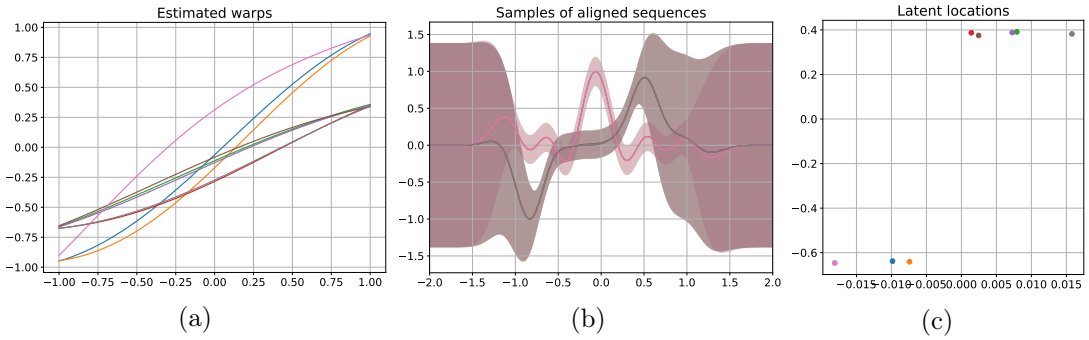


Figure 5-8: Alignment of toy sequences within multiple groups using MOGP with flexible warps. As the ends of the warping functions are not fixed, they are optimised to values such that the input space is contracted (from $[-1, 1]$ to approximately $[-0.6, 0.3]$, see Fig. (a)), allowing the model to hallucinate the shape for the cube function that best matches the shape (and lengthscale) of the sinc function (see Fig. (b)).

kernel $\mathbf{K}_\theta$ is estimated to be such that the sequences with the shortest lengthscale (in this case, the sinc$(\cdot)$) gets fitted well. Consequently, the lengthscale is too short for the sequences with the longer true lengthscale (the cubes). Furthermore, note that in this example, the ends of the warping functions are fixed to $[-1, 1]$. Allowing the ends of the

warping functions to vary, in this case, results in the warpings being used to contract the input space to encourage the two groups to look more alike, essentially making the length-scale shorter for the cube sequences and extrapolating the cube sequences in an arbitrary way in the unused (after contraction) part of the latent space (see Fig. 5-8). Reviewing the existing work and exploring new ways to use different GP covariance functions for each of the groups is left for future work.

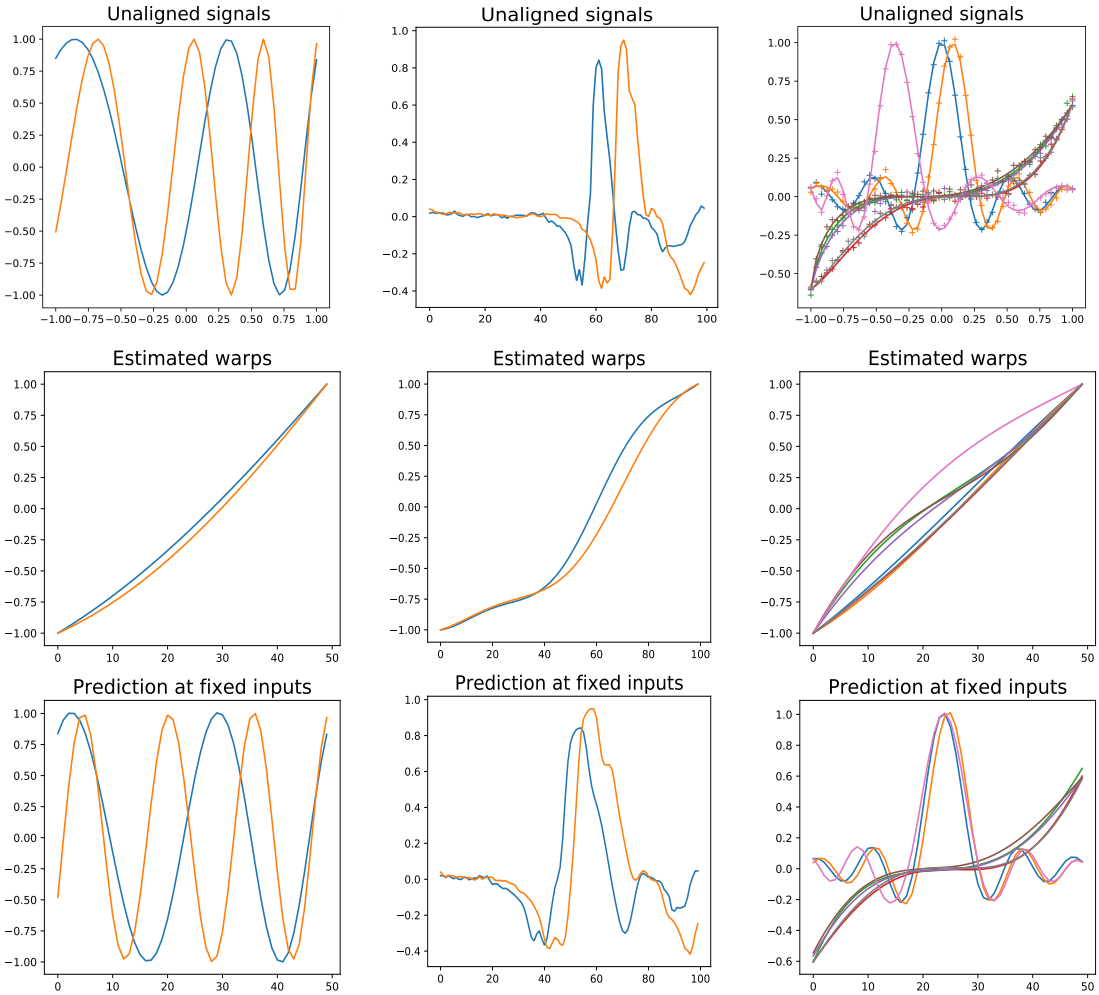
Computational complexity Due to the construction of the covariance matrix in Eq. 5.12, the computational complexity of a standard MOGP is $\mathcal{O}(J^3N^3)$, or $\mathcal{O}(J^3N^3 + J^3)$ if the correlations in $\mathbf{C}$ are modelled using a GP-LVM. This makes the use of a naive implementation of MOGP infeasible beyond simple toy problems. Some of the existing inference methods for MOGP [Stegle et al., 2011] reduce the complexity to $\mathcal{O}(J^3 + N^3)$, however, the correlation matrix defined in Eq. 5.12 does not have the same Kronecker product structure as the MOGP formulation defined in [Stegle et al., 2011]. Consequently, further work is needed to determine if it possible to bound the complexity of the proposed approach similarly to [Stegle et al., 2011] or to reduce it using sparse approximations. Recall that the complexity of the alignment approach proposed in Ch. 3 is $\mathcal{O}(JN^3 + J^3)$ (without using sparse approximations).

Issues with optimisation While we have not tested this alignment model thoroughly, our initial experimentation (using Tensorflow's own optimisation routines such as Adam [Kingma and Ba, 2014]) suggests that a model based on MOGP is sensitive to both the initialisation of the parameters and the latent space as well as the hyperpriors. Further investigation into the properties of this model and some alternative approaches to optimisation (for example, natural gradients) is needed to compare how this model compares to the approaches proposed in the previous chapters.

5.3 Stationarity

We end this chapter with a short discussion of stationarity (also called time-invariance) as it is one of the pivotal themes of this thesis. Informally, a stochastic process is stationary if its statistical properties do not change as the input space (typically referring to time) is shifted. As defined in Sec. 2.1.1, a covariance function $k(\mathbf{x}, \mathbf{x}')$ (and the resulting GP) is stationary if it is a function of $||\mathbf{x} - \mathbf{x}'||_2$ only (and not the actual values of the inputs $\mathbf{x}$ and $\mathbf{x}'$). As we have seen in the previous section (specifically, Fig. 5-7 and 5-8),

modelling non-stationary data with a stationary process leads to artifacts and sabotages the quality of the data fits.



(a) (b) (c)

Figure 5-9: Modelling non-stationary data with a composition of a monotonic warping function g_j and a GP f_j with a stationary kernel for three data sets. The top row shows the input data for for three different cases: chirps, heartbeats and warped sinc and cube functions. The middle row shows the warps g_j estimated for each of the sequences. The bottom row contains the predictive means of the fitted GPs evaluated at fixed evenly spaced inputs.

In the context of alignment, the observed sequences $\{\mathbf{y}_j\}$ are non-stationary (as they are warped by monotonic warps g_j), and we model them as compositions of these monotonic warps and stationary underlying functions f so that $\mathbf{y}_j = f\left(g_j(\mathbf{x})\right) + \epsilon$. In this section we explore the applications of such composite models (monotonic warp + stationary

GP) for modelling non-stationary observations in general, without having a goal of aligning them in the first place. This puts our alignment model into a broader context of accounting for the non-stationary behaviour in time-series data which is an active area of research (see, for example, [Snoek et al., 2014, Raket et al., 2016]).

Fig. 5-9 shows the results on two toy data sets and a data set of two heartbeat sequences. The first data set (see Fig. 5-9a) is a chirp function defined as $f(x) = \sin(2\pi \exp(x + l))$, and is non-stationary by definition. The second data set (see Fig. 5-9b) contains two sequences of a heartbeat as previously defined in Sec. 3.7.4. The third data set (see Fig. 5-9c) is generated by warping the inputs to a sinc and a cube function, making the resulting sequences non-stationary (this data set is identical to the one used in Fig. 5-7). In these experiments each of the sequences $\mathbf{y}_j$ are fitted independently using a composition of a GP f_j with a SE kernel and a monotonic warping function g_j and no alignment objective is used.

As expected, the chirp data (Fig. 5-9a) is broken down into a sine wave and a monotonic warp. In the case of the heartbeats (Fig. 5-9b), the data appears to contain two modes – the flat noisy region for $x < 40$ and a beat for $x > 40$; the warping function is thus used to compress the flat regions so that the resulting sequences can be fitted using a GP with a SE kernel. Finally, in the case of the sinc and cube functions (Fig. 5-9c), due to the way the data is generated (by warping the inputs to these two stationary functions), the transformation of the inputs also results in the sequences looking more aligned.

This suggests that the proposed composite construction with the monotonic parameterisation of the warps can be used to model non-stationary time-series data similar to the data shown in Fig. 5-9 without the need to transform the data as a pre-processing step to GP regression. Finally, a parallel can be drawn with the sparse GP methods in which the inducing locations are directly optimised, and the GP is fitted to the pairs of inducing locations and inducing points. For example, as noted by [Snelson and Ghahramani, 2006], the sparse pseudo-input GPs in the Fully Independent Training Conditional (FITC) can be interpreted as a GP with a non-stationary covariance function parameterised using the pseudo-inputs.

6

Monotonic GP flow

We now turn our attention to the monotonic warping function. Thus far we have only considered point estimates for the warps, ignoring the fact that there may exist different explanations (in our case formulated as a composition) for the observed data. Capturing this uncertainty requires formulating the warping functions as trajectories from a monotonic random process. In this chapter we review the existing approaches to monotonic regression and propose a novel formulation based on the recent work on GP flows [Hegde et al., 2019].

6.1 Overview

Monotonic regression is a task of inferring the relationship between a dependent variable y and an independent variable x when it is known that the relationship $y = f(x)$ is monotonic, *i.e.* the derivative $f'(x)$ is either non-negative or non-positive in the entire domain of $f(x)$ (we assume $f(x)$ is differentiable in the entire domain). Monotonic functions (and monotonic random processes) have previously been studied in areas as diverse as physical sciences for estimating the temperature of a cannon barrel over time [Lavine and Mockus, 1995], marine biology for surveying of fauna on the seabed of the Great Barrier Reef [Hall and Huang, 2001], geology for chronology of sediment samples [Haslett and Parnell, 2008], public health for relating obesity and body fat [Dette and Scheder, 2006], sociology for relating education, work experience and salary [Dette and Scheder, 2006], design of computer networking systems [Golchi et al., 2015], economics for estimating personal income [Canini et al., 2016], insurance for predicting mortality rates [Durot and Lopuhaä, 2018], biology of establishing the diagnostic value of bio-markers for Alzheimer's disease and for trajectory estimation in brain imaging [Lorenzi et al.,

2019, Nader et al., 2019], meteorology for estimation of wind-induced under-catch of winter precipitation [Kim et al., 2018] and many others.

Monotonicity also appears in the more general context of hierarchical models where we want to transform a (simple and typically stationary) input distribution to a (complicated and non-stationary) data distribution. More specifically, monotonicity constraints have been used in hierarchical models with warped inputs, for example, in Bayesian optimisation of non-stationary functions [Snoek et al., 2014] and in mixed effects models for temporal warps of time-series data [Kaiser et al., 2018, Raket et al., 2016].

Extensive study by the statistics [Ramsay, 1988, Sill and Abu-Mostafa, 1997] and machine learning communities [Andersen et al., 2018, Riihimäki and Vehtari, 2010, Lorenzi and Filippone, 2018] has resulted in a variety of frameworks. While many traditional approaches use constrained parametric splines, they are not sufficiently expressive and, typically, do not include prior beliefs about the characteristics of the underlying function (such as smoothness). Consequently, many contemporary methods consider monotonicity in the context of continuous random processes, mostly based on Gaussian processes (GPs) [Rasmussen and Williams, 2005]. As a nonparametric Bayesian model, a GP is an attractive foundation to build flexible and theoretically sound models that provide estimates of uncertainty and automatic complexity control. However, imposing monotonicity constraints on a GP has proven to be problematic [Lin and Dunson, 2014, Riihimäki and Vehtari, 2010] as it requires both formulating a prior that is monotonic as well as constraining the (predictive) posterior to be monotonic. This is particularly challenging as monotonicity is a global property, implying that the function values are correlated for all inputs, irrespective of the lengthscale of the covariance [Andersen et al., 2018].

In this chapter we propose a novel nonparametric Bayesian model of monotonic functions that is based on recent work on differential equations (DEs). At the heart of such models is the idea of approximating the derivatives of a function rather than studying the function directly. DE models have gained popularity in the machine learning community and they have been successfully applied in conjunction with both neural networks [Chen et al., 2018] and GPs [Heinonen et al., 2018, Yildiz et al., 2018a, Yildiz et al., 2018b]. We consider a recently proposed framework, called differential GP flows [Hegde et al., 2019], that performs classification and regression by learning a stochastic differential equation (SDE) transformation of the input space (for a thorough introduction to SDEs, see for example [Øksendal, 1992]). It admits an expressive yet computationally convenient parametrisation using GPs.

Utilising the uniqueness theorem for the solutions of SDEs [Øksendal, 1992], we formulate a novel stochastic random process that is guaranteed to be monotonic. We show that, unlike some of the previous work on monotonic random processes, the proposed approach is guaranteed to lead to monotonic samples from the model (defined as a flow field), and it performs competitively on a set of regression benchmarks.

6.2 Related work

6.2.1 Splines

Many classical approaches to monotonic regression rely on spline smoothing: given a basis of monotone spline functions, the underlying function is approximated using a non-negative linear combination of these basis functions and the monotonicity constraints are satisfied in the entire domain [Wahba, 1978] by construction. For example, [Ramsay, 1998] considers a family of functions defined by the differential equation $D^2 f = \omega D f$ which contains the strictly monotone twice differentiable functions, and approximates ω using a basis of M-splines and I-splines. [Shively et al., 2009] consider a finite approximation using quadratic splines and a set of constraints on the coefficients that ensure isotonicity at the interpolation knots. The use of piecewise linear splines was explored by [Haslett and Parnell, 2008] who use additive i.i.d. gamma increments and a Poisson process to locate the interpolation knots; this leads to a process with a random number of piecewise linear segments of random length, both of which are marginalised analytically. Further examples of spline based approaches rely on cubic splines [Wolberg and Alfy, 2002], mixtures of cumulative distribution functions [Bornkamp and Ickstadt, 2009] and an approximation of the unknown regression function using Bernstein polynomials [Curtis and Ghosh, 2011].

6.2.2 Monotonic stochastic processes

A common approach is to ensure that the monotonicity constraints are satisfied at a finite number of input points. For example, [Da Veiga and Marrel, 2012] use a truncated multi-normal distribution and an approximation of conditional expectations at discrete locations, while [Maatouk, 2017] and [Lopez-Lopera et al., 2019] proposed finite-dimensional approximations based on deterministic basis functions evaluated at a set of knots. Another popular approach proposed by [Riihimäki and Vehtari, 2010] is based on including the derivatives information at a number of input locations by forcing

the derivative process to be positive at these locations. Extensions to this approach include both adapting to new application domains [Golchi et al., 2015, Lorenzi et al., 2019, Siivola et al., 2016] and proposing new inference schemes [Golchi et al., 2015]. However, these approaches do not guarantee monotonicity in the entire domain as they impose constraints at a finite number of points only. [Lin and Dunson, 2014] propose another GP based approach that relies on projecting sample paths from a GP to the space of monotone functions using pooled adjacent violators algorithm which does not impose smoothness. Furthermore, the projection operation complicates the inference of the parameters of the GP and produces distorted credible intervals. [Lenk and Choi, 2017] design shape restricted functions by enforcing that the derivatives of the functions are squared Gaussian processes and approximating the GP using a series expansion with the Karhunen-Loève representation and numerical integration. [Andersen et al., 2018] also model a derivative of a function with a composition of a GP and a non-negative function to ensure it has a constant sign; we refer to this method as transformed GP.

The models of [Lenk and Choi, 2017] and [Andersen et al., 2018] represent a derivative of a function with a non-negative transformation of a GP ($f'(x) = t(g(x))$ where $t(x) \geq 0$ for all x, and $g \sim \mathcal{GP}(\cdot, \cdot)$). Exact inference in such models is generally intractable and these models resort to finite-dimensional approximations of the Gaussian process g. Theoretical results for the marginal likelihood of such models are available in certain special cases, in particular in the case of g being a Wiener process and $t(x) = x^2$ [Kac, 1949] or $t(x) = \exp(x)$ [Matsumoto et al., 2005]. However, the former result only provides the Laplace transform of the marginal likelihood, while the latter one involves a computation of an integral of a highly oscillatory function, requiring high-precision numerical integration and limiting the practical applicability of these methods.

6.3 Background

Any random process can be defined through its finite-dimensional distribution [Øksendal, 1992]. This implies that modelling a certain set of functions $\{g(x)\}$ as trajectories of such a process requires their definition through the finite-dimensional joint distributions $p(g(x_1), \ldots, g(x_n))$. Constraining the functions to be monotonic necessitates choosing a family of joint probability distributions that satisfies the monotonicity constraint:

$$p(g(x_1), \ldots, g(x_n)) = 0, \quad \text{unless} \quad g(x_1) \leq \ldots \leq g(x_n). \tag{MC}$$

One way to achieve this is to truncate commonly used joint distributions (e.g. Gaussian) [Maatouk, 2017]. Inference in such models relies on rejection sampling for truncated Gaussian random variables that are restricted to convex sets [Maatouk and Bay, 2016]. Another approach is to define a random process to have monotone trajectories by construction (e.g. Compound Poisson process [Haslett and Parnell, 2008]) though this often requires making simplifying assumptions on the trajectories (and therefore on $\{g(x)\}$). In this section we use solutions of stochastic differential equations (SDEs) to define a random process with monotonic trajectories by construction without making strong simplifying assumptions.

6.3.1 Gaussian process flows

SDE solutions Our model builds on the general framework for modelling functions as SDE solutions introduced in [Hegde et al., 2019]. The approach is based on considering the following SDE:

$$dS(t,\omega;x) = \mu\left(S(t,\omega;x)\right,t)dt + \sqrt{\sigma^2\left(S(t,\omega;x),t\right)}\,dW(t,\omega) \tag{6.1}$$

where $W(t,\omega)$ is the Wiener process. The Wiener process is a continuous time random process with a zero initial state and independent Gaussian increments [Yildiz et al., 2018b]. The solution of this SDE is a stochastic process $S(t,\omega;x)$ which is a function of three arguments: the time t, the initial value x at time $t=0$, and the element $\omega \in \Omega$ of the underlying sample space Ω. Typically, the dependencies on the initial value and ω are omitted, with the stochastic process being denoted as S_t, however, the dependence on x and ω are crucial for our construction of the monotonic flow model, therefore we explicitly keep them in the notation.

For a fixed time $t=T$, the corresponding SDE solution $S(T,\omega;x)$ is a random variable that depends on the initial condition x. Therefore, there exists a mapping of an arbitrary initial condition to this solution at time T: $x \mapsto S(T,\omega;x)$, and such mappings effectively define distributions over functions (similar to GPs, for example). The family of such distributions is parametrised by functions $\mu\left(S(t,\omega;x),t\right)$ (drift) and $\sigma\left(S(t,\omega;x),t\right)$ (diffusion), which are defined in [Hegde et al., 2019] using a sparse Gaussian process [Titsias, 2009] (hence the name GP flows).

Flow GP Specifically, assume we have N one-dimensional inputs (denoted jointly as $\mathbf{x} = \{x_i\}_{i=1}^N \in \mathbb{R}^N$) and a single-output GP $f \sim \mathcal{GP}(0,\, k(\cdot,\cdot))$, with $f : \mathbb{R}^2 \to \mathbb{R}$

since f is a function of both $\mathbf{x}$ and time t. We specify the GP via a set of M inducing outputs $\mathbf{U} = \{u_i\}_{i=1}^M$, $U_i \in \mathbb{R}$, corresponding to inducing input locations $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^M$, $\mathbf{z}_i \in (\{x\} \times \{t\}) = \mathbb{R}^2$, in a manner similar to [Snelson and Ghahramani, 2006, Titsias, 2009]. The predictive posterior distribution of such a GP is:

$$\begin{aligned} p(f(\mathbf{x}, t) \mid \mathbf{U}, \mathbf{Z}) &\sim \mathcal{N}(\widetilde{\mu}(\mathbf{x}, t), \widetilde{\Sigma}(\mathbf{x}, t)) \\ \widetilde{\mu}(\mathbf{x}, t) &= \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{U} \\ \widetilde{\Sigma}(\mathbf{x}, t) &= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{K}_{\mathbf{zx}} \end{aligned} \tag{6.2}$$

where the matrix $\mathbf{K}_{\mathbf{ab}} := k(\mathbf{a}, \mathbf{b})$. The SDE functions are defined to be $\mu(S(t, \omega; \mathbf{x}), t) := \widetilde{\mu}(S(t, \omega; \mathbf{x}), t)$ and $\sigma(S(t, \omega; \mathbf{x}), t) := \widetilde{\Sigma}(S(t, \omega; \mathbf{x}), t)$ implying that Eq. 6.1 is completely defined by a GP f and its set of inducing points $\mathbf{U}$. Similarly to [Hegde et al., 2019], the joint density of a single path then is:

$$p\left(\mathbf{y}, S(T, \omega; \mathbf{x}), f, \mathbf{U}\right) = \underbrace{p(\mathbf{y} \mid S(T, \omega; \mathbf{x}))}_{\text{likelihood}} \underbrace{p\left(S(T, \omega; \mathbf{x}) \mid f\right)}_{\text{SDE}} \underbrace{p\left(f \mid \mathbf{U}\right) p\left(\mathbf{U}\right)}_{\text{GP prior of } f(\mathbf{x})}. \tag{6.3}$$

Inference Inferring $\mathbf{U}$ is intractable in closed form, hence the posterior of $\mathbf{U}$ is approximated by a variational distribution $q(\mathbf{U}) \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$, the parameters of which are optimised by maximising the likelihood lower bound $\mathcal{L}$:

$$\log p(\mathcal{D}) \geq \mathcal{L} := \mathbb{E}_{q(\mathbf{U})}\mathbb{E}_{p(S(T,\omega;\mathbf{x}) \mid \mathbf{U})}[\log p\left(\mathbf{y} \mid S(T, \omega; \mathbf{x})\right)] - \mathrm{KL}[\, q(\mathbf{U}) \mid\mid p(\mathbf{U})\,]\,. \tag{6.4}$$

The expectation $\mathbb{E}_{p(S(T,\omega;\mathbf{x}) \mid \mathbf{U})}$ is approximated by sampling the numerical approximations of the SDE solutions, which is particularly convenient to do with $\mu\left(S(t, \omega; \mathbf{x}), t\right)$ and $\sigma\left(S(t, \omega; \mathbf{x}), t\right)$ defined as parameters of a GP posterior, because sampling such an SDE solution only requires generating samples from the posterior of the GP given the inducing points $\mathbf{U}$ (see [Hegde et al., 2019] for a thorough discussion of this procedure). The second term in Eq. 6.4 is a KL divergence between two Gaussian distributions available in closed form.

6.4 Monotonic Gaussian process flow

In this section we show that the SDE solutions $S(T, \omega; \mathbf{x})$ are a monotonic function of the initial condition $\mathbf{x}$ for any $T > 0$ subject to some requirements on the sampling used for numerically estimating the first term in Eq. 6.4. Such a model of the monotonic functions can be used to model the observations directly, *i.e.* $y_i \approx S(T, \omega; x_i)$, or as a

first layer in a two-layer hierarchical model in which the monotonic transformation can be considered as a warping of the input space (further discussed in Ch. 7).

We begin with an intuitive discussion of why $S(T, \omega; \mathbf{x})$ is a monotonic function of x using a fluid flow field analogy before providing a formal argument.

1. A general ordinary smooth differential equation $du(t) = \phi(u)dt$ may be thought of as a fluid flow field. Its solutions $u(t, x_1), \ldots, u(t, x_n)$ corresponding to the initial values $x_1, \ldots, x_n$ are trajectories or streams of particles in this field starting at these initial values $x_1, \ldots, x_n$. A fundamental property of such flows is that one can never cross the streams of the flow field. Therefore, if particles are evolved simultaneously under a flow field their ordering cannot be permuted; this gives rise to a monotonicity constraint.

2. A stochastic differential equation, however, introduces random perturbations into the flow field so particles evolving *independently* could jump across flow lines and change their ordering. However, a single, coherent draw from the SDE (corresponding to an individual realisation of the paths $W(\cdot, \omega)$) will always produce a valid flow field (the flow field will simply change between draws). Thus, particles evolving jointly under a single draw will still evolve under a valid flow field and therefore never permute.

6.4.1 SDE solutions are monotonic functions of initial values

It transpires that the joint distribution $p\left(S(T, \omega; x_1), \ldots, S(T, \omega; x_N)\right)$ of solutions of the SDE in Eq. 6.1 with initial values $x_1 \leq \ldots \leq x_N$ (which we compactly denote as $S(T, \omega; \mathbf{x})$, with $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N$) satisfies the monotonicity constraint defined in Eq. MC. This is the consequence of a general result that SDE solutions $S(t, \omega; x)$ are unique and continuous under certain regularity assumptions for any initial value x (see, for example, Theorem 5.2.1 in [Øksendal, 1992]). Specifically, a random variable $S(t, \omega; x)$ is a unique and continuous function of t for any element of the sample space $\omega \in \Omega$.

Using this result we conclude that if we have two initial conditions x and x' such that $x \leq x'$, the corresponding solutions at some time T also obey this ordering, *i.e.* $S(T, \omega; x) \leq S(T, \omega; x')$ for $\omega \in \Omega$. Indeed, were that not the case, the continuity of $S(t, \omega; x)$ as a function of t implies that there exists some $0 \leq t_c \leq T$ such that $S(t_c, \omega; x) = S(t_c, \omega; x')$ (*i.e.* the trajectories corresponding to initial values x and x' cross), resulting in the SDE having two different solutions for the initial condition

$x_c := S(t_c, \omega; x) = S(t_c, \omega; x')$ (namely $S(T - t_c, \omega; x_c) = S(T, \omega; x)$ and $S(T - t_c, \omega; x_c) = S(T, \omega; x')$) violating the uniqueness result.

The above argument assumes a fixed flow field (defined by the drift and the diffusion functions) and a fixed Wiener realisation (corresponding to $W(\cdot, \omega)$) and implies that individual solutions (*i.e.* solutions to a single draw) of the SDEs at a fixed time T, $S(T, \omega; x)$, are monotonic functions of the initial conditions, and hence they define a random process with monotonic trajectories. The actual prior distribution of such trajectories depends on the exact form of the functions $\mu\left(S(t, \omega; x), t\right)$ and $\sigma\left(S(t, \omega; x), t\right)$ in Eq. 6.1 (e.g. if $\sigma(S(t, \omega; x), t) = 0$, the SDE is simply an ordinary DE and $S(T, \omega; x)$ is a deterministic function of x independent of ω, meaning that the prior distribution consists of a single monotonic function). A prior distribution of $\mu\left(S(t, \omega; x), t\right)$ and $\sigma\left(S(t, \omega; x), t\right)$ thus induces a prior distribution over the monotonic functions $S(T, \omega; x)$, and inference in this model consists of computing the posterior distribution of these functions conditioned on the observed noisy sample from a monotonic function.

Numerical solution of the SDE To ensure that the SDE solutions are monotonic functions of the initial values, we make assumptions about the Wiener process realisations $W(\cdot, \omega)$. To compute the SDE solutions under such assumptions, we draw a Wiener process realisation as well as the flow field drift and diffusion, and given these draws, we use the Euler-Maryama numerical solver [Kloeden and Platen, 1992] (following [Hegde et al., 2019]). Specifically, starting with the initial state $(x = x_1, t = 0), \ldots, (x = x_N, t = 0)$, we use Eq. 6.2 to compute the drift and diffusion at the current state, and the discretised version of Eq. 6.1 (*i.e.* with Δt and ΔW instead of dt and dW) to compute the state update Δx. This gives the new state $(x_1 + \Delta x_1, \Delta t), ..., (x_n + \Delta x_n, \Delta t)$, and repeating this procedure $(T/\Delta t)$ times, we arrive at the state $(S(T, \omega; x_1), T), ..., (S(T, \omega; x_N), T)$, corresponding to the approximate SDE solution at time T. The monotonic trajectories are recovered by the numerical SDE solver in the limit of the step size going to zero, $\Delta t \to 0$. Therefore, the step size must be sufficiently small w.r.t. the smoothness of the flow; since we use a GP to define the flow, the smoothness is determined by the lengthscale of the kernel. The effect of the choice of the kernel and the prior on the lengthscale is left for future work.

6.4.2 Notable differences to [Hegde et al., 2019]

1. In [Hegde et al., 2019], a regular GP is placed on top of the SDE solutions $S(T, \omega; \mathbf{x})$, so that $p\left(\mathbf{y} \mid S(T, \omega; \mathbf{x})\right)$ is a GP with a Gaussian likelihood in Eq. 6.4. In contrast,

since we are modelling monotonic functions and $S(T, \omega; \mathbf{x})$ are monotonic functions of $\mathbf{x}$, we define $p(\mathbf{y} \mid S(T, \omega; \mathbf{x}))$ to be directly the likelihood

$$p(\mathbf{y} \mid S(T, \omega; \mathbf{x})) = \mathcal{N}(\mathbf{y} \mid S(T, \omega; \mathbf{x}), \sigma^2 \mathbb{I}). \tag{6.5}$$

2. The argument in this section assumes a fixed flow field (defined by the drift and the diffusion functions) and a fixed Wiener realisation (denoted by ω). Therefore, a critical difference in our inference procedure is that at every iteration of the numerical SDE solver, we use the same drift and diffusion functions, and sample the increments assuming a fixed Wiener realisation for all trajectories (*i.e.* trajectories starting at different initial values) in the flow field. This ensures that they are taken from the same instantaneous realisation of the stochastic flow field and thus the monotonicity constraint is preserved.

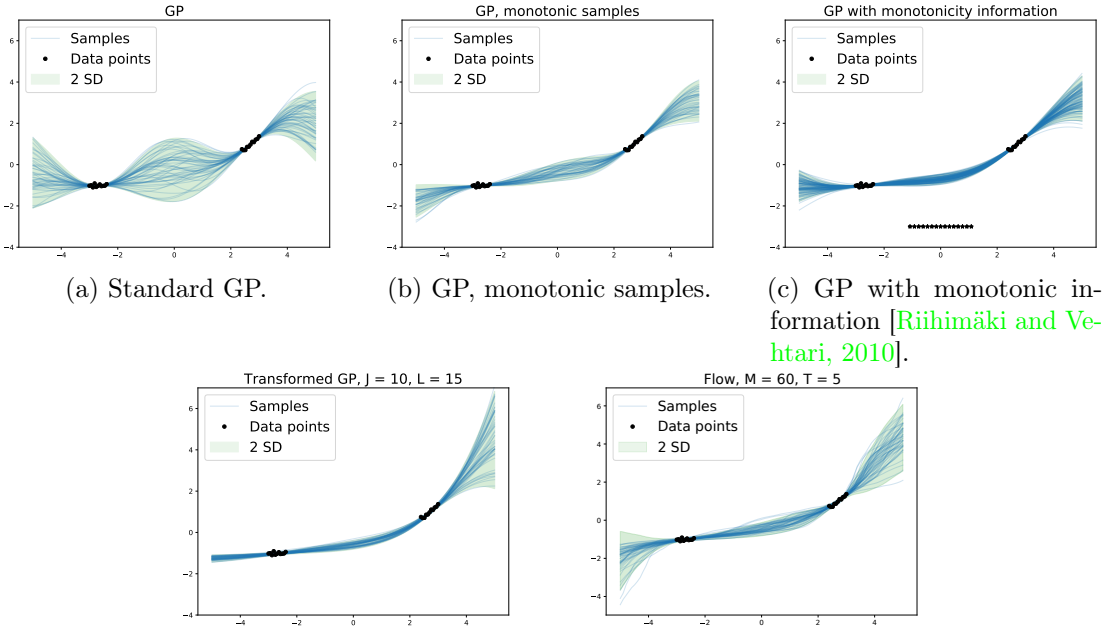
6.5 Experiments

Implementation Our model is implemented in Tensorflow [Abadi et al., 2015]. For the evaluations in Tables 6.1 and 6.2 we use 10000 iterations with the learning rate of 0.01 that gets reduced by a factor of $\sqrt{10}$ when the objective does not improve for more than 500 iterations. For numerical solutions of the SDEs, we use the Euler-Maruyama solver with 20 time steps, as proposed in [Hegde et al., 2019].

Computational complexity The computational complexity of drawing a sample from the monotonic flow model is $\mathcal{O}(N_{\text{steps}}(NM^2 + N))$, where N_{steps} is the number of steps in numerical computation of the approximate SDE solution, NM^2 is the complexity of computing the GP posterior for N inputs based on M inducing points, and N is the complexity of drawing a sample from this posterior. We typically draw fewer than 5 samples to limit the computational cost.

6.5.1 Uncertainty quantification of monotonic models

We start by providing an example (Fig. 6-1) that illustrates the uncertainty estimates of different monotonic models when applied to a toy data set that is known to be monotonic increasing. Our main observation is that imposing a monotonic prior results in the predictive posterior distributions with a smaller variance and allows for more



(a) Standard GP.

(b) GP, monotonic samples.

(c) GP with monotonic information [Riihimäki and Vehtari, 2010].

(d) Transformed GP [Andersen et al., 2018] (VI-full).

(e) Flow (ours).

Figure 6-1: Comparison of the confidence intervals for standard GP, and monotonic regression methods. The samples from the fitted models are shown in blue and the 2 standard deviations from the mean are shown in green.

meaningful extrapolations in comparison to the non-monotonic models; this can be seen by comparing panel (a) that uses standard (non-monotonic) GP to the other panels in Fig. 6-1.

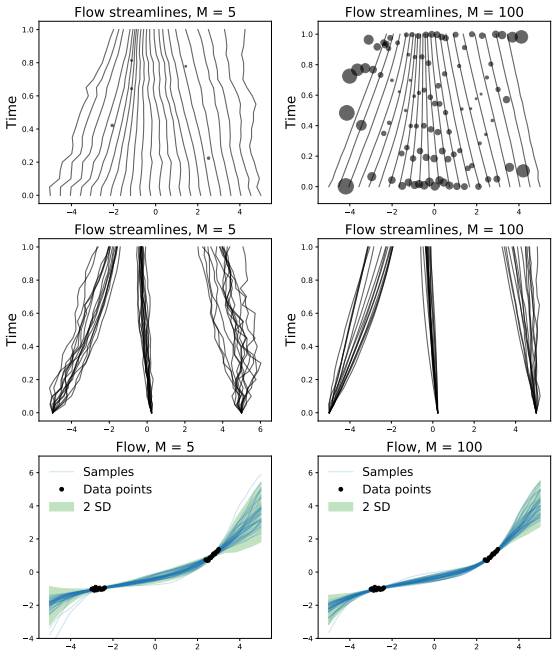
In standard (non-monotonic) regression, GPs are used as the gold standard for the quantification of uncertainty [Foong et al., 2019]. However, a standard GP is a poor baseline for the predictive posterior variance of a monotonic random process due to the additional constraints of monotonicity which lead to tighter confidence intervals as fewer explanations (functions) are compatible with the observed data. To obtain a stronger baseline, we fit a standard GP (Fig. 6-1a) and consider only those samples from the posterior which are monotonically increasing in the domain in which we perform extrapolation ($[-5, 5]$); these samples, along with their mean and 2 SD away from the mean, are shown in Fig. 6-1b. We further note that plotting the error bars using a Gaussian density may be misleading in monotonic regression as the samples from such a process may not be symmetric around the mean, especially when the data are nearly constant, which can be seen by looking at the samples.

The GP with monotonicity information [Riihimäki and Vehtari, 2010] (Fig. 6-1c) is not

able to guarantee that the samples are monotonic, especially in parts of the domain away from the data, while the transformed GP [Andersen et al., 2018] (Fig. 6-1d) tends to underestimate the uncertainty, potentially due to the Dirichlet conditions imposed on the boundaries of the domain. Meanwhile, the uncertainty estimates of our proposed monotonic flow are comparable to the baseline (*i.e.* the monotonic samples from a standard GP) during extrapolation and the samples from the flow are guaranteed to be monotonic.

6.5.2 Example flow fields

The monotonic flow model can be visualised by plotting the streamlines of the input values in the flow field as a function of time. They are shown in Fig. 6-2 as either one coherent draw from the flow (top panels), or as independent samples at a given value of the inputs (middle panels). The latter also help visualise the uncertainty in the model as these samples show the range of possible outputs $S(T, \omega; \mathbf{x})$ for a given input location $\mathbf{x}$. The flow trajectories corresponding to a coherent draw from the flow GP indeed do not cross each other, resulting in a monotonic function that fits the data. This visually illustrates that each draw from a flow GP posterior corresponds to a monotonic function, hence the posterior (and the prior) of the monotonic flow consists entirely of monotonic functions (in contrast to models enforcing monotonicity at a finite number of points, *e.g.* [Riihimäki and Vehtari, 2010]).



(a) Flow streamlines, 5 inducing points. (b) Flow streamlines, 100 inducing points.

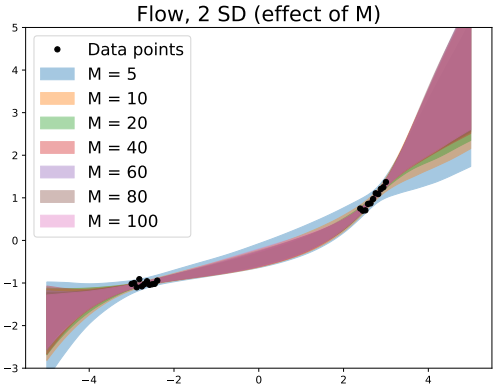
Figure 6-2: A coherent sample (top) and a set of independent samples at three input locations (middle) from a fitted flow (bottom). The circles (top figures) show the locations of the inducing points and are scaled by their (relative) variances.

Most of the inducing points of the flow GP are located along the trajectories starting at the observed x-values. The variance of the inducing points increases with the distance from the trajectories corresponding to the observed values, which implies that away

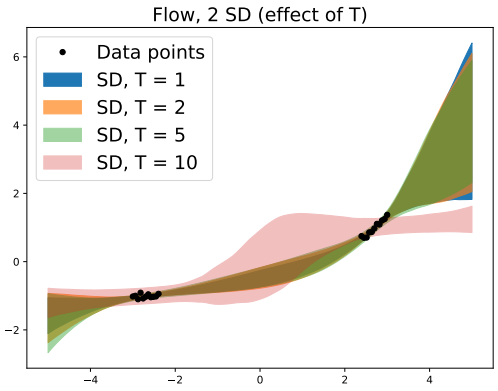
from the observations the inducing points have little influence on the streamlines and the predictions at such x-values correspond to the flow GP prior. For example, a flow GP with a zero mean function induces a distribution over the monotonic functions with the identity as the mean, implying that the extrapolations away from the observations are centered around the identity function (as shown in the bottom panel in Fig. 6-2). Hence the mean function of the flow GP defines the form of the extrapolations, allowing us to impose some prior assumptions about them (in contrast to the finite-dimensional GP approximation of [Andersen et al., 2018], which has boundary conditions limiting the range of possible extrapolations).

Uncertainty as a function of M The mean and the variance of the inducing points in the flow GP depend on their number M as follows: given few inducing points, they are typically optimised to be located close to the observations so that the resulting model fits the observations well (with low estimated observational noise). Meanwhile, given a large number of inducing points, some of them are used to fit the data well while others are placed in regions with no observations (see, for example, the regions in between the data ($[-2, 2]$) in Fig. 6-2b) and optimised to have higher variance $\mathbf{S}$ in those regions. However, the overall uncertainty estimates in the monotonic flow model do not depend much on the number of inducing points. Fig. 6-3a shows how the uncertainty for this data set depends on the number of inducing points: the estimates are nearly identical for $M > 5$ while $M = 5$ may not be enough to explain the data well, hence the observational noise gets overestimated, which in turn results in a higher variance in extrapolation.

Uncertainty as a function of T Similarly, Fig. 6-3b details the dependence on the flow time T. It shows the fits of four monotonic flow models with $T \in \{1, 2, 5, 10\}$, each of which uses 20 steps in the numerical approximation to the solution that estimates the expectation in Eq. 6.4. For $T \in \{1, 2, 5\}$ the fits are quite similar, but the fit for $T = 10$ is poor suggesting a potential numerical approximation issue since the size of the discretised time steps ($dt = T/20$) in the Euler-Maryama method grows linearly with T, and for larger values of T the numerical approximation might be poor. We also note that as the flow time increases, the resulting fits seem to become more extreme and the predictive uncertainty estimates away from the data increase (*e.g.* see the uncertainty estimates in the left part of Fig. 6-3b), which might also be indicative of an issue with the numerical approximation.



(a) Flow comparison for different number of inducing points M. This shows that 5 inducing points might not be enough to model this data set while the flows for all other values of M are similar.



(b) Flow comparison for $T = 1, 2, 5, 10$. The effect of the total time T seems small for most values of T. When T is high (in this example, $T = 10$) then there may be some issues with numerical stability that leads to poor performance of this model.

Figure 6-3: Effect of the number M of inducing points and the total flow time T on the estimated uncertainty (coloured regions correspond to 2 SD away from the mean of the samples from the flow). Results for 5 random trials.

6.5.3 Regression

We test the monotonic flow model on the task of estimating 1D monotonic curves from noisy data. We use a set of 6 benchmark functions from previous studies [Lin and Dunson, 2014, Maatouk, 2017, Shively et al., 2009]. The functions we use for evaluations are the following:

$$f_1(x) = 3, \quad x \in (0, 10] \qquad \text{(flat function)}$$

$$f_2(x) = 0.32\,(x + \sin(x)), \quad x \in (0, 10] \qquad \text{(sinusoidal function)}$$

$$f_3(x) = 3 \text{ if } x \in (0, 8], \quad f_3(x) = 6 \text{ if } x \in (8, 10] \qquad \text{(step function)}$$

$$f_4(x) = 0.3x, \quad x \in (0, 10] \qquad \text{(linear function)}$$

$$f_5(x) = 0.15\,\exp(0.6x - 3), \quad x \in (0, 10] \qquad \text{(exponential function)}$$

$$f_6(x) = 3\,/\,[1 + \exp(-2x + 10)], \quad x \in (0, 10] \qquad \text{(logistic function)}$$

Examples of three such functions are shown in Fig. 6-4. The training data is generated by evaluating these functions at n equally spaced points and adding i.i.d. Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, 1)$. We note that many real-life data sets that benefit from including the monotonicity constraints have similar trends and high levels of noise (e.g. [Curtis and

Table 6.1: Root-mean-square error ± SD (×100) of 20 trials for data of size $n = 100$

	flat	sinusoidal	step	linear	exponential	logistic
GP	15.1	21.9	27.1	16.7	19.7	25.5
GP projection [Lin and Dunson, 2014]	11.3	21.1	25.3	16.3	19.1	22.4
Regression splines [Shively et al., 2009]	9.7	22.9	28.5	24.0	21.3	19.4
GP approximation [Maatouk, 2017]	8.2	20.6	41.1	15.8	20.8	21.0
GP with derivatives [Riihimäki and Vehtari, 2010]	16.5 ± 5.1	19.9 ± 2.9	68.6 ± 5.5	16.3 ± 7.6	27.4 ± 6.5	30.1 ± 5.7
Transformed GP [Andersen et al., 2018] (VI-full)	**6.4** ± 4.5	20.6 ± 5.9	52.5 ± 3.6	**11.6** ± 5.8	17.5 ± 7.3	**17.1** ± 6.2
Monotonic Flow (ours)	6.8 ± 3.2	**17.9** ± 4.2	**20.5** ± 5.0	13.2 ± 6.7	**14.4** ± 4.8	18.1 ± 5.0

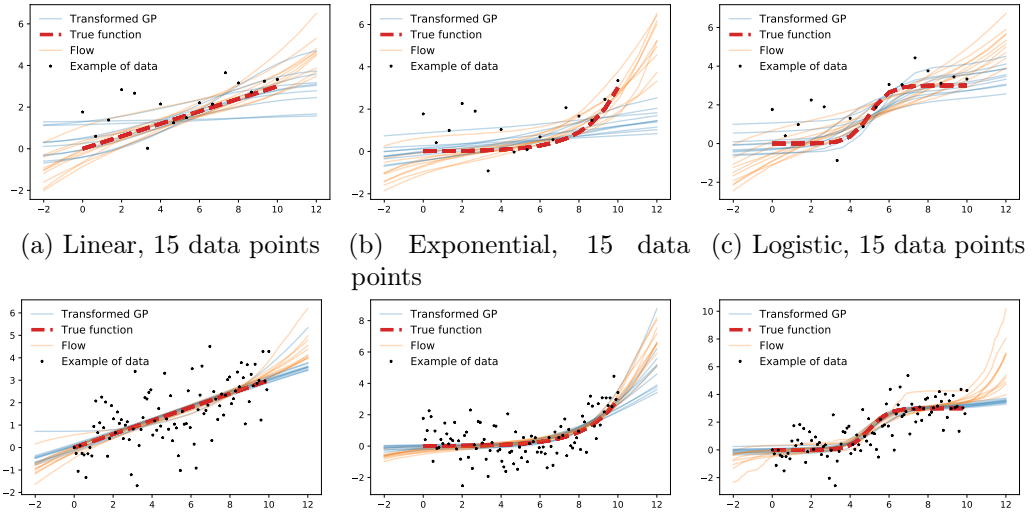
Ghosh, 2011, Haslett and Parnell, 2008, Kim et al., 2018]). Following the literature, we used the root-mean-square-error (RMSE) to evaluate the performance of the model.

100 data points In Table 6.1 we provide the results obtained by fitting different monotonic models to data sets containing $n = 100$ points. As baselines we include: GPs with monotonicity information [Riihimäki and Vehtari, 2010][1], transformed GPs [Andersen et al., 2018][2], and other results reported in the literature. We report the RMSE means and the SD from 20 trial runs with different random noise samples and show example fits in the bottom row of Fig. 6-4. This figure contains the means of the predicted curves from 10 trials with the best parameter values (each trial contains a different sample of standard Gaussian random noise). We plot samples as opposed to the mean and the standard deviation as, due to the monotonicity constraint, samples are more informative than sample statistics. For the GP with monotonicity information we choose M virtual points and place them equidistantly in the range of the data; we provide the best RMSEs for $M \in [10, 20, 50, 100]$. For the transformed GP we report the best results for the boundary conditions $L \in [10, 15, 20, 30]$ and the number of terms in the approximation $J \in [2, 3, 5, 10, 15, 20, 25, 30]$. For both models we use a squared exponential kernel. Our method depends on the time T, the kernel and the number of inducing points M. For this experiment, we consider $T \in [1, 5]$, $M = 40$ and two kernel options, squared exponential and ARD Matérn 3/2. The lowest RMSE are achieved using the flow and the transformed GP. Overall, our method performs very competitively, achieving the best results on 3 functions and being within the standard deviation of the best result on all others. We note that the training data is very noisy (see Fig. 6-4), therefore using prior monotonicity assumptions achieves significantly improved results over a regular GP.

[1] Implementation available from https://research.cs.aalto.fi/pml/software/gpstuff/.
[2] Implementation provided in personal communications.

Table 6.2: Root-mean-square error ± SD (×100) of 20 trials for data of size $n = 15$

	flat	sinusoidal	step	linear	exponential	logistic
Transformed GP [Andersen et al., 2018] (VI-full)	**18.5** ± 14.4	40.0 ± 17.5	101.9 ± 11.4	37.4 ± 22.8	52.9 ± 11.9	51.7 ± 19.6
Monotonic Flow (ours)	21.7 ± 15.0	**39.1** ± 13.0	**64.5** ± 10.7	**30.8** ± 12.0	**32.8** ± 17.9	**43.2** ± 15.2



(a) Linear, 15 data points (b) Exponential, 15 data points (c) Logistic, 15 data points

(d) Linear, 100 data points (e) Exponential, 100 data points (f) Logistic, 100 data points

Figure 6-4: Mean fits for 10 trials with different random noise as estimated by the flow and the transformed GP [Andersen et al., 2018] (the noise samples are identical for both methods; we plot the data from one trial).

15 data points In Table 6.2 and Fig. 6-4 (top row) we provide the comparison of the flow and the transformed GP in a setting when only $n = 15$ data points are available. Our fully nonparametric model is able to recover the structure in the data significantly better than the transformed GP which usually reverts to a nearly linear fit on all functions. This might be explained by the fact that the transformed GP is a parametric approximation of a monotonic GP, hence the more parameters are included, the larger the variety of the functions it can model. However, estimating a large (w.r.t. data set size) number of parameters is challenging given a small set of noisy observations. The transformed GP model [Andersen et al., 2018] tends to underestimate the value of the function on the left side of the domain and overestimate the value on the right. The mean of our prior of the monotonic flow with a stationary flow GP kernel is an identity function, so given a small set of noisy observations, the predictive posterior mean quickly reverts to the prior distribution near the edges of the data interval.

Non-Gaussian noise The inference procedures for the monotonic flow and for the two-layer model can be easily applied to arbitrary likelihoods, because they are based on stochastic variational inference and do not require the closed form integrals of the likelihood density.

6.6 Discussion

In this chapter we have reviewed some of the existing work on monotonic random processes and have proposed a novel nonparametric model of monotonic functions based on a random process with monotonic trajectories that confers improved performance over the state-of-the-art as well as other favorable characteristics. Many real-life regression tasks deal with functions that are known to be monotonic, and explicitly imposing this constraint helps uncover the structure in the data, especially when the observations are noisy or data are scarce.

Discriminative classifiers benefit from hierarchical structures to collapse the input domain to encode invariances. This can be embodied as a hierarchy of (non-injective) mappings. If we wish to build generative probabilistic models (e.g. for unsupervised representation learning) we seek to explain the observed data with the models achieving high marginal data likelihood. To do so, we want to transform a (simple) input distribution to a (complicated) data distribution. In a hierarchical generative model, this necessitates a composition of injective mappings for all but the output layer to ensure that the resulting distribution does not collapse to a degenerate one. This property is met if the hidden layers in the model comprise monotonic transformations.

The compositions of a monotonic warp and a GP can be put in the context of hierarchical probabilistic models. Specifically, in the presence of additional mid-hierarchy marginal information or domain specific knowledge of the compositional priors [Kaiser et al., 2018], existing hierarchical models may necessitate a composition of monotonic mappings for all but the output layer. That further advocates the study of nonparametric models of monotonic functions, which can serve as a general purpose first layer in a hierarchical model, especially when the data is known to be non-stationary.

In the next chapter we return to the alignment task and use the monotonic flow as a model for the time warpings. This allows us to use the probabilistic formulation of the monotonic flow for the quantification of the warping uncertainty, which in turn allows us to reason about the existence of different groups of sequences within the data set.

7

Alignment using monotonic Gaussian process flows

After studying the probabilistic models of monotonic functions in the previous chapter, in this chapter we return to the temporal alignment of sequences. Our main goal is to use these models of monotonic functions to equip the alignment models introduced in Ch. 3 and 4 with probabilistic warps, capable of capturing and propagating the warping uncertainty.

7.1 Introduction

One of the difficulties with the temporal alignment task is its inherent ambiguity. We model the observations as a composition of the latent function f and the warp g:

$$\mathbf{y} = (f \circ g)(\mathbf{x}), \tag{7.1}$$

where both f and g are unknown, which means the problem is under-constrained, allowing for infinitely many solutions. We refer to this ambiguity as compositional uncertainty: even noiseless observed data could be generated by compositions of a multitude of different functions which are consistent with the prior. This is illustrated in Fig. 7-1, in which the same two observed sequences (third column) can be represented as compositions of different warps (first column) and latent functions (second column; note that the same latent function is used for both warps satisfying the alignment constraint).

In Ch. 3, we introduced the GP priors on the warps and the latent functions, which, in

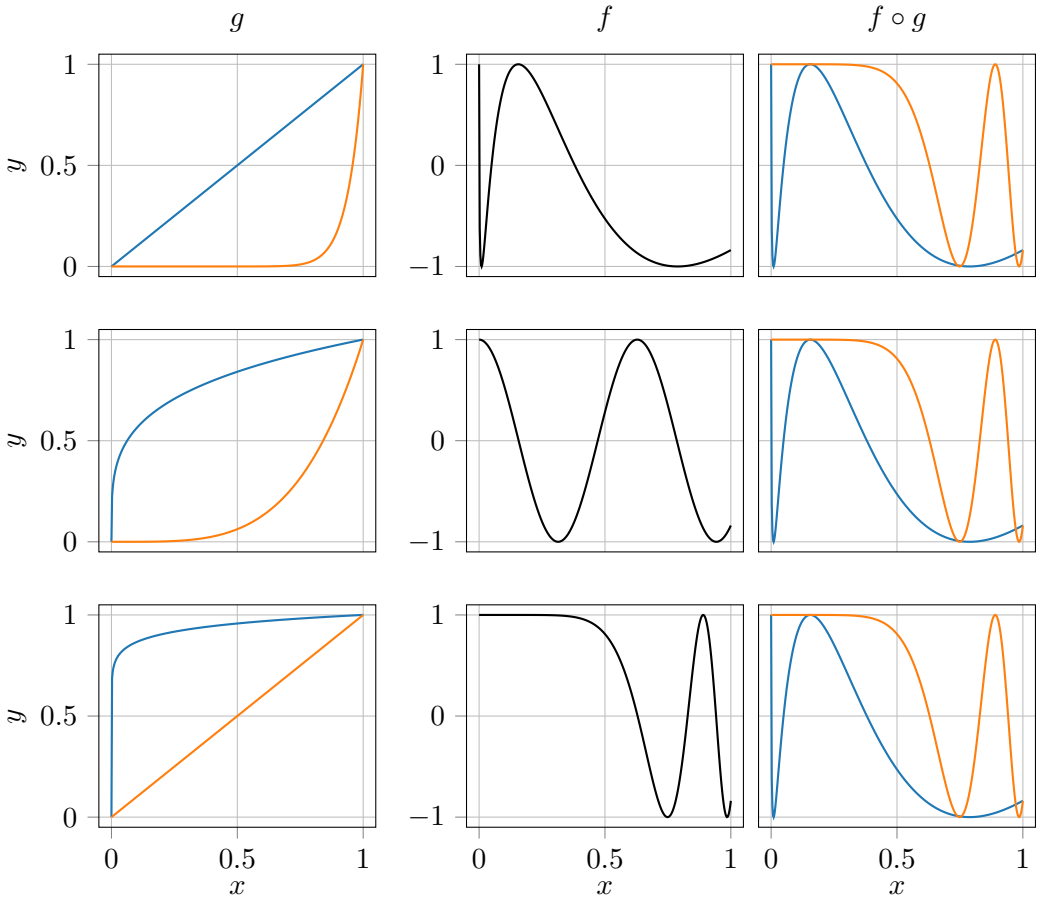


Figure 7-1: Illustration of ambiguity in a two-layer model: a composite function $\mathbf{y} = f \circ g$ (third column) can be represented as a composition of different warps g (first column) and latent functions f (second column).

conjunction with MAP estimates, allowed us to systematically choose one of the possible alignment solutions (*i.e.* compositions of warps and latent functions). For example, the warps in the second row of Fig. 7-1 are less extreme and the corresponding latent function is more stationary than in the other rows, meaning that such a solution has higher prior probability in our model, and the optimisation is more likely to converge to it rather than to the other possible solutions shown in the other rows.

7.1.1 Bayesian inference in temporal alignment model

Bayesian modelling allows us not only to compute the most likely alignment solution but to retain the uncertainty about the solution by computing the entire distribution of the compositions of warps and latent functions consistent with the data. Conceptually,

we can do so using Bayes' rule because we now have all components of our alignment model (*i.e.* the warps, the latent functions and the alignment objective) defined as prior probability distributions over the corresponding objects. However, in practice computing the exact posterior distribution is intractable computationally and hence we resort to variational inference to find an approximate posterior. Next, we briefly outline what types of uncertainty we would like to propagate through the model and what requirements on the variational posterior they pose, while a detailed discussion of these issues is provided in further sections in this chapter.

7.1.2 Types of uncertainty in alignment model

Warping uncertainty As illustrated in Fig. 7-1, there are multiple different warps that allow us to fit the observations using the same latent function for all sequences (resulting in aligned latent functions). In the same figure we can see that there are two types of dependencies that need to be captured by the variational distribution:

1. For each sequence the variational distributions $q(g_i)$ and $q(f_i)$ over the warps and the latent functions must be dependent, since any composition $f_i \circ g_i$ drawn from these distributions must fit the data. If $q(g_i)$ and $q(f_i)$ are independent, the only way to make every draw fit the observations is to collapse these distributions to a point mass around a particular choice of the warp and the latent function, the composition of which fits the data.

2. The distributions $q(g_1), \ldots, q(g_J)$ over the warps corresponding to different sequences of the same group (*i.e.* aligned to the same sequence) must be dependent, since they are used in composition with the *same* latent function. For example, if we use a blue warp from the first row in Fig. 7-1, we must use the orange warp from the same row, otherwise these two warps cannot be used in conjunction with the same latent function to fit the observations.

To satisfy these dependencies we use variational distributions over the warps and the latent functions factorised as

$$q(g_1, \ldots, g_J, f_1, \ldots, f_J) = q(g_1, \ldots, g_J) q(f_1 \mid g_1) \ldots q(f_J \mid g_J). \tag{7.2}$$

The detailed discussion of this variational distribution is provided in the further sections.

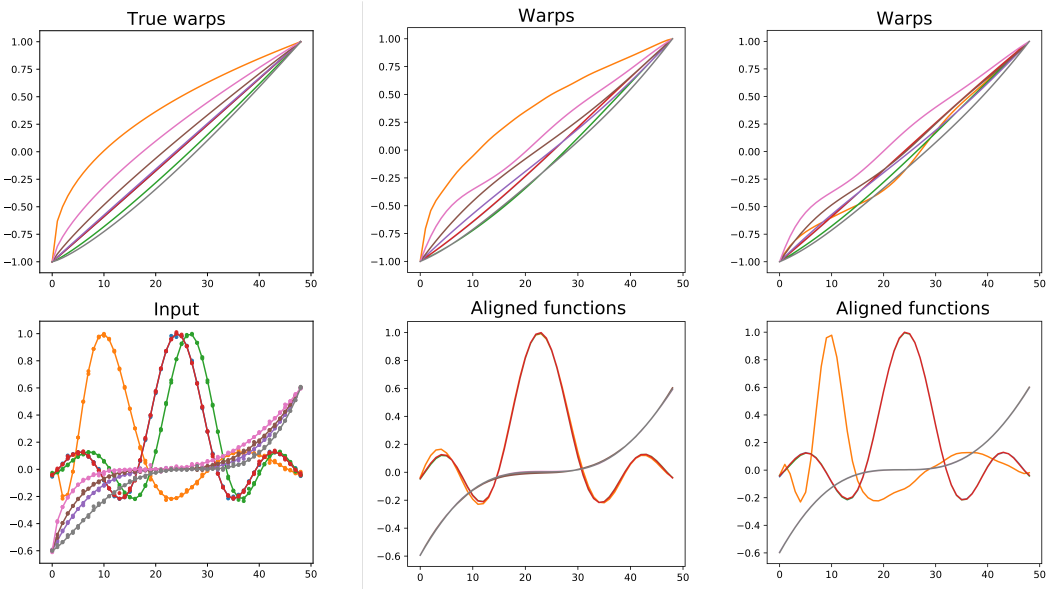


Figure 7-2: The observations (bottom left) were generated by applying warping functions (top left) to a sinc or cubic function. The point estimates in the alignment model of previous chapters return one of the two possible solutions (middle and right columns). One solution (middle) aligns all the sequences into two groups but uses more extreme warps (note the orange warp) while the other solution (right) assigns the orange sequence to a new cluster (and therefore uses a warp that is close to identity for this sequence). Both are plausible given our priors on the warps $g_j(\cdot)$ and the functions $f_j(\cdot)$, therefore, a preferred model would preserve the uncertainty about the warps and the cluster assignments and hence capture the full range of possible solutions.

Group assignment uncertainty Probabilistic alignment objectives (such as GPLVM or DPMM, discussed in Ch. 3 and 4, respectively) allow us to not only automatically assign sequences to multiple groups but also to model the uncertainty over the group assignment for each sequence. For example, if all but one sequences are close to each other, we might be uncertain if the remaining one should aligned to the rest (requiring a warp which significantly deviates from the identity warp), or assigned to its own cluster. Such a case is illustrated in Fig. 7-2.

This example shows that the variational distribution over the warps must depend on the cluster assignments (either explicitly on the distribution over cluster assignments in the DPMM case, or implicitly on the distribution of the latent space in the GPLVM). Otherwise, the model would collapse to the most likely group assignment rather than capturing the uncertainty over such assignments. As an illustration of this idea, notice how the warp of the orange curve in Fig. 7-2 changes depending on assigning it to the already existing group (second column) or to the new one (third column).

7.2 Compositions of monotonic flow and GPs

In this section we study the inference procedures that are appropriate in order to capture the posterior uncertainty in the alignment model. We begin by drawing some parallels to compositions of GPs (with no monotonicity constraints) and discuss the typical inference procedures used in these model. We then adapt one of these inference schemes of compositional GPs to the two-layer model of a monotonic flow and a GP introduced in Sec. 7.1.

7.2.1 Aside: Deep GPs

Our model is a composition of a monotonic warp and a GP modelling the latent function, which is similar to compositions of GPs often called deep GPs [Damianou and Lawrence, 2013]. A deep GP is not itself a GP and exact Bayesian inference is also intractable for such models. Therefore, various approximate methods have been developed [Salimbeni and Deisenroth, 2017, Havasi et al., 2018], of which variational inference is of particular interest to us.

The typical variational inference approaches for deep GPs consist of augmenting each GP with a set of inducing points and computing the output of each layer as a predictive posterior conditioned on these inducing points. The corresponding variational lower bound can be computed using stochastic Monte-Carlo estimators for the expectations or in closed form for certain choices of kernels (for more details see [Damianou and Lawrence, 2013]). That setting is very similar to our model since we also define the monotonic flow using the inducing points, which enables us to apply the variational inference procedures from the deep GPs literature to the two-layer compositions of the monotonic flow and a GP (subject to some straightforward modifications of the deep GP lower bound to include the monotonic flow which we discuss in Sec. 7.2.2).

More specifically, we follow the approach of [Salimbeni and Deisenroth, 2017] that is easy to extend to our setting. We briefly review it here and adapt it to the monotonic flow in Sec. 7.2.2.

Assume we have a data set $(\mathbf{x}, \mathbf{y}) = \{(x_i, y_i)\}_{i=1}^N$, which we model by a composition of two GPs, i.e. $\mathbf{y} \sim f_2(f_1(\mathbf{x}))$ with $f_i \sim \mathcal{GP}(\mu_i, k_i)$. Using notation $\mathbf{f}_i$ to refer to the finite-dimensional evaluations of f_i at $\mathbf{x}$ (i.e. $p(\mathbf{f}_1 \mid \mathbf{x}) \sim f_1(\mathbf{x})$, $p(\mathbf{f}_2 \mid \mathbf{f}_1) \sim f_2(\mathbf{f}_1)$), the

marginal data likelihood of such a two-layer GP is

$$p(\mathbf{y} \mid \mathbf{x}) = \iint p(\mathbf{y} \mid \mathbf{f}_2) p(\mathbf{f}_2 \mid \mathbf{f}_1) p(\mathbf{f}_1 \mid \mathbf{x}) d\mathbf{f}_2 d\mathbf{f}_1. \tag{7.3}$$

However, such an integral is intractable, because integrating over $\mathbf{f}_1$ requires integrating the Gaussian density $p(\mathbf{f}_2 \mid \mathbf{f}_1)$ through the (non-linear) covariance matrix $k_2(\mathbf{f}_1, \mathbf{f}_1)$. A standard approach allowing us to compute a bound on this intractable likelihood involves augmenting the model with the so-called inducing points. For each individual GP in a deep GP we introduce the input-output pairs $\{\mathbf{z}_i, \mathbf{U}_i\}$ (where i indexes GPs in the deep GP; $i \in \{1, 2\}$ in our case), which are similar to regular observations as a prior on these points is the corresponding GP prior $p(\mathbf{U}_i \mid \mathbf{z}_i) \sim f_i(\mathbf{z}_i)$. The joint distribution then becomes

$$\begin{aligned} p(\mathbf{y}, \mathbf{f}_2, \mathbf{f}_1, \mathbf{U}_2, \mathbf{U}_1 \mid \mathbf{z}_2, \mathbf{z}_1, \mathbf{x}) = & p(\mathbf{y} \mid \mathbf{f}_2) p(\mathbf{f}_2 \mid \mathbf{U}_2, \mathbf{z}_2, \mathbf{f}_1) p(\mathbf{f}_1 \mid \mathbf{U}_1, \mathbf{z}_1, \mathbf{x}) \\ & p(\mathbf{U}_2 \mid \mathbf{z}_2) p(\mathbf{U}_1 \mid \mathbf{z}_1), \end{aligned} \tag{7.4}$$

where $p(\mathbf{f}_2 \mid \mathbf{U}_2, \mathbf{z}_2, \mathbf{f}_1)$ and $p(\mathbf{f}_1 \mid \mathbf{U}_1, \mathbf{z}_1, \mathbf{x})$ are the GP posteriors at inputs $\mathbf{f}_1$ and $\mathbf{x}$ respectively given the inducing points $\{\mathbf{z}_2, \mathbf{U}_2\}$ and $\{\mathbf{z}_1, \mathbf{U}_1\}$.

Now, instead of integrating $\mathbf{U}_i$ in Eq. 7.4, we can treat them as parameters to be optimised. Specifically, we introduce a variational distribution $q(\mathbf{U}_1, \mathbf{U}_2; \Theta)$ (typically a multivariate Gaussian) with parameters Θ allowing us to compute the following lower bound on $p(\mathbf{y} \mid \mathbf{x})$:

$$\log p(\mathbf{y}|\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{U}_1, \mathbf{U}_2) q(\mathbf{f}_2 \mid \mathbf{U}_1, \mathbf{U}_2)}[p(\mathbf{y} \mid \mathbf{f}_2)] - \mathrm{KL}[\, q(\mathbf{U}_1, \mathbf{U}_2) \,||\, p(\mathbf{U}_1) p(\mathbf{U}_2) \,]. \tag{7.5}$$

Choosing the variational distribution to be a factorised Gaussian $q(\mathbf{U}_1, \mathbf{U}_2) = q(\mathbf{U}_1) q(\mathbf{U}_2)$, $q(\mathbf{U}_i) \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$ allows us to efficiently estimate the first term in Eq. 7.5 by sampling as in [Salimbeni and Deisenroth, 2017], or to compute it analytically under certain assumptions on the GP covariance functions $k_2(\cdot, \cdot)$ and $k_1(\cdot, \cdot)$ [Damianou and Lawrence, 2013]. The inference under such approximation consists of finding the variational parameters Θ (*e.g.* the mean $\mathbf{m}_i$ and the covariance $\mathbf{S}_i$ in the case of Gaussian variational distributions) maximising the lower bound Eq. 7.5.

However, factorised variational posterior does not capture the dependencies between the layers, which are necessary for propagating the warping uncertainty (Sec. 7.1.2). In the following sections we propose a way to overcome this limitation.

7.2.2 Variational inference for compositions of flow and GPs

We now discuss how the inference scheme for deep GPs outlined in the previous section can be easily adapted to two-layer compositions of the flow, as introduced in Ch. 6, and an output GP. To do so, we augment both the monotonic flow and the GP that is placed on top of the outputs of the flow with the inducing points, allowing us to use a lower bound on the likelihood similar to Eq. 7.5.

We consider compositions of a monotonic flow defined using a flow field GP $f^{\mathrm{g}} \sim \mathcal{GP}(0, k_{\mathrm{g}}(\cdot,\cdot))$ with corresponding inducing points $\mathbf{U}^{\mathrm{g}} = \{u_i^{\mathrm{g}}\}$ (Sec. 6.3 and 6.4), and an output function $f \sim \mathcal{GP}(0, k_f(\cdot,\cdot))$. We augment the output GP f with a set of M inducing points $\mathbf{U}^f = \{u^f\}_{i=1}^M$ corresponding to inducing locations $\mathbf{z}^f = \{z_i^f\}_{i=1}^M$ ($u_i^f, z_i^f \in \mathbb{R}$). Given the inputs and the noisy observations, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the marginal likelihood of the composite model is as follows:

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}) = \int & p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f, \mathbf{z}^f) p(\mathbf{U}^f \mid \mathbf{z}^f) \\ & p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}}, \mathbf{z}^{\mathrm{g}}) p(\mathbf{U}^{\mathrm{g}} \mid \mathbf{z}^{\mathrm{g}}) \,\mathrm{d}\mathbf{f}\, \mathrm{d}S(\mathbf{x})\, \mathrm{d}\mathbf{U}^{\mathrm{g}}, \end{aligned} \tag{7.6}$$

where $p(\mathbf{y} \mid \mathbf{f})$ is the observational noise likelihood, $S(\mathbf{x}) := S(T, \omega; \mathbf{x})$ is the monotonic flow SDE solution with dependencies on ω and T omitted to simplify the notation, and $p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f)$ is the posterior of the output GP given the inducing points $\mathbf{U}^f$ and evaluated at the flow outputs $S(\mathbf{x}) \in \mathbb{R}^N$:

$$\begin{aligned} p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f) &\sim \mathcal{N}(\boldsymbol{\mu}_f(S(\mathbf{x})),\, \boldsymbol{\Sigma}_f(S(\mathbf{x}))), \\ \boldsymbol{\mu}_f(S(\mathbf{x})) &= \mathbf{K}^f_{S(\mathbf{x})\mathbf{z}^f} \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f}\right)^{-1} \mathbf{U}^f, \\ \boldsymbol{\Sigma}_f(S(\mathbf{x})) &= \mathbf{K}^f_{S(\mathbf{x})S(\mathbf{x})} - \mathbf{K}^f_{S(\mathbf{x})\mathbf{z}^f} \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f}\right)^{-1} \mathbf{K}^f_{\mathbf{z}^f S(\mathbf{x})}, \end{aligned} \tag{7.7}$$

where $\mathbf{K}^f_{ab} = k_f(a, b)$. Exact inference (*i.e.* computing the posterior $p(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f \mid \mathcal{D})$ over the inducing points) is intractable in closed form, therefore we compute an approximate posterior using variational inference. Introducing a variational distribution $q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f)$, we obtain a likelihood lower bound similar to Eq. 7.5:

$$\begin{aligned} \log p(\mathbf{y} \mid \mathbf{x}) \geq\, & \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f) p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}}) p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f)} [\log p(\mathbf{y} \mid \mathbf{f})] - \\ & - \mathrm{KL}[q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f) \,||\, p(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f)]. \end{aligned} \tag{7.8}$$

The two inner expectations are taken over the distributions of the SDE solutions $S(\mathbf{x})$ (monotonic flow outputs) and the output GP evaluated at $S(\mathbf{x})$. Assuming a factorised

Gaussian variational distribution $q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f) = q(\mathbf{U}^{\mathrm{g}})q(\mathbf{U}^f)$ with $q(\mathbf{U}^{\mathrm{g}}) \sim \mathcal{N}(\mathbf{m}_{\mathrm{g}}, \mathbf{S}_{\mathrm{g}})$ and $q(\mathbf{U}^f) \sim \mathcal{N}(\mathbf{m}_f, \mathbf{S}_f)$, we can analytically integrate $\mathbf{U}^f$ out in Eq. 7.8 obtaining

$$
\begin{aligned}
q(\mathbf{f} \mid S(\mathbf{x})) &= \int q(\mathbf{U}^f) p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f) \mathrm{d}\mathbf{U}^f = \mathcal{N}(\tilde{\boldsymbol{\mu}}_f, \tilde{\boldsymbol{\Sigma}}_f), \\
\tilde{\boldsymbol{\mu}}_f &= \mathbf{K}^f_{S(\mathbf{x})\mathbf{z}^f} \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f}\right)^{-1} \mathbf{m}_f, \\
\tilde{\boldsymbol{\Sigma}}_f &= \mathbf{K}^f_{S(\mathbf{x})S(\mathbf{x})} - \mathbf{K}^f_{S(\mathbf{x})\mathbf{z}^f} \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f}\right)^{-1} \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f} - \mathbf{S}_f\right) \left(\mathbf{K}^f_{\mathbf{z}^f\mathbf{z}^f}\right)^{-1} \mathbf{K}^f_{\mathbf{z}^f S(\mathbf{x})}.
\end{aligned}
\tag{7.9}
$$

Assuming a Gaussian likelihood $p(\mathbf{y} \mid \mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma_f^2 \mathbb{I})$ we can write the first term in Eq. 7.8 as

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f) p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}}) p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f)}[\log p(\mathbf{y} \mid \mathbf{f})] = \\
&\quad = \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}) p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})} \mathbb{E}_{q(\mathbf{U}^f) p(\mathbf{f} \mid S(\mathbf{x}), \mathbf{U}^f)}[\log p(\mathbf{y} \mid \mathbf{f})] \\
&\quad = \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}) p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})} \mathbb{E}_{q(\mathbf{f} \mid S(\mathbf{x}))} \left[-\frac{N}{2} \log(2\pi\sigma_f^2) - \frac{1}{2\sigma_f^2} (\mathbf{y} - \mathbf{f})^T (\mathbf{y} - \mathbf{f}) \right] \\
&\quad = \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}) p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})} \left[-\frac{N}{2} \log(2\pi\sigma_f^2) - \frac{1}{2\sigma_f^2} (\mathbf{y} - \tilde{\boldsymbol{\mu}}_f)^T (\mathbf{y} - \tilde{\boldsymbol{\mu}}_f) - \mathrm{Tr}\left(\tilde{\boldsymbol{\Sigma}}_f \right) \right].
\end{aligned}
\tag{7.10}
$$

The expectation over $\mathbf{U}^{\mathrm{g}}$, the inducing points for the flow GP, in the equation above can be computed analytically [Hegde et al., 2019], and the remaining expectation over $p(S(\mathbf{x}))$ can be approximated by sampling the numerical solutions of the flow SDE as discussed in Ch. 6. The second term in Eq. 7.8 (the KL divergence) can be computed analytically for a Gaussian variational distribution.

The assumption of the factorised variational distribution over the inducing points leads to a convenient form of the marginal likelihood bound in Eq. 7.8 with all expectations computed analytically apart from the SDE numerical solutions. However, a problem with such an approach is that it does not model the dependencies between the flow warp and the output GP, which is necessary in order to capture the warping uncertainty as argued in Sec. 7.1.2. This is also demonstrated in the first row of Fig. 7-3, which shows a fit of a composition of a monotonic flow and a GP with factorised variational distribution over the inducing points (the setting discussed in this section). We can see that the monotonic flow collapsed to a single solution not capturing the compositional (or warping) uncertainty demonstrated in Fig. 7-1.

Example: Composite regression

Let us consider the task of fitting a two-layer model to noisy data of a chirp function $y = \sin(2\exp(x+1)) + \varepsilon, \varepsilon \sim \mathcal{N}(0, 0.1)$. Imposing monotonic constraints on the first layer allows us to warp the inputs to the second layer in a way that the observations and the warped inputs can be modelled using a standard stationary kernel. Fig. 7-3 shows the fitted function, the warps produced by the monotonically constrained first layer, and the fitted function in the warped coordinates (i.e. samples of the output GP against the samples from the flow).

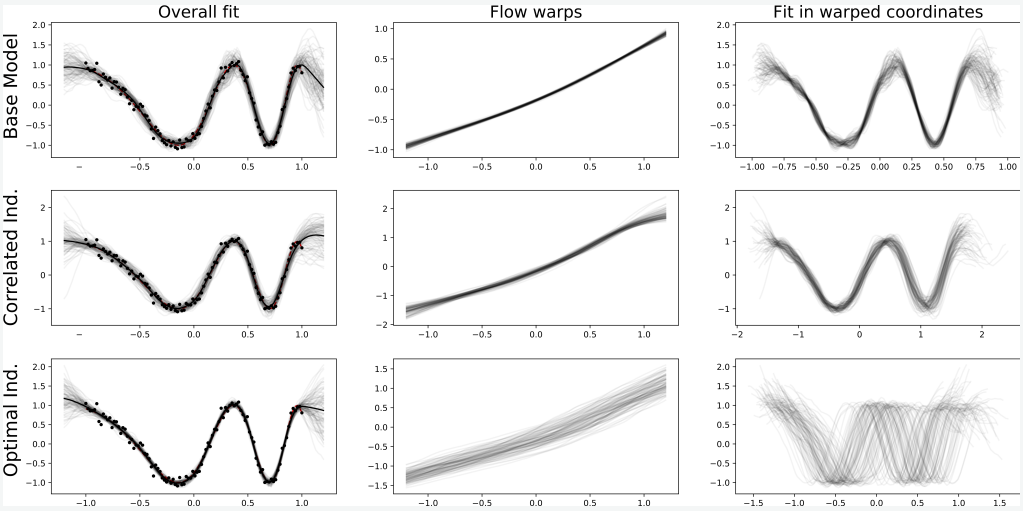


Figure 7-3: Layers of a two-layer models fitted to a chirp function. From left to right we plot the overall fit to the noisy observations (black dots), the warp produced by the monotonic first layer, and the fit in the warped coordinates; we show the base model, one with correlated inducing points, and one with optimal inducing points.

The base model (factorised variational distribution of inducing points) keeps most of the uncertainty in one of the layers (in this case the second) while the flow nearly collapses to a point estimate. Correlating the inducing points allows the model to distribute the uncertainty across the two layers providing a hierarchical decomposition of the uncertainty. Using the optimal inducing points, however, provides a wide range of possible warps without compromising on the overall quality of the fit to the observed data.

The discussion of hierarchical uncertainty relates to the previous work on GP priors with uncertain inputs [Girard et al., 2003], where the uncertainty in the

inputs $\mathbf{x}$ leads to higher uncertainty in the outputs $f(\mathbf{x})$. Analogously, if the layers of the hierarchical model are not correlated, then the uncertainty in the first layer (or any intermediate layer) plays the role of input noise and may lead to a higher uncertainty in the outputs.

7.3 Capturing warping uncertainty

We now return to the discussion of the variational distribution introduced in Eq. 7.2, that allows us to model the dependencies between the flow and the output GP. This section consists of two parts. Firstly, we discuss variational distributions for compositions of a monotonic flow and a GP, which model the dependencies over the inducing points across the two layers, and avoid the collapsing behaviour demonstrated in the first row of Fig. 7-3. Secondly, we use such variational distributions in the alignment setting (i.e. in the setting of multiple compositions of a monotonic flow and a GP, each of them corresponding to one of the observed sequences, and all of them sharing the same second layer GP), and discuss how to correlate variational distributions across the flows for different sequences. Overall, these two steps allow us to construct a variational distribution which follows the factorisation of Eq. 7.2, and captures the warping uncertainty in the alignment.

7.3.1 Introducing dependencies between inducing points

Jointly Gaussian inducing points The first approach we consider is based on modelling the inducing points in the first layer (monotonic flow) and in the second one (GP) with a joint Gaussian variational distribution. Namely, we assume $q(\mathbf{U}^g, \mathbf{U}^f) \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$. Unlike the case of factorised inducing points, we cannot analytically integrate out $\mathbf{U}^g$ and $\mathbf{U}^f$ in Eq. 7.8 under the assumption of a correlated Gaussian distribution of the inducing points. To overcome this limitation, we estimate the expectations over $\mathbf{U}^g$ and $\mathbf{U}^f$ by sampling, resulting in the following procedure to compute the lower bound in Eq. 7.8:

1. Draw N_s samples of inducing points $\{(\mathbf{U}_s^g, \mathbf{U}_s^f)\}_{s=1}^{N_s} \sim q(\mathbf{U}^g, \mathbf{U}^f)$.

2. For each pair $(\mathbf{U}_s^g, \mathbf{U}_s^f)$ of drawn inducing points:

(a) Draw an output $S^s(\mathbf{x}) \sim p\left(S(\mathbf{x}) \,|\, \mathbf{U}_s^{\mathrm{g}}\right)$ of the monotonic flow using the SDE numerical integration [Hegde et al., 2019],

(b) Draw $\mathbf{f}_s \sim p\left(\mathbf{f} \,|\, S^s(\mathbf{x}), \mathbf{U}_s^f\right)$, an evaluation of the second layer GP at $S^s(\mathbf{x})$ as inputs.

3. Empirically estimate the expectation in Eq. 7.8 as $\frac{1}{N_s}\sum_{s=1}^{N_s} \log p\left(\mathbf{y} \,|\, \mathbf{f}_s\right)$.

4. The KL-divergence in Eq. 7.8 can be computed analytically since both $q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f)$ and $p(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f)$ are jointly Gaussian.

The advantage of this approach of introducing dependencies between the inducing points in the flow and in the second layer GP is that it can be used within the variational inference framework discussed in Sec. 7.2.2 without any conceptual changes. Moreover, the estimation of the expectations over the inducing points in Eq. 7.8 by sampling means that a jointly Gaussian distribution $q(\mathbf{U}^{\mathrm{g}}, \mathbf{U}^f)$ can be replaced by any other distribution as long as we can sample from it.

In the second row of Fig. 7-3 we show an example fit of a flow-GP composition with jointly Gaussian variational distribution over the inducing points. We can see that such a model indeed captures more warping uncertainty than the one with factorised inducing points (*i.e.* the samples of the flow warp and the second layer GP in the second row of the figure are more diverse than those in the first row). However, it is also evident that a significant amount of warping uncertainty is not captured by this model as the independent samples from the composition are still quite similar to each other.

One problem with correlating the layers via their inducing points is that they are only proxies of the computations performed by the flow and the GP, rather than distributions over the actual outputs of the layers. Put differently, conditioned on the inducing points, the monotonic flow and the second layer GP are independent (as can be seen in the sampling procedure above), however both layers are not deterministic transformations given the inducing points (*e.g.* multiple draws from the monotonic flow with the same set of inducing points are different, albeit similar). This means that conditioned on the inducing points, compositions of every draw from the flow and every draw of the second layer GP must fit the observations. Since such draws are conditionally independent, this results in an underestimated warping uncertainty (following the same argument as in Sec. 7.1.2).

The argument above suggests that underestimation of the warping uncertainty observed in the second row of Fig. 7-3 is a result of a general approach of propagating such uncertainty through the correlated inducing points, rather than a consequence of a

particular way in which they are correlated (*e.g.* using a jointly Gaussian distribution). Next, we discuss an alternative approach of introducing correlations between the output of the flow and the second layer GP which alleviates the underestimation of the warping uncertainty.

Direct dependency between flow output and output GP Our goal is to directly correlate the transformation implemented by the second layer GP with the output of the monotonic flow. More specifically, we are looking for the following factorisation of the variational distribution over the inducing points and the outputs of the layers:

$$q(\mathbf{U}^{\mathrm{g}}, S(\mathbf{x}), \mathbf{U}^{f}, \mathbf{f}) = q(\mathbf{U}^{\mathrm{g}})p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))p(\mathbf{f} \mid \mathbf{U}^{f}, S(\mathbf{x})). \tag{7.11}$$

Put in words, we would like to find the distribution $q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))$ of the inducing points of the second layer GP which correspond to a GP mapping the flow outputs $S(\mathbf{x})$ to the observations $\mathbf{y}$. The main difference from the jointly Gaussian inducing points discussed above is that here we directly correlate the the second layer GP with the flow output, rather than correlating it with the flow inducing points (which leads to an underestimation of the warping uncertainty as discussed above).

To do so, we use the result of [Titsias, 2009, Eq. (10)], who derived the optimal variational distribution of the inducing points in a sparse GP given the fixed inputs and observations. In our case this result exactly gives the form of $q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))$ allowing us to define all terms in Eq. 7.11, with $q(\mathbf{U}^{\mathrm{g}}) \sim \mathcal{N}(\mathbf{m}^{\mathrm{g}}, \mathbf{S}^{\mathrm{g}})$ being a free-from Gaussian. The likelihood lower bound in this case is very similar to the one in Eq. 7.8, with the modifications reflecting the factorisation of Eq. 7.11. Specifically, the likelihood lower bound is as follows:

$$\begin{aligned}
\log p(\mathbf{y} \mid \mathbf{x}) &\geq \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}, S(\mathbf{x}), \mathbf{U}^{f}, \mathbf{f})}\left[\log \frac{p(\mathbf{U}^{\mathrm{g}})\cancel{p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})}p(\mathbf{U}^{f})\cancel{p(\mathbf{f} \mid \mathbf{U}^{f}, S(\mathbf{x}))}p(\mathbf{y} \mid \mathbf{f})}{q(\mathbf{U}^{\mathrm{g}})\cancel{p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})}q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))\cancel{p(\mathbf{f} \mid \mathbf{U}^{f}, S(\mathbf{x}))}}\right] \\
&= \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}}, S(\mathbf{x}), \mathbf{U}^{f}, \mathbf{f})}[\log p(\mathbf{y} \mid \mathbf{f})] \\
&\quad - \mathbb{E}_{q(\mathbf{U}^{\mathrm{g}})p(S(\mathbf{x}) \mid \mathbf{U}^{\mathrm{g}})q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))}\left[\log \frac{q(\mathbf{U}^{f} \mid \mathbf{y}, S(\mathbf{x}))}{p(\mathbf{U}^{f})}\right] \\
&\quad - \mathrm{KL}[\, q(\mathbf{U}^{\mathrm{g}}) \,||\, p(\mathbf{U}^{\mathrm{g}}) \,].
\end{aligned} \tag{7.12}$$

The last term (KL divergence between two Gaussian distributions) can be computed analytically, while the other two terms can be estimated by sampling. The inference procedure is as follows:

1. Sample N_s outputs of the flow $\{S^s(\mathbf{x})\}_{s=1}^{N_s} \sim q(\mathbf{U}^g)p(S(\mathbf{x}) \mid \mathbf{U}^g)$. $\mathbf{U}^g$ can be integrated out analytically in the flow GP following [Hegde et al., 2019].

2. For each of the S sampled flow outputs, analytically compute

$$\mathrm{KL}\Big[\, q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x})) \,||\, p(\mathbf{U}^f) \,\Big] = \mathbb{E}_{q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x}))}\left[\log \frac{q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x}))}{p(\mathbf{U}^f)}\right], \quad (7.13)$$

   the KL divergence between two Gaussian distributions (recall that $q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x}))$ is Gaussian given by Eq. (10) in [Titsias, 2009]). Averaging them, we obtain an empirical estimate of the second term in Eq. 7.12:

$$\mathbb{E}_{q(\mathbf{U}^g)p(S \mid \mathbf{U}^g)q(\mathbf{U}^f \mid \mathbf{y}, S(\mathbf{x}))}\left[\log \frac{q(\mathbf{U}^f \mid \mathbf{y}, S(\mathbf{x}))}{p(\mathbf{U}^f)}\right] \approx \frac{1}{N_s}\sum_{s=1}^{N_s} \mathrm{KL}\Big[\, q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x})) \,||\, p(\mathbf{U}^f) \,\Big]. \quad (7.14)$$

3. For each of the S sampled flow outputs, draw the inducing points of the GP from $\mathbf{U}_s^f \sim q(\mathbf{U}^f \mid \mathbf{y}, S^s(\mathbf{x}))$, and the GP outputs from $\mathbf{f}^s \sim p(\mathbf{f} \mid \mathbf{U}_s^f, S^s(\mathbf{x}))$. Compute the empirical estimate of the first term in Eq. 7.12 as

$$\mathbb{E}_{q(\mathbf{U}^g, S(\mathbf{x}), \mathbf{U}^f, \mathbf{f})}[\log p(\mathbf{y} \mid \mathbf{f})] \approx \frac{1}{N_s}\sum_{s=1}^{N_s} \log p(\mathbf{y} \mid \mathbf{f}^s). \quad (7.15)$$

In the third row of Fig. 7-3 we show an example of a flow-GP composition fitted by maximising the lower bound in Eq. 7.12, the estimate of which is obtained using the procedure above. We can see in this figure that such a fit captures significantly more uncertainty in both of the layers than the fits corresponding to factorised inducing points and the jointly Gaussian inducing points.

Having discussed the propagation of warping (or compositional) uncertainty in compositions of the monotonic flow and GP, next we return to the alignment setting where we consider the case of multiple such compositions sharing the second layer GP and fitted simultaneously.

7.3.2 Correlating flows across multiple compositions

In the simplest case where there is only one group of sequences, the alignment model consists of modelling the observed sequences as compositions of a monotonic warp (different for each sequence) with the same GP (latent function). In Sec. 7.1.2 we argue

that in this case the warps for different sequences are dependent, and hence it is not sufficient to propagate uncertainty within the flow-GP compositions for each sequence separately – it is also necessary to propagate the uncertainty across the sequences.

The monotonic flow warp for each sequence is modelled using the inducing points $\{\mathbf{U}^{g_j}\}_{j=1}^{J}$ (where index j refers to one of J observed sequences). To correlate the flow warps across sequences, we define the variational distribution over the inducing points across sequences to be jointly Gaussian: $q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J}) \sim \mathcal{N}(\mathbf{m}^{g}, \mathbf{S}^{g})$. We modify the variational distribution for a single flow-GP composition (Eq. 7.11) to encode the factorisation of Eq. 7.2, obtaining the following joint variational distribution for J compositions of the correlated monotonic flow warps and the second layer GPs:

$$\begin{aligned} q(\{\mathbf{U}^{g_j}\},\{S^j(\mathbf{x})\},\{\mathbf{U}^{f_j}\},\{\mathbf{f}_j\}) &= q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J}) \\ &\times \prod_{j=1}^{J} p(S^j(\mathbf{x}) \mid \mathbf{U}^{g_j}) q(\mathbf{U}^{f_j} \mid \mathbf{y}_j, S^j(\mathbf{x})) p(\mathbf{f}_j \mid \mathbf{U}^{f_j}, S^j(\mathbf{x})). \end{aligned} \tag{7.16}$$

The corresponding likelihood lower bound in this case is obtained similarly to Eq. 7.12 (for conciseness omitting the terms that cancel out):

$$\begin{aligned} \log p(\mathbf{Y} \mid \mathbf{x}) &\geq \mathbb{E}_{q(\{\mathbf{U}^{g_j}\},\{S^j(\mathbf{x})\},\{\mathbf{U}^{f_j}\},\{\mathbf{f}_j\})} \left[\log \frac{\prod_{j=1}^{J} p(\mathbf{U}^{g_j}) p(\mathbf{U}^{f_j}) p(\mathbf{y}_j \mid \mathbf{f}_j)}{q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J}) \prod_{j=1}^{J} q(\mathbf{U}^{f_j} \mid \mathbf{y}_j, S^j(\mathbf{x}))} \right] \\ &= \mathbb{E}_{q(\{\mathbf{U}^{g_j}\})} \left[\sum_{j=1}^{J} \mathbb{E}_{q(S^j(\mathbf{x}), \mathbf{U}^{f_j}, \mathbf{f}_j)} [\log p(\mathbf{y}_j \mid \mathbf{f}_j)] \right] \\ &\quad - \mathbb{E}_{q(\{\mathbf{U}^{g_j}\})} \left[\sum_{j=1}^{J} \mathbb{E}_{q(p(S^j(\mathbf{x}) \mid \mathbf{U}^{g_j}) q(\mathbf{U}^{f_j} \mid \mathbf{y}_j, S^j(\mathbf{x}))} \left[\log \frac{q(\mathbf{U}^{f_j} \mid \mathbf{y}_j, S^j(\mathbf{x}))}{p(\mathbf{U}^{f_j})} \right] \right] \\ &\quad - \mathrm{KL}[\, q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J}) \,||\, p(\mathbf{U}^{g_1}) \ldots p(\mathbf{U}^{g_J}) \,]\,. \end{aligned} \tag{7.17}$$

Conditioned on the flow inducing points $\{\mathbf{U}^{g_j}\}$, the other variables in Eq. 7.16 are conditionally independent across the sequences. This suggests a simple modification of the sampling estimate of Eq. 7.12 to estimate the lower bound in Eq. 7.17. Namely, the procedure is as follows:

1. Sample N_s draws of the inducing points for each of the flow warps:

$$\{(\mathbf{U}_s^{g_1}, \ldots, \mathbf{U}_s^{g_J})\}_{s=1}^{N_s} \sim q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J}).$$

2. Conditioned on these draws, use the steps 1 — 5 of the estimator for Eq. 7.12 for each of the J sequences to obtain estimates of the first two terms in Eq. 7.17.

3. Compute the third term in Eq. 7.17 analytically as a KL-divergence between two Gaussians.

7.4 Group assignment uncertainty in alignments

In this section we discuss the problem of modelling the group assignment uncertainty in alignments. This corresponds to the scenario previously discussed in Fig. 7-2 where there exist two possible explanations for the data, both of which are plausible under our priors. As discussed in Sec. 7.1.2, to capture such uncertainty the joint variational distribution must allow for dependencies between the flow warps and the group assignments for the sequences. Our goal in this section is to formally define the problem, introduce an example of a potential variational distribution, formulate the inference procedure using such a distribution, and finally discuss the limitations of the proposed approach. While we do not provide an extensive treatment of the problem of the group assignment uncertainty, we highlight the main challenges and set directions for future work.

Problem setting Let us denote the group assignment of each latent function f_j (for which we have a corresponding observed sequence $\mathbf{y}_j$) as $c_j \in \{1, \dots, K\}$ (assuming there are K possible assignment options). For instance, Fig. 7-2 illustrates a case where there are two possible group assignments ($K = 2$). The latent functions of all curves in that figure apart from the orange one are assigned to the same group, while the latent function of the orange curve might be assigned to one of the three different groups.

The alignment constraint in this setting means that the two latent functions assigned to the same group must be identical. This is similar to the mixture model objective discussed in Ch. 4, however, here we aim to capture the uncertainty about the assignments $\{c_j\}$ (which collapse to point estimates in Ch. 4). This similarity to the mixture model objective implies that for fixed groups assignments (*e.g.* for each draw from the variational posterior over $\{c_j\}$), we can impose the same alignment constraint (objective) on the latent functions $\{f_j\}$ as in Ch. 4, which we elaborate on further later in this section.

The group assignments $\{c_j\}$ of the latent functions induce the group assignments of the corresponding warps (because the warps and the latent functions are dependent to fit the observations as discussed in Sec. 7.1 and 7.3). However, unlike the latent functions

the warps assigned to the same groups are not identical (if the observed sequences are different, but the latent functions are the same, the warps must be different). In the rest of this section, our goal is to (1) introduce a joint variational distribution correlating the group assignments and the warps (and hence the latent functions through the warps), and (2) impose an alignment objective on the latent functions to constrain them to be consistent with the group assignments.

Joint distribution of groups assignments and warps To introduce the dependencies between the group assignments $\{c_j\}$ and the warps (modelled using the inducing points $\{\mathbf{U}^{g_j}\}$), we need to define the joint distribution $q(\{c_j\}, \{\mathbf{U}^{g_j}\})$. Since it should be multi-modal to capture the settings similar to the one in Fig. 7-2, it is natural to define $q(\mathbf{U}^{g_j})$ as a mixture with c_j being the assignments to one of K mixture components. Specifically, we define $q(c_j) \sim \text{Categorical}(K)$, $q(\mathbf{U}^{g_j} \mid c_j = k) \sim \mathcal{N}(\mathbf{m}_j^k, \mathbf{S}_j^k)$, and hence $q(\mathbf{U}^{g_j}) \sim \sum_{k=1}^{K} q(c_j = k)\mathcal{N}(\mathbf{m}_j^k, \mathbf{S}_j^k)$. The joint distribution over multiple sequences can be defined independently over each sequence as

$$q(\{\mathbf{U}^{g_j}\}, \{c_j = k_j\}) = \prod_{j=1}^{J} q(c_j = k_j) q(\mathbf{U}^{g_j} \mid c_j = k_j) \sim \prod_{j=1}^{J} q(c_j = k_j)\mathcal{N}(\mathbf{m}_j^{k_j}, \mathbf{S}_j^{k_j}). \tag{7.18}$$

Such variational distribution of the warping inducing points $\mathbf{U}^g$ factorises over j which implies that we break the dependence between the warps for the different sequences. This is a simplifying assumption that we make in order to make inference easier and test the effect of the uncertainty in the group assignments in isolation from the estimation of the warping uncertainty. As a consequence of this modelling choice, we expect to see less warping uncertainty within each of the mixture components, similarly to the factorised distribution of the warps $q(g_1, \ldots, g_j) = q(g_1) \ldots q(g_J)$.

We combine Eq. 7.16 and Eq. 7.18 to obtain the joint variational distribution of all variables in the flow-GP compositions for each sequence with the corresponding group assignments:

$$\begin{aligned} q(\{c_j\}, \{\mathbf{z}_j\}, \{\mathbf{U}^{g_j}\}, \{S^j(\mathbf{x})\}, \{\mathbf{U}^{f_j}\}, \{\mathbf{f}_j\}) = \prod_{j=1}^{J} q(c_j) q(\mathbf{U}^{g_j} \mid c_j) \\ \times \prod_{j=1}^{J} p(S_j(\mathbf{x}) \mid \mathbf{U}^{g_j}) q(\mathbf{U}^{f_j} \mid \mathbf{Y}_j, S_j(\mathbf{x})) p(\mathbf{f}_j \mid \mathbf{U}^{f_j}, S_j(\mathbf{x})). \end{aligned} \tag{7.19}$$

The likelihood lower bound is obtained similarly to Eq. 7.17 (for conciseness omitting the terms that cancel out):

$$
\begin{aligned}
\log p(\mathbf{Y} \mid \mathbf{x}) &\geq \mathbb{E}_{q(\{c_j\},\{\mathbf{U}^{g_j}\},\{S_j(\mathbf{x})\},\{\mathbf{U}^{f_j}\},\{\mathbf{f}_j\})}\left[\log \frac{\prod_{j=1}^{J} p(c_j)p(\mathbf{U}^{g_j})p(\mathbf{U}^{f_j})p(\mathbf{Y}_j \mid \mathbf{f}_j)}{\prod_{j=1}^{J} q(\mathbf{U}^{f_j} \mid \mathbf{Y}_j, S_j(\mathbf{x}))q(c_j)q(\mathbf{U}_k^{g_j} \mid c_j)}\right] \\
&= \sum_{j=1}^{J} \mathbb{E}_{q(c_j)q(\mathbf{U}^{g_j} \mid c_j)q(S_j(\mathbf{x}) \mid \mathbf{U}^{g_j})q(\mathbf{U}^{f_j} \mid \mathbf{Y}_j, S_j(\mathbf{x}))p(\mathbf{f}_j \mid \mathbf{U}^{f_j}, S_j(\mathbf{x}))}[\log p(\mathbf{Y}_j \mid \mathbf{f}_j)] \\
&\quad - \sum_{j=1}^{J} \mathbb{E}_{q(c_j)q(\mathbf{U}^{g_j} \mid c_j)q(S_j(\mathbf{x}) \mid \mathbf{U}^{g_j})q(\mathbf{U}^{f_j} \mid \mathbf{Y}_j, S_j(\mathbf{x}))}\left[\log \frac{q(\mathbf{U}^{f_j} \mid \mathbf{Y}_j, S_j(\mathbf{x}))}{p(\mathbf{U}^{f_j})}\right] \\
&\quad - \sum_{j=1}^{J} \mathbb{E}_{q(c_j)q(\mathbf{U}^{g_j} \mid c_j)}\left[\log \frac{q(\mathbf{U}^{g_j} \mid c_j)}{p(\mathbf{U}^{g_j})}\right] \\
&\quad - \sum_{j=1}^{J} \mathrm{KL}[\, q(c_j) \,||\, p(c_j) \,] \,.
\end{aligned}
\tag{7.20}
$$

Inference Consider each term in the lower bound of Eq. 7.20:

1. To estimate the first term, for each sequence j, sample N_s draws of inducing points from a mixture distribution by drawing a cluster assignment c_j according to the probabilities $q(c_j)$ (using a reparameterisation of the categorical distribution based on a continuous approximation [Jang et al., 2017]), and then drawing a sample from the corresponding Gaussian component $q(\mathbf{U}_k^{g_j} \mid c_j = k)$. Using these samples, estimate the term as in Eq. 7.15.

2. Estimate the second term as in Eq. 7.14.

3. The third term is the expectation (taken with respect to $q(c_j)$) of a KL-divergence between two Gaussian distributions for each of the J sequences. Conditioned on the N_s samples of the cluster assignments $\{c_j^s\}$ from step 1, estimate the term as:

$$
\sum_{j=1}^{J} \mathbb{E}_{q(c_j)q(\mathbf{U}^{g_j} \mid c_j)}\left[\log \frac{q(\mathbf{U}^{g_j} \mid c_j)}{p(\mathbf{U}^{g_j})}\right] \approx \sum_{j=1}^{J}\left[\frac{1}{N_s}\sum_{s=1}^{N_s} \mathrm{KL}[\, q(\mathbf{U}^{g_j} \mid c_j^S) \,||\, p(\mathbf{U}^{g_j}) \,]\right].
$$

4. The fourth term is the sum of the KL-divergences between Categorical distributions and is available in closed form. As the default prior we consider a Categorical distribution with event probabilities equal to $1/K$.

Alignment objective In the previous chapters we discussed two probabilistic alignment objectives, GP-LVM (Sec. 3.4) and the mixture model (Sec. 4.3). Given the model of the sequences as described above, the mixture model objective arises naturally. The reason for that is that it models the group assignments explicitly (so we can directly perform inference over c_j), while the GP-LVM does that implicitly (via the latent space representations $\mathbf{z}_j$), requiring us to consider explicit group assignments in the latent space introducing an additional level of complexity.

Specifically, for each draw $(c_1, \ldots, c_J) \sim q(c_1) \ldots q(c_J)$ of the cluster assignments, we compute the pairwise distances between the evaluations of latent functions at fixed inputs:

$$\mathcal{L}_{\text{align}} = \sum_{k=1}^{K} \sum_{i<j} \mathbb{1}(c_i = c_j = k) \, ||f_i(\mathbf{x}) - f_j(\mathbf{x})||^2, \tag{7.21}$$

with $\mathbb{1}(c_i = c_j = k) = 1$ if $c_i = c_j = k$ and zero otherwise. The alignment constraint in Eq. 7.21 is subtracted from the likelihood lower bound in Eq. 7.20 to obtain a final optimisation objective.

The pairwise distances in Eq. 7.21 essentially correspond to a simplified mixture model objective (Sec. 4.3) with each mixture component having the same variance. In general, the pairwise distances can be replaced with Gaussian densities for each of the components, or with any other function computing the similarity between the latent functions assigned to the same group.

7.5 Alignment experiments

In this section we present the experiments illustrating the methods for capturing warping (Sec. 7.3) and group assignment (Sec. 7.4) uncertainties.

We consider a toy data set that consists of three warped sequences of 50 data points. The observed sequences are generated using $\mathbf{y}_j = \text{sinc}(\pi \, g_j(\mathbf{x})) + \varepsilon_j$ with $\varepsilon_j \sim \mathcal{N}(0, 0.1)$.

Warping uncertainty In the first experiment we compare the alignment results on this data set using the original model discussed in Ch. 3, and its extensions aimed at capturing the warping uncertainty discussed in Sec. 7.3. The results are given in Fig. 7-4. The first column shows the input data, the second column shows the alignments obtained using the original model (Ch. 3) while the third and fourth columns show the results

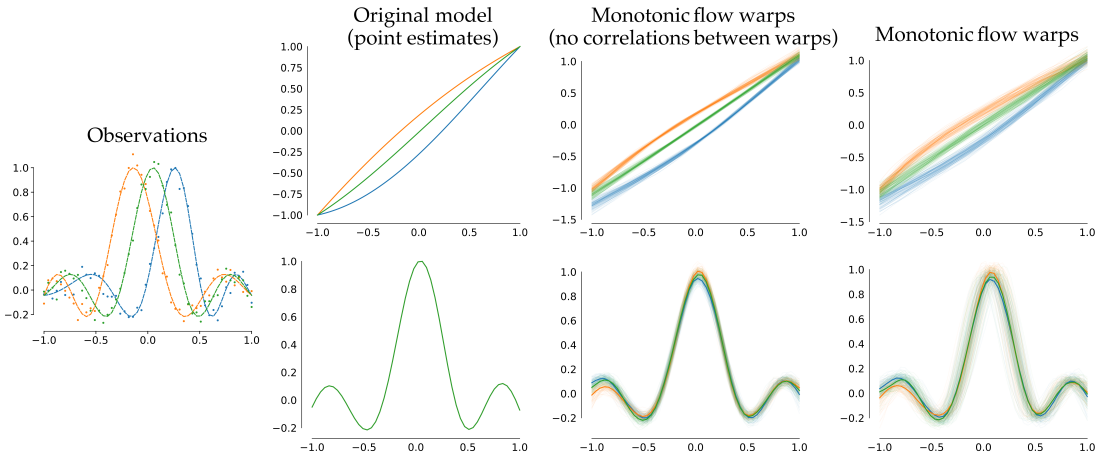
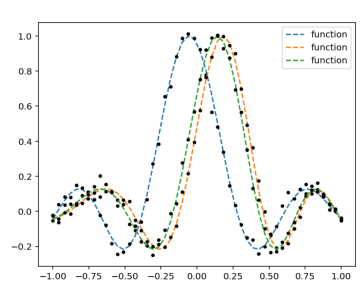


Figure 7-4: Given noisy observations of warped sequences, we compare the uncertainty in the warps for the proposed model (with and without correlations) and for the model from Ch. 3.

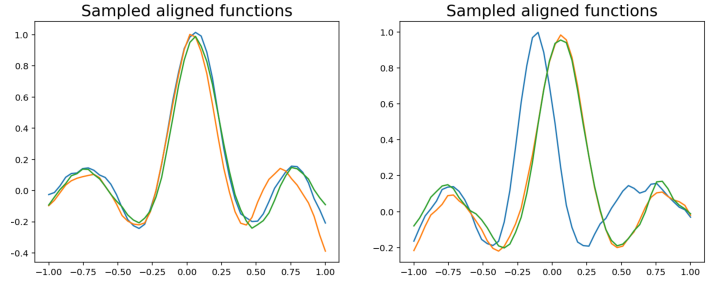
obtained using the variational distributions with correlations between the flow warps g_j and the latent functions f_j (Sec. 7.3). In the third column the variational distribution of the flow inducing points factorises over the three sequences while in the fourth one it is jointly Gaussian (*i.e.* the factor $q(\mathbf{U}^{g_1}, \ldots, \mathbf{U}^{g_J})$ in Eq. 7.16 is either factorised or assumed to be jointly Gaussian).

The original GP-GPLVM model, which does not model the dependencies between the warps and the second layer GPs, provides only a point estimate of the warps and the latent functions, as argued in Sec. 7.1.2. Including such dependencies in the joint variational distribution results in a distribution of alignments capturing the inherent warping uncertainty (third column in Fig. 7-4). Additionally correlating the variational distributions of the flows across the sequences (fourth column in Fig. 7-4) allows us to capture even more warping uncertainty, which is in agreement with our argument in Sec. 7.3.2.

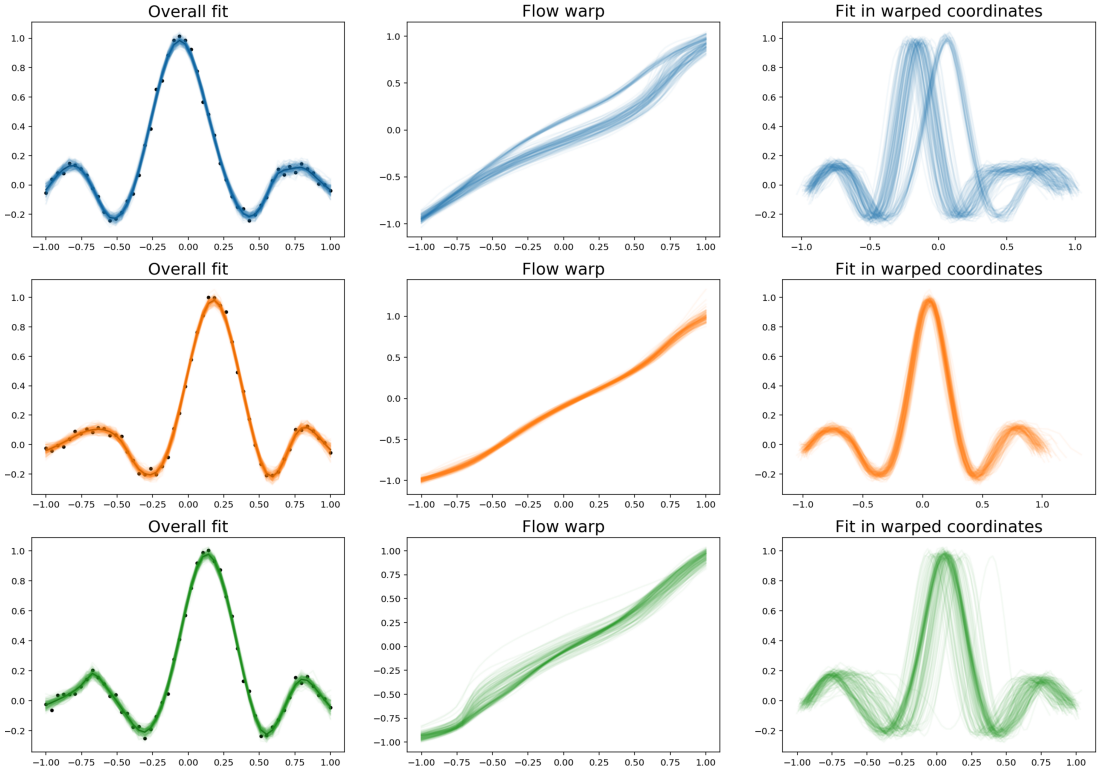
Group assignment uncertainty Next, we discuss a numerical example of the uncertainty in group assignments. We consider a toy data set in Fig. 7-5a, which consists of three sequences, two of which (orange and green) are very similar to each other while the third sequence (blue) is less similar to the other two. All of these sequences can be aligned, however, the warp that separates the blue sequence from the other two sequences is less likely under our prior which encourages warps that are close to an identity. Therefore, there exist two possible solutions: (1) all three sequences are aligned or (2) two of the sequences (green and orange) are aligned while the third sequence



(a) Observations.



(b) Examples of sampled aligned functions.



(c) Fitted sequences (left), estimated warps (middle) and fits in the warped coordinates (right) for the 3 sequences.

Figure 7-5: Illustration of uncertainty in warps and cluster assignments. When the warps and the cluster assignment are allowed to be bi-modal, the model captures two possible solutions: one that assigns all sequences to a single cluster and aligns them within that cluster, and another solution that favours the model with two separate clusters. This can be seen in the fit in warped coordinates figure for the blue curve where the majority of the samples are assigned to one cluster (which corresponds to aligning the blue function to its own cluster, as shown on the right in Fig. (b)), while a small subset is assigned to a new cluster (which corresponds to all sequences being aligned together, as shown on the left in Fig. (b)).

(blue) is not aligned (and hence forms its own cluster) but is modelled using an identity warp which is more likely under the warping prior[1].

In Sec. 7.4 we discussed a method capturing such uncertainty in group assignments by finding solutions consistent with multiple possible group assignments rather then collapsing to a single solution. For this experiment we use the variational distribution in Eq. 7.18 with $K = 2$ mixture components, corresponding to the setting in which each sequence can be potentially assigned to one of the two groups. That allows for bi-modal warps and consequently for bi-modal compositions $\{f_j(g_j(\mathbf{x}))\}$. We can see in Fig. 7-5c that such an approach indeed captures the uncertainty in the group assignment of the blue sequence represented through the bi-modal compositions of the flow warp and the latent function. In Fig. 7-5b we show two draws from the variational distribution of the latent functions, where we see that the green and the orange curves got aligned together (*i.e.* they have the same latent function) while the blue one is either assigned to the same group as the other two sequences or forms its own group.

While this example demonstrates some of the behaviours that can be interpreted as group assignment uncertainty, it leaves many questions unanswered. Primarily, the inference in this model is complex due to all the additional correlations (as explained in Sec. 7.4) and it relies heavily on sampling, making it slow, cumbersome and difficult to implement.

7.6 Discussion

In this chapter we revisited the alignment models of Ch. 3 and 4 and discussed how to define every component of these models probabilistically to capture the uncertainties arising in the alignment task. We built on the monotonic flow model introduced in Ch. 6 to construct two-layer compositions in which the first layer is a monotonic warp while the second layer is a standard GP. We discussed variational inference methods in these compositions, and specifically, the inference methods capturing the inherent compositional uncertainty in such two-layer models. Next, we applied these inference techniques to the alignment problem, and illustrated their utility in capturing the warping uncertainty in the compositions of warps and latent functions. In the second

[1]Note that the latent functions f_j are modelled using a GP with the stationary SE kernel, and the stationarity of the kernel introduces a prior that may compete with the warping prior (which encourages identity warps), *i.e.* if the observed data is very non-stationary, then it might be beneficial to use the warping functions to alter the inputs so that the data modelled by the latent function is stationary, as explained in Sec. 5.3. This effect is consistent with the alignment objective in this example, and therefore we do not consider it in isolation.

part of the chapter, we discussed the group assignment uncertainty in the context of alignments, proposed a variational inference method capturing such uncertainty and illustrated its behaviour with a toy example.

We argue that the uncertainty estimates provide a more informative model of the data and help uncover structures in the data which are not captured by the point estimates used in the previous chapters. One limitation of the inference schemes in Sec. 7.3 and 7.4 aimed at capturing warping and group assignment uncertainties is that they fall short of capturing both types of uncertainties concurrently. However, while we separate the two types of uncertainty for illustrative and practical purposes, there is no fundamental reason to consider them separately. Furthermore, another reason for considering these uncertainties separately is that as the complexity of the variational distributions increases, the inference becomes more complex and prone to local minima. Specifically, special care needs to be taken when choosing the number of samples to be taken when estimating the expectations, and existing literature of nested estimators needs to be consulted [Rainforth et al., 2019].

While in this chapter we concentrated our attention on a compositional model with a monotonic flow as the first layer, other monotonic regression approaches can also be utilised as the warping function in a two-layer compositional model. We have explored the option of using a transformed GP [Andersen et al., 2018] as the monotonic warp in a two-layer deep GP. However, our empirical results suggest that such a composition is very sensitive to the initialisation and suffers from poor local minima during optimisation. Furthermore, imposing priors (such as the preference for an identity warp) in this model is not straightforward. We provide a further discussion of this compositional model and some results on a toy problem in the Appendix A.1.

Most of the existing work on hierarchical GP models focuses on predictive uncertainty rather than compositional uncertainty, or uncertainty about the function implemented by each individual layer [Damianou and Lawrence, 2013, Damianou, 2015]. The inference in such models is typically performed using mean-field variational approximations. The correlated variational distributions discussed in this chapter offer one way to overcome the limitations of the mean-field models. Another notable exception to the mean-field inference is the work on stochastic gradient Hamiltonian Monte Carlo [Havasi et al., 2018] as an inference scheme, that recognises the issue of compositional uncertainty, highlighting the fact that most of the existing (variational) approaches to inference are limited to estimating unimodal posterior distributions when the true distribution is often multi-modal. This directly relates to the group assignment uncertainty in the alignment task as discussed in Sec. 7.4. As inference using MC estimates is typically

very costly, [Havasi et al., 2018] note that it is beneficial to decouple the model in terms of the inducing points for the mean and the variance. This results in a highly non-convex optimization problem that requires careful parameterisation to improve the stability of convergence [Havasi et al., 2018]. A toy example of a two-layer model fitted using SGHMC is given in Appendix A.2. Unfortunately, we found the poor stability of the model to be prohibitive even on toy examples.

Deep Gaussian processes offer one possible way of constructing hierarchical generative models. They allow us to model complex functions (*e.g.* with non-Gaussian marginal distributions) using compositions of standard Gaussian processes. This is a common theme in deep learning in general, which deals with combining simple functions into complex hierarchical models, capable of representing extremely complicated functions (and data sets). In other words, the hierarchical structure of the model is used to increase the capacity of the model.

Deep learning comprises of a myriad of different approaches [Goodfellow et al., 2016], including some that are considered to be Bayesian in nature [Gal, 2016] and are closely related to GPs [Neal, 1996, Lee et al., 2018]. While Bayesian deep learning aims to perform Bayesian inference in deep neural networks, most deep learning approaches only compute the point estimates of the neural network weights. The reason for this is that in practice the point estimates often suffice, while the uncertainty in intermediate levels of the hierarchy is hard to represent, quantify and interpret. Meanwhile, the issues discussed in this chapter are an example of a situation which requires estimates of uncertainty in intermediate layers and use of priors in intermediate layers that correspond to the prior beliefs about the hierarchical structure of the data (as is the case in the alignment model). Furthermore, it highlights the shortcomings of the standard approximations in variational inference, in particular, the need for multi-modal variational distributions [Lawrence, 2000]. All these points are interesting directions for future work.

8

Final conclusions and future work

This dissertation focuses on the problem of the temporal alignment of time series data and the uncertainties in the resulting compositional models. We identify three constituent parts of the alignment problem: the model of the warps, the model of the latent functions and the alignment objective, which lead to the main questions investigated in this thesis: (1) the formulation of a probabilistic alignment objective, (2) the investigation of the compositional models of two functions, and (3) the analysis of the uncertainties present in the compositional models of alignments.

Probabilistic alignment objectives

We focused on the scenario in which the data set may contain multiple groups of sequences where the sequences within each group come from the same latent function but are evaluated at unknown warped inputs. We further assumed that the group assignments are unknown a priori. Therefore, the alignment objective should encode the assumption that the simplest way to explain the observations is to align them within groups. This motivated the study of probabilistic alignment objectives, such as the GP-LVM and the BMM, that can help increase the automation of the alignment task. The GP-LVM objective (introduced in Ch. 3) provides a low-dimensional latent representation of the aligned sequences, which implicitly encodes the group assignments. Based on the qualitative and quantitative experiments given in Sec. 3.7, it performs reliably in practice. Meanwhile, the mixture model objective (presented in Ch. 4) provides explicit group assignments but the specification of the model and the inference in it have proven to be problematic. Originally, these objectives are formulated as regularisers on the aligned sequences rather than a joint probabilistic model of the observations. As discussed in

Ch. 5, defining the alignment models proposed in Ch. 3 and 4 as a joint probabilistic model is complicated by the fact that while the aligned sequences are not observed, they need to be treated as such in order to keep the dependence between the alignment objective and the model of the observations. As an alternative solution, the alignment model was reformulated as a two-layer multi-output GP (Sec. 5.2) which provides a natural framework for the alignment task when the data contains a single latent function. The extension of that framework to the case of multiple underlying functions is possible, however, it requires assumptions which might be too simplistic for modelling any real data. Finally, we note that such alignment objectives may be applied to a variety of different data sets characterised by the fact that the inputs are perturbed using some operation that can be parametrised (for example, a cyclic shift or a rotation). This offers a way to design new clustering models that are invariant to a chosen operation.

Compositional models of two functions

The model for the alignment task are naturally formulated as a composition of two functions: the (temporal) warping and the latent function. In the Bayesian formulation of this composition we place priors on each of the two constituent functions individually. Constructing a nonparametric model satisfying the constraint (or the prior belief) that the warping function needs to be monotonic requires designing a random process such that each sample from this process is guaranteed to be monotonic. The monotonic GP flow (introduced in Ch. 6) offers one possible formulation of such a process. The latent functions (which correspond to the second layer in the two-layer model) are modelled using GPs with a kernel that represents our prior belief about the characteristics of the latent (aligned) functions.

Uncertainties in alignments

While a standard (single layer) GP is often treated as the gold standard of uncertainty quantification in models of time series [Foong et al., 2019], the two-layer hierarchical structure of the alignment model requires a non-trivial extension of the GP framework in order to capture the uncertainties that are of interest in the alignment model. The two-layer structure of the warps and the latent functions can be interpreted as a deep GP with an additional monotonicity constraint on the first layer.

In the existing literature of deep GPs the uncertainty is typically considered mostly in the context of extrapolations, *i.e.* the typical question is whether the model generalises

well (in terms of providing reasonable uncertainty estimates) in parts of the input domain with no observations. Meanwhile, the uncertainty quantification in the alignment model involves finding all possible compositions of the warps and the latent functions that simultaneously explain the observations and that identify such latent functions that are optimal under the alignment objective.

Consequently, quantifying the uncertainty in the alignment model necessitates a reformulation of the standard inference scheme used in deep GP models, as explained in Ch. 7. Traditionally, variational distributions over a set of variables are designed in such a way that would allow tractable inference over that set of variables. In deep GPs such variables are the inducing points in each layer and tractable variational inference requires imposing strong independence assumptions between the layers of the model resulting in the inputs to each layer being independent of the outputs of the previous layer [Damianou, 2015]. Reintroducing some of the correlations in the variational approximations of [Damianou, 2015] results again in intractable integrals, which can then be approximated by sampling [Salimbeni and Deisenroth, 2017]. However, in Ch. 7 we argue that the factorisation of the variational distribution proposed in [Salimbeni and Deisenroth, 2017] is still not sufficient to preserve the uncertainty in the context of alignments and an appropriate inference scheme should be designed with this application in mind. In models of alignments our goal is to uncover the combinations of the warps and the latent functions which are plausible under our priors (while we are less concerned with the generalisation capacity of our compositional model outside of the input domain). Specifically, this means introducing correlations between the two functions in the compositions of warps and latent functions, and among the compositions themselves.

8.1 Future work

The discussions and results presented in this dissertation suggest multiple avenues for future work, some of which were mentioned in the preceding chapters. In this section we revisit and summarise some of these directions for future work and put them into a broader context.

Industrial applications

As discussed in Ch 1, the alignment problem arises naturally in the domain of character animations and procedural content generation. The temporal alignments of recorded

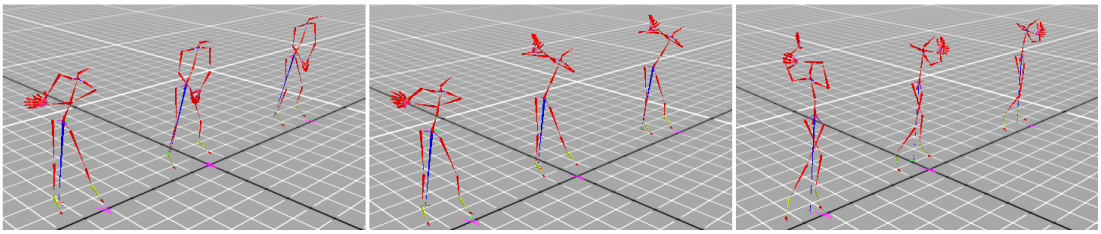


Figure 8-1: Alignment of motion capture data (golf swing); the three figures show three different frames in a sequence. The characters on the left and in the middle correspond to the two input sequences while the rightmost character shows the version of the first sequence (leftmost) aligned to the second sequence (middle).

animation sequences are essential for the proper transitional blending between animation clips. For example, Fig. 8-1 shows the alignment of two motion capture clips of golf swings before and after temporal alignment. In order to generate a new sequence that stylistically combines the two given sequences into a new one, it is essential to make sure that the sequences are synchronised in terms of timing, and this can be done using the approach defined in Ch. 3. Furthermore, the identification of the clusters in the low-dimensional latent space allows us to determine if it is possible to blend the clips with one another or if they capture fundamentally different motions and hence cannot be blended together. While the GP-LVM latent space offers one possible solution for blending between animation clips to generate novel ones, further work needs to be done to integrate the preferred solution into the existing animation framework used by the artists, keeping the workflows intuitive, simple to use and consistent with other editing tools such as inverse kinematics solvers [Grochow et al., 2004]. Additional thought might need to be given to the computational complexity of the alignment models, as depending on the size of the data set, it might be prohibitive for real-time applications. Sparse GP approximations for the modelling of the sequences address this issue to some extent but additional approximations or batch processing is needed to scale the model to large data sets.

Alignment and multi-output GPs (MOGP)

Extensions of MOGP alignment From a modelling perspective, the MOGP approach of Sec. 5.2 offers a more principled formulation of the alignment model than the ones presented in Ch. 3 and 4. However, in its current form it can recognise multiple groups of sequences only under certain simplifying assumptions (primarily same length-scale for all sequences), and it comes at a price of higher computational complexity in comparison to the approaches introduced in Ch. 3 and 4. Both of these concerns offer

directions for future work.

Links to current models One may also further explore the link between the regulariser formulation of Ch. 3 and 4 and the MOGP model introduced in Sec. 5.2. One question is whether the regulariser formulation may be used as an approximation to some other MOGP models (other than the alignment model) where the two covariances (over the inputs in each sequence and over all sequences) change at each iteration and the covariance matrix has the structure as defined in Sec. 5.2.

Compositional uncertainty in deep GPs

In Ch. 7 we formulate the problem of estimating compositional uncertainty (*i.e.* the uncertainty that refers to the fact that different compositions of functions may explain the data well and all be plausible under our priors) and lay some foundations in terms of choosing appropriate inference schemes. However, many aspects of the problem require further attention both in the context of alignments and for other models of compositional GPs.

Technical challenges The proposed variational approximations contain elements of variational inference as well as sampling, where the latter is necessary in order to maintain some of the correlations between the layers in the hierarchies and among the compositions that correspond to each of the observed sequences. Such a formulation leads to nested sampling that contributes to the poor scalability and the unstable optimisation that is prone to getting stuck in local minima. Consequently, testing and comparing different models and inference schemes can be challenging.

Significance of compositional uncertainty The quantification of uncertainty in compositional models remains largely an open question. Beyond the alignment model, the compositional uncertainty is rarely considered in hierarchical models [Havasi et al., 2018]. This poses the question of the importance of compositional uncertainty and the implications of using such inference schemes that fail to capture this uncertainty. It is possible that most applications do not require the explicit estimation of uncertainty in each of the layers of the hierarchy, as long as the uncertainty is quantified sufficiently well when generalising to new inputs. However, it is worth investigating which applications might require a careful treatment of the compositional uncertainty and it is not unreasonable to ask how the compositional uncertainty affects the generalisation capacity of

hierarchical models of GPs. Similar considerations might be of interest in the context of Bayesian deep learning, where capturing the epistemic uncertainty (the uncertainty related to the choice of the model given the limited data) is also challenging [Wilson, 2019].

Monotonic constraints on intermediate layers in deep GPs

Many of the existing deep GP models rely on careful initialisations in order to aid convergence [Salimbeni and Deisenroth, 2017]. One of the solutions suggested in the literature is including a linear mean function for the intermediate layers in the hierarchy [Duvenaud et al., 2014, Salimbeni and Deisenroth, 2017] to try to avoid the pathological case of highly non-injective intermediate layers, as this leads to some layers collapsing to a point. Therefore, it might be worth investigating in what circumstances the monotonicity constraints are appropriate as a replacement for the linear mean function. If we assume that the only limitation to modelling any one-dimensional time-series data with a GP is due to the stationarity of the standard choices of the kernel functions, then a two-layer deep GP with the first layer constrained to be monotonic is sufficient to model any such data. However, monotonicity is a property of one-dimensional functions only, hence it may not be appropriate for data sets with more than one dimension (unless it makes sense to constrain each dimension to be monotonic). Finally, one may question the utility of deep GPs as models for problems where each individual layer has no understandable and explainable meaning. It may be argued that a Bayesian neural networks offer a simpler, faster and more scalable tool [Gal, 2016], and deep GPs should only be used when the problem exhibits a compositional structure so that meaningful priors (over functions) can be imposed on each layer in the hierarchy. Further research is needed to establish the best practices when designing and using complex hierarchical Bayesian models.

Optimisation

For the work covered in this dissertation, relatively little care has been taken in terms of choosing appropriate techniques and tools for the optimisation of the various objective functions. More specifically, we have relied on out-of-the-box stochastic gradient-based optimisers (primarily, Adam [Kingma and Ba, 2014] in Tensorflow). While some of the models described in this dissertation appear to converge without any issues (for example, the MAP estimate based alignment objective of Ch. 3), the others may benefit from choosing the optimisation tools more carefully. This is particularly true in the case of the

monotonic flow of Ch. 6 and the compositional models of Ch. 7. One promising direction discussed in [Hensman et al., 2013] is the use of natural gradients. More generally, with more complex models and inference schemes, the importance of optimisation only increases, and new tools may need to be designed with the specific applications (such as variational methods) in mind.

8.2 Final remark

> *"White elephant – A possession that is useless or troublesome, especially one that is expensive to maintain or difficult to dispose of* [1] *"*
>
> – Oxford English Dictionary

The utility of an ML system is measured by its ability to extract useful knowledge from the data. The definition of useful knowledge, though, is often ambiguous. In recent times, the availability of larger data sets has motivated the creation of complex models with a large number of parameters that achieve high prediction accuracy. However, highly complex models may suffer from reduced interpretability of the decisions made by the resulting ML system [Gilpin et al., 2018] and poor ability to generalise to unseen inputs and data set shifts [Quionero-Candela et al., 2009], which may yield the system impractical. While further efforts are needed to address these and other issues with ML frameworks, it is our hope that the work presented in this dissertation has provided convincing justification for the proposed approaches to modelling and inference to be seen as a valuable addition to the existing literature on learning under uncertainty.

[1] Origin: "*From the story that the kings of Siam gave such animals as a gift to courtiers they disliked, in order to ruin the recipient by the great expense incurred in maintaining the animal.*" – Oxford English Dictionary.

Bibliography

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

[Álvarez et al., 2010] Álvarez, M., Luengo, D., Titsias, M., and Lawrence, N. D. (2010). Efficient multioutput gaussian processes through variational inducing kernels. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Álvarez et al., 2012] Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266.

[Andersen et al., 2018] Andersen, M. R., Siivola, E., Riutort-Mayol, G., and Vehtari, A. (2018). A non-parametric probabilistic model for monotonic functions. *"All Of Bayesian Nonparametrics" Workshop at NeurIPS*.

[Anirudh et al., 2015] Anirudh, R., Turaga, P., Su, J., and Srivastava, A. (2015). Elastic Functional Coding of Human Actions: From Vector-Fields to Latent Variables. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Antoniak, 1969] Antoniak, C. E. (1969). *Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems*. PhD thesis, University of California, Los Angeles (USA).

[Baisero et al., 2015] Baisero, A., Pokorny, F. T., and Ek, C. H. (2015). On a Family of Decomposable Kernels on Sequences. *arXiv preprint:1501.06284v1*.

[Barber, 2012] Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

[Bauer et al., 2016] Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Ben-David, 2018] Ben-David, S. (2018). Clustering - what both theoreticians and practitioners are doing wrong. In *Conference on Artificial Intelligence (AAAI)*.

[Bentley et al., 2011] Bentley, P., Nordehn, G., Coimbra, M., and Mannor, S. (2011). The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results. http://www.peterjbentley.com/heartchallenge.

[Bernardo and Smith, 2007] Bernardo, J. and Smith, A. (2007). *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley.

[Berndt and Clifford, 1994] Berndt, D. J. and Clifford, J. (1994). Using Dynamic Time Warping to Find Patterns in Time Series. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.

[Bietti and Chizat, 2014] Bietti, A. and Chizat, L. (2014). Inference in Dirichlet Process Mixtures with Applications to Text Document Clustering. http://alberto.bietti.me/files/dpmixtures.pdf.

[Bilionis et al., 2013] Bilionis, I., Zabaras, N., Konomi, B. A., and Lin, G. (2013). Multi-output separable gaussian process: Towards an efficient, fully bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212–239.

[Bishop et al., 1997] Bishop, C., Svensén, M., and Williams, C. (1997). Gtm: A principled alternative to the self-organizing map. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Blei and Jordan, 2005] Blei, D. M. and Jordan, M. I. (2005). Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144.

[Bonilla et al., 2008] Bonilla, E. V., Chai, K. M., and Williams, C. (2008). Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Bornkamp and Ickstadt, 2009] Bornkamp, B. and Ickstadt, K. (2009). Bayesian nonparametric estimation of continuous monotone functions with applications to dose-response analysis. *Biometrics*, 65(1):198–205.

[Broderick, 2014] Broderick, T. (2014). *Clusters and Features from Combinatorial Stochastic Processes*. PhD thesis, University of California, Berkeley, USA.

[Campbell and Kautz, 2014] Campbell, N. D. F. and Kautz, J. (2014). Learning a Manifold of Fonts. *ACM Transactions on Graphics*, 33(4).

[Canini et al., 2016] Canini, K., Cotter, A., Gupta, M., Milani Fard, M., and Pfeifer, J. (2016). Fast and flexible monotonic functions with ensembles of lattices. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Chen et al., 2018] Chen, R., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Clarke et al., 1993] Clarke, F. H., Stern, R. J., and Wolenski, P. R. (1993). Subgradient criteria for monotonicity, the lipschitz condition, and convexity. *Canadian Journal of Mathematics*, 45(6):1167–1183.

[Cui et al., 2014] Cui, Z., Chang, H., Shan, S., and Chen, X. (2014). Generalized Unsupervised Manifold Alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Curtis and Ghosh, 2011] Curtis, S. M. and Ghosh, S. K. (2011). A variable selection approach to monotonic regression with bernstein polynomials. *Journal of Applied Statistics*, 38(5):961–976.

[Cuturi, 2011] Cuturi, M. (2011). Fast Global Alignment Kernels. In *The International Conference on Machine Learning (ICML)*.

[Cuturi et al., 2007] Cuturi, M., Vert, J. P., Birkenes, O., and Matsui, T. (2007). A Kernel for Time Series Based on Global Alignments. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[da Silva et al., 2008] da Silva, M., Abe, Y., and Popović, J. (2008). Interactive simulation of stylized human locomotion. *ACM Transactions on Graphics (TOG)*.

[Da Veiga and Marrel, 2012] Da Veiga, S. and Marrel, A. (2012). Gaussian process modeling with inequality constraints. *Annales de la Faculté des sciences de Toulouse : Mathématiques*.

[Damianou, 2015] Damianou, A. (2015). *Deep Gaussian Processes and Variational Propagation of Uncertainty.* PhD thesis, University of Sheffield (UK).

[Damianou and Lawrence, 2013] Damianou, A. and Lawrence, N. (2013). Deep gaussian processes. In *The International Conference on Artificial Intelligence and Statistics (AISTATS).*

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

[Dette and Scheder, 2006] Dette, H. and Scheder, R. (2006). Strictly monotone and smooth nonparametric regression for two or more variables. *Canadian Journal of Statistics*, 34(4):535–561.

[Dryden and Mardia, 2016] Dryden, I. L. and Mardia, K. V. (2016). *Statistical Shape Analysis, with Applications in R. Second Edition.* John Wiley and Sons.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* John Willey & Sons, New Yotk.

[Duncker and Sahani, 2018] Duncker, L. and Sahani, M. (2018). Temporal alignment and latent gaussian process factor inference in population spike trains. In *Advances in Neural Information Processing Systems (NeurIPS).*

[Durot and Lopuhaä, 2018] Durot, C. and Lopuhaä, H. (2018). Limit theory in monotone function estimation. *Statistical Science*, 33(4):547–567.

[Dutilleul, 1999] Dutilleul, P. (1999). The mle algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, 64(2):105–123.

[Duvenaud et al., 2014] Duvenaud, D., Rippel, O., Adams, R., and Ghahramani, Z. (2014). Avoiding pathologies in very deep networks. In *The International Conference on Artificial Intelligence and Statistics (AISTATS).*

[Dyke et al., 2013] Dyke, H., Vixie, K., and Asaki, T. (2013). Cone monotonicity: Structure theorem, properties, and comparisons to other notions of monotonicity. *Abstract and Applied Analysis*, 2013.

[Ernst et al., 2014] Ernst, O. G., Sprungk, B., and Starkloff, H.-J. (2014). Bayesian inverse problems and kalman filters. In Dahlke, S., Dahmen, W., Griebel, M., Hackbusch, W., Ritter, K., Schneider, R., Schwab, C., and Yserentant, H., editors, *Extraction of Quantifiable Information from Complex Systems*, pages 133–159. Springer.

[Ferguson, 1973] Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2):209–230.

[Fischhoff, 2003] Fischhoff, B. (2003). Hindsight ≠ foresight: The effect of outcome knowledge on judgment under uncertainty. *Quality & Safety in Health Care*, 12:304–311.

[Florovsky, 1969] Florovsky, G. (1969). The study of the past. *Ideas of History. Vol. II: The critical philosophy of history*, pages 351–369.

[Foong et al., 2019] Foong, A. Y. K., Burt, D. R., Li, Y., and Turner, R. E. (2019). Pathologies of factorised gaussian and mc dropout posteriors in bayesian neural networks. *arXiv preprint:1909.00719*.

[Gal, 2016] Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

[Garreau et al., 2014] Garreau, D., Lajugie, R., Arlot, S., and Bach, F. (2014). Metric Learning for Temporal Sequence Alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Gilpin et al., 2018] Gilpin, L., Testart, C., Fruchter, N., and Adebayo, J. (2018). Explaining explanations to society. In *Workshop on Ethical, Social and Governance Issues in AI at NeurIPS*.

[Girard et al., 2003] Girard, A., Rasmussen, C. E., Candela, J. Q., and Murray-Smith, R. (2003). Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Golchi et al., 2015] Golchi, S., Bingham, D., Chipman, H., and Campbell, D. (2015). Monotone emulation of computer experiments. *SIAM-ASA Journal on Uncertainty Quantification*, 3(1):370–392.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[Gower, 1975] Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.

[Grochow et al., 2004] Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. (2004). Style-based Inverse Kinematics. In *ACM SIGGRAPH*.

[Guilbault et al., 2004] Guilbault, R. L., Bryant, F. B., Brockway, J. H., and Posavac, E. J. (2004). A meta-analysis of research on hindsight bias. *Basic and Applied Social Psychology*, 26(2-3):103–117.

[Hall and Huang, 2001] Hall, P. and Huang, L.-S. (2001). Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, 29(3):624–647.

[Haslett and Parnell, 2008] Haslett, J. and Parnell, A. (2008). A simple monotone process with application to radiocarbon-dated depth chronologies. *Journal of the Royal Statistical Society. Series C*, 57:399–418.

[Havasi et al., 2018] Havasi, M., Hernández-Lobato, J. M., and Murillo-Fuentes, J. J. (2018). Inference in deep gaussian processes using stochastic gradient hamiltonian monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Hegde et al., 2019] Hegde, P., Heinonen, M., Lähdesmäki, H., and Kaski, S. (2019). Deep learning with differential gaussian process flows. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Heinonen et al., 2018] Heinonen, M., Yildiz, C., Mannerström, H., Intosalmi, J., and Lähdesmäki, H. (2018). Learning unknown ode models with gaussian processes. In *The International Conference on Machine Learning (ICML)*.

[Hensman et al., 2013] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

[Hjort et al., 2010] Hjort, N., Holmes, C., Mueller, P., and Walker, S. (2010). *Bayesian Nonparametrics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

[Hsu et al., 2005] Hsu, E., Pulli, K., and Popović, J. (2005). Style translation for human motion. *ACM Transactions on Graphics (TOG)*, 24(3):1082–1089.

[James et al., 2011] James, V. H., Guntupalli, J. S., Connolly, A. C., Halchenko, Y. O., Conroy, B. R., Gobbini, M. I., Hanke, M., and Ramadge, P. J. (2011). A Common, High-Dimensional Model of the Representational Space in Human Ventral Temporal Cortex. *Neuron*, 72(2):404–416.

[Jang et al., 2017] Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *The International Conference on Learning Representations (ICLR)*.

[Kac, 1949] Kac, M. (1949). On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):1–13.

[Kaiser et al., 2018] Kaiser, M., Otte, C., Runkler, T., and Ek, C. H. (2018). Bayesian alignments of warped multi-output gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Keogh and Pazzani, 2001] Keogh, E. J. and Pazzani, M. J. (2001). Derivative Dynamic Time Warping. In *SIAM International Conference on Data Mining*.

[Kersting et al., 2007] Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2007). Most likely heteroscedastic gaussian process regression. In *The International Conference on Machine Learning (ICML)*.

[Kim et al., 2018] Kim, D., Ryu, H., and Kim, Y. (2018). Nonparametric bayesian modeling for monotonicity in catch ratio. *Communications in Statistics: Simulation and Computation*, 47(4):1056–1065.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations (ICLR)*.

[Kloeden and Platen, 1992] Kloeden, P. and Platen, E. (1992). The numerical solution of stochastic differential equations. In *Stochastic Modelling and Applied Probability*, volume 23. Springer.

[Koehler, 1991] Koehler, D. (1991). Explanation, imagination, and confidence in judgment. *Psychological bulletin*, 110:499–519.

[Kurtek et al., 2012] Kurtek, S., Srivastava, A., Klassen, E., and Ding, Z. (2012). Statistical Modeling of Curves using Shapes and Related Features. *Journal of the American Statistical Association*, 107(499):1152–1165.

[Kurtek et al., 2011] Kurtek, S., Srivastava, A., and Wu, W. (2011). Signal Estimation Under Random Time-warpings and Nonlinear Signal Alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Lab, 2016] Lab, C. M. G. (2016). Motion Capture Database . "http://mocap.cs.cmu.edu/info.php.

[Larsen et al., 2006] Larsen, F., Van Den Berg, F., and Engelsen, S. (2006). An exploratory chemometric study of 1h nmr spectra of table wines. *Journal of Chemometrics*, 20(5).

[Lavine and Mockus, 1995] Lavine, M. and Mockus, A. (1995). A nonparametric bayes method for isotonic regression. *Journal of Statistical Planning and Inference*, 46(2).

[Lawrence et al., 2002] Lawrence, N., Seeger, M., and Herbrich, R. (2002). Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Lawrence, 2000] Lawrence, N. D. (2000). *Variational Inference in Probabilistic Models*. PhD thesis, Cambridge University.

[Lawrence, 2005] Lawrence, N. D. (2005). Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research (JMLR)*, 6:1783–1816.

[Lázaro-Gredilla, 2012] Lázaro-Gredilla, M. (2012). Bayesian Warped Gaussian Processes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Lee et al., 2018] Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., and Sohl-dickstein, J. (2018). Deep neural networks as gaussian processes. *ICLR*.

[Lenk and Choi, 2017] Lenk, P. and Choi, T. (2017). Bayesian analysis of shape-restricted functions using gaussian process priors. *Statistica Sinica*, 27(1):43–69.

[Levine et al., 2012] Levine, S., Wang, J. M., Haraux, A., Popović, Z., and Koltun, V. (2012). Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (TOG)*.

[Lin and Dunson, 2014] Lin, L. and Dunson, D. (2014). Bayesian monotone regression using gaussian process projection. *Biometrika*, 101(2):303–317.

[Listgarten et al., 2005] Listgarten, J., Neal, R. M., Roweis, S. T., and Emili, A. (2005). Multiple Alignment of Continuous Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Liu et al., 2018] Liu, H., Cai, J., and Ong, Y.-S. (2018). Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102 – 121.

[Lopez-Lopera et al., 2019] Lopez-Lopera, A. F., John, S., and Durrande, N. (2019). Gaussian process modulated cox processes under linear inequality constraints. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Lorbert and Ramadge, 2012] Lorbert, A. and Ramadge, P. J. (2012). Kernel Hyperalignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Lorenzi and Filippone, 2018] Lorenzi, M. and Filippone, M. (2018). Constraining the dynamics of deep probabilistic models. In *The International Conference on Machine Learning (ICML)*.

[Lorenzi et al., 2019] Lorenzi, M., Filippone, M., Frisoni, G., Alexander, D., and Ourselin, S. (2019). Probabilistic disease progression modeling to characterize diagnostic uncertainty: Application to staging and prediction in alzheimer's disease. *NeuroImage*, 190:56–68.

[Louizos and Welling, 2016] Louizos, C. and Welling, M. (2016). Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors. *arXiv preprint:1603.04733*.

[Maatouk, 2017] Maatouk, H. (2017). Finite-dimensional approximation of gaussian processes with inequality constraints. *arXiv preprint:1706.02178*.

[Maatouk and Bay, 2016] Maatouk, H. and Bay, X. (2016). A New Rejection Sampling Method for Truncated Multivariate Gaussian Random Variables Restricted to Convex Sets. In *Monte Carlo and Quasi-Monte Carlo Methods*, volume 163 of *Springer Proceedings in Mathematics & Statistics*. Springer International Publishing.

[MacKay, 1995] MacKay, D. (1995). Bayesian neural networks and density networks. *Nuclear Inst. and Methods in Physics Research, A*, 354(1):73–80.

[Marron et al., 2015] Marron, J., Ramsay, J., Sangalli, L., and Srivastava, A. (2015). Functional data analysis of amplitude and phase variation. *Statistical Science*, 30(4):468–484.

[Matsumoto et al., 2005] Matsumoto, H., Yor, M., et al. (2005). Exponential functionals of brownian motion, i: Probability laws at fixed time. *Probability surveys*, 2:312–347.

[McCall et al., 2012] McCall, C., Reddy, K. K., and Shah, M. (2012). Macro-class selection for hierarchical k-nn classification of inertial sensor data. In *PECCS*.

[Müller, 2007] Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[Nader et al., 2019] Nader, C. A., Ayache, N., Robert, P., and Lorenzi, M. (2019). Monotonic gaussian process for spatio-temporal trajectory separation in brain imaging data. *arXiv preprint:1902.10952*.

[Neal, 1996] Neal, R. M. (1996). *Priors for Infinite Networks*. Springer New York.

[Øksendal, 1992] Øksendal, B. (1992). *Stochastic Differential Equations (3rd Ed.): An Introduction with Applications*. Springer-Verlag.

[Park and Choi, 2010] Park, S. and Choi, S. (2010). Hierarchical gaussian process regression. In *Proceedings of Asian Conference on Machine Learning*.

[Quinonero Candela and Rasmussen, 2005] Quinonero Candela, J. and Rasmussen, C. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research (JMLR)*.

[Quionero-Candela et al., 2009] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. The MIT Press.

[Rainforth et al., 2019] Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. (2019). On nesting monte carlo estimators. *Proceedings of Machine Learning Research*, 80.

[Raket et al., 2016] Raket, L. L., Grimme, B., Schöner, G., Igel, C., and Markussen, B. (2016). Separating timing, movement conditions and individual differences in the analysis of human movement. *PLOS Computational Biology*.

[Ramsay, 1988] Ramsay, J. (1988). Monotone regression splines in action. *Statistical Science*, 3(4):425–441.

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

[Rasmussen and Ghahramani, 2000] Rasmussen, C. E. and Ghahramani, Z. (2000). Occam's razor. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Rasmussen and Williams, 2005] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[Riihimäki and Vehtari, 2010] Riihimäki, J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *Proceedings of Machine Learning Research*, pages 645–652.

[Salimbeni and Deisenroth, 2017] Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

[Shively et al., 2009] Shively, T. S., Sager, T. W., and Walker, S. G. (2009). A bayesian approach to nonparametric monotone function estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(1):159–175.

[Shoemake, 1985] Shoemake, K. (1985). Animating rotation with quaternion curves. *SIGGRAPH 1985*.

[Siivola et al., 2016] Siivola, E., Piironen, J., and Vehtari, A. (2016). Automatic monotonicity detection for gaussian processes. *arXiv preprint:1610.05440*.

[Sill and Abu-Mostafa, 1997] Sill, J. and Abu-Mostafa, Y. (1997). Monotonicity hints. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Snelson and Ghahramani, 2006] Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Snelson et al., 2004] Snelson, E., Ghahramani, Z., and Rasmussen, C. E. (2004). Warped Gaussian Processes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Snoek et al., 2014] Snoek, J., Swersky, K., Zemel, R., and Adams, R. (2014). Input warping for bayesian optimization of non-stationary functions. In *The International Conference on Machine Learning (ICML)*.

[Solin and Särkkä, 2019] Solin, A. and Särkkä, S. (2019). Hilbert space methods for reduced-rank gaussian process regression. *STATISTICS AND COMPUTING*.

[Srivastava et al., 2011] Srivastava, A., Wu, W., Kurtek, S., Klassen, E., and Marron, J. S. (2011). Registration of Functional Data Using Fisher-Rao Metric. *arXiv preprint:1103.3817*.

[Srivastava et al., 2018] Srivastava, A., Wu, W., Kurtek, S., Klassen, E., and Marron, J. S. (2018). Elastic Functional Data Analysis. http://ssamg.stat.fsu.edu/software.

[Stegle et al., 2011] Stegle, O., Lippert, C., Mooij, J. M., Lawrence, N. D., and Borgwardt, K. (2011). Efficient inference in matrix-variate gaussian models with iid observation noise. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Teh, 2010] Teh, Y. W. (2010). Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.

[Tipping and Bishop, 1999] Tipping, M. and Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 61(3):611–622.

[Titsias, 2009] Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574.

[Trigeorgis et al., 2016] Trigeorgis, G., Nicolaou, M. A., Zafeiriou, S., and Schuller, B. W. (2016). Deep Canonical Time Warping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Trigeorgis et al., 2017] Trigeorgis, G., Nicolaou, M. A., Zafeiriou, S., and Schuller, B. W. (2017). Deep Canonical Time Warping for Simultaneous Alignment and Representation Learning of Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

[Tucker et al., 2013] Tucker, J. D., Wu, W., and Srivastava, A. (2013). Generative Models for Functional Data using Phase and Amplitude Separation. *Computational Statistics and Data Analysis*, 61(Supplement C):50–66.

[Vu et al., 2012] Vu, H. T., Carey, C. J., and Mahadevan, S. (2012). Manifold Warping: Manifold Alignment over Time. In *National Conference on Artificial Intelligence*.

[Wahba, 1978] Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society. Series B*, 49.

[Wang et al., 2006] Wang, J. M., Fleet, D. J., and Hertzmann, A. (2006). Gaussian process dynamical models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Wilson, 2019] Wilson, A. G. (2019). The case for bayesian deep learning. *NYU Courant Technical Report*. https://cims.nyu.edu/~andrewgw/caseforbdl.pdf.

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

[Yildiz et al., 2018a] Yildiz, C., Heinonen, M., Intosalmi, J., Mannerstrom, H., and Lahdesmaki, H. (2018a). Learning stochastic differential equations with gaussian processes without gradient matching. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*.

[Yildiz et al., 2018b] Yildiz, C., Heinonen, M., and Lähdesmäki, H. (2018b). A nonparametric spatio-temporal SDE model. In *Spatiotemporal Workshop at NeurIPS*.

[Yu et al., 2009] Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Zhou, 2012] Zhou, F. (2012). Generalized Time Warping for Multi-modal Alignment of Human Motion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhou and de al Torre, 2016] Zhou, F. and de al Torre, F. (2016). Generalized Canonical Time Warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(2).

[Zhou and de la Torre, 2009] Zhou, F. and de la Torre, F. (2009). Canonical Time Warping for Alignment of Human Behavior. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Zhou and de la Torre, 2018] Zhou, F. and de la Torre, F. (2018). Software for Canonical Time Warping. http://www.f-zhou.com/ta_code.html.

Appendix A

Comparison of two-layer deep GP models

A.1 Comparison to transformed GP

As an alternative to the monotonic flow, discussed in Ch. 6, we have considered using transformed GPs [Andersen et al., 2018] in a two layer model where the first layer is modelled by the transformed GP (and hence is constrained to be monotonic), and the second layer is a standard GP.

The transformed GP model is defined as a solution $g(x)$ to a differential equation $g'(x) = v(x) \geq 0$ where $v : \mathbb{R} \to \mathbb{R}_+$ is a non-negative function. Defining this function as a composition of two functions, $v(\cdot) := t(h(\cdot))$, the solution to the differential equation is written as:

$$g(x) = g_0 + \int_a^x t(h(s))\mathrm{d}s \tag{A.1}$$

where $t : \mathbb{R} \to \mathbb{R}_+$ is a non-negative function and $h \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$. This approach models the derivative of the unknown function (rather than the function itself) with a GP. Since the modelled function is assumed to be monotonic, its derivative must have a constant sign, which is achieved by placing an arbitrary non-negative function $t(x)$ on top of the GP h.

Choosing the function $t(x)$ to be $t(x) = x^2$ and approximating $h(x)$ with a finite basis function expansion [Solin and Särkkä, 2019], allows us to compute an analytic estimate of the integral in Eq. A.1. Given a domain $x \in [-L, L]$ for some $L > 0$ and Dirichlet

boundary conditions on h, $h(-L) = h(L) = 0$, the solution of Eq. A.1 is approximated as:

$$g(x) \approx g_0 + \boldsymbol{\alpha}^T \boldsymbol{\psi}(x) \boldsymbol{\alpha} \quad \text{with } \alpha_j \sim \mathcal{N}\left(0, S_\theta(\lambda_j^{\frac{1}{2}})\right), \tag{A.2}$$

where $S_\theta(\cdot)$ is the spectral density of the stationary covariance function $k(\cdot, \cdot)$ with a set of hyperparameters θ, λ_j and ψ_j arise from the Hilbert space approximation of the covariance operator, and J denotes the number of terms in the approximation.

Eq. A.2 provides an approximation of a nonparametrically defined monotonic function $g(x)$ with a parametric model with parameters $\boldsymbol{\alpha}$ and g_0. The joint probability of these parameters is:

$$p(\mathbf{y}, \boldsymbol{\alpha}, g_0) = p(\mathbf{y} \mid \boldsymbol{\alpha}, g_0)\, p(\boldsymbol{\alpha})\, p(g_0) \tag{A.3}$$

where $p(g_0)$ is the prior on the intercept (the place where the function crosses the x-axis). The posterior distribution $p(\boldsymbol{\alpha}, g_0 \,|\, \mathbf{y})$ is not analytically tractable [Andersen et al., 2018].

As in Ch. 7, we are interested in a composition of two functions $f(g(x))$ where the $g(\cdot)$ is defined to be monotonic while $f(\cdot)$ is a standard zero-mean GP, $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$. We augment the output GP f with a set of M inducing points $\mathbf{U}^f = \{u^f\}_{i=1}^M$ with corresponding inducing locations $\mathbf{z} = \{z_i\}_{i=1}^M$. Denoting $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, g_0\}$, the joint distribution of the composition is:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{U}^f, \boldsymbol{\theta} \mid \mathbf{x}) = p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{U}^f, \boldsymbol{\theta}, \mathbf{x}) p(\mathbf{U}^f \mid \mathbf{z}) p(\boldsymbol{\theta}). \tag{A.4}$$

The second term is $p(\mathbf{U}^f \mid \boldsymbol{\theta}, \mathbf{x}) \sim \mathcal{N}\left(0, k(\boldsymbol{\alpha}^T \boldsymbol{\psi}(\mathbf{x}) \boldsymbol{\alpha} + g_0, \boldsymbol{\alpha}^T \boldsymbol{\psi}(\mathbf{x}) \boldsymbol{\alpha} + g_0)\right)$, and $\boldsymbol{\theta}$ appears in the kernel non-linearly, making the integration difficult. Defining the approximate posterior as:

$$q(\mathbf{f}, \mathbf{U}^f, \boldsymbol{\theta}) = p(\mathbf{f} \mid \mathbf{U}^f, g(\boldsymbol{\theta})) q(\mathbf{U}^f, \boldsymbol{\theta}), \tag{A.5}$$

the corresponding evidence lower bound is:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f}, \mathbf{U}^f, \boldsymbol{\theta})} \left[\log p(\mathbf{y} \mid \mathbf{f})\right] - \mathrm{KL}\Big[\, q(\mathbf{U}^f, \boldsymbol{\theta}) \,||\, p(\mathbf{U}^f)\, p(\boldsymbol{\theta}) \,\Big]. \tag{A.6}$$

The first term in $\mathcal{L}$ can be estimated by sampling [Salimbeni and Deisenroth, 2017], as discussed in Sec. 7.2.1. Furthermore, to allow for correlations between the parameters $\boldsymbol{\theta}$ of the first layer and the inducing points $\mathbf{U}^f$ of the outputs (which is necessary to capture compositional uncertainty, Sec. 7), we sample them from a joint Gaussian distribution.

More specifically, we

1. Draw N_s samples of inducing points and parameters $\boldsymbol{\theta}$, $\{(\mathbf{U}_s^f, \boldsymbol{\theta}_s)\}_{s=1}^{N_s} \sim q(\mathbf{U}^f, \boldsymbol{\theta})$.
2. For each pair $(\mathbf{U}_s^f, \boldsymbol{\theta}_s)$:
   (a) Estimate $g(\mathbf{x})$ using Eq. A.2,
   (b) Draw $\mathbf{f}_s \sim p\left(\mathbf{f} \,|\, \mathbf{U}_s^f, \boldsymbol{\theta}\right)$, an evaluation of the second layer GP at $g(\mathbf{x})$ as inputs, using the analytic expressions for the first two moments of the posterior distribution, see Eq. 7-8 in [Salimbeni and Deisenroth, 2017].
3. Empirically estimate the expectation in $\mathcal{L}$ as $\frac{1}{N_s} \sum_{s=1}^{N_s} \log p\left(\mathbf{y} \,|\, \mathbf{f}_s\right)$.
4. KL-divergence can be calculated analytically for (jointly) Gaussian distributions $q(\mathbf{U}^f, \boldsymbol{\theta})$ and $p(\mathbf{U}^f)p(\boldsymbol{\theta})$.

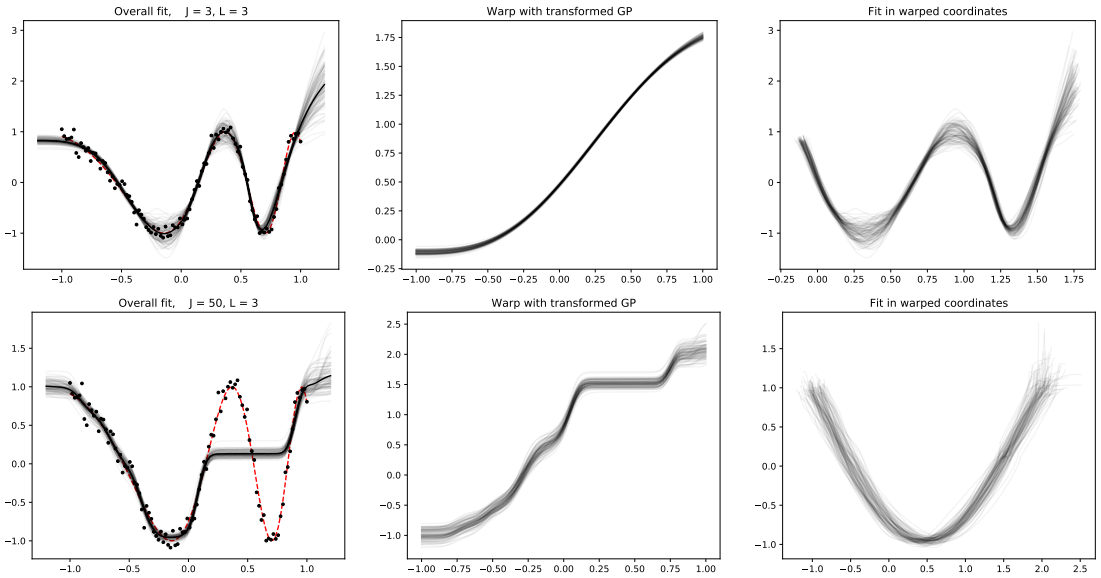


Figure A-1: A two-layer model fitted using transformed GP [Andersen et al., 2018] and an output GP (similar to the model defined in Sec. 7.3 in the main part of the thesis). From left to right, we plot the fitted function, the first layer of the model, and the fitted function in the warped coordinates. The rows correspond to different values of J, the number of terms in the approximation of the kernel.

Experiments

Let us revisit the example of the chirp function previously shown in Fig. 7-3. The results for the two-layer model that combines the transformed GP and a standard GP are shown

in Fig. A-1 (the parameter values are $J \in [3, 50]$, that correspond to a very small and a very large number of basis functions in the approximation, and $L = 3$ that agrees with the recommendation provided in [Andersen et al., 2018]).

We found this model to be less stable than the two-layer model with monotonicity constraints discussed in Sec. 7.2.2, both with and without the correlations between the two layers (corresponding to jointly Gaussian or factorised $q(\mathbf{U}^f, \boldsymbol{\theta})$). Furthermore, we observed that this model is often unable to fit the chirp function well, and when it does fit the function well, it finds a solution with almost no uncertainty in the first layer. This might be due to the poor local minima which become harder to avoid as the number of parameters in the GP approximation increases. One way to improve the performance of the model is through careful initialisation which is complicated by the use of the approximation of the covariance function of the GP. Note that the monotonic flow of Ch. 6 is naturally initialised to an identity function if the flow vector field is close to zero.

A.2 Comparison to SGHMC

In a recent paper, Havasi *et al.* [Havasi et al., 2018] discuss the issue of the intermediate layers in a DGP collapsing to a single mode (as previously discussed in Ch. 7). Their Hamiltonian Monte Carlo-based stochastic inference scheme is able to estimate non-Gaussian posteriors, and can capture multi-modality and estimate uncertainty in the intermediate layers of a DGP. In order to compare the uncertainty estimates of our two-layer setting, we fitted a two-layer DGP model to the data identical to the one in Fig. 7-3 and in A-1 using the implementation provided by the authors. Fig. A-2 shows the fitted model for different random initialisations of the SGHMC. We note that the first layer, referred to as the warp, is not constrained, unlike in our method. The output shown in the top row is comparable to the results produced by our method. However, the other outputs have a short lengthscale in both layers and thus seem to overfit. In these experiments we use the parameter values from the original paper (in particular, `num posterior samples = 100`, `posterior sample spacing = 50`), and reduce the learning rate to 0.001 (increasing the number of iterations by a factor of 50).

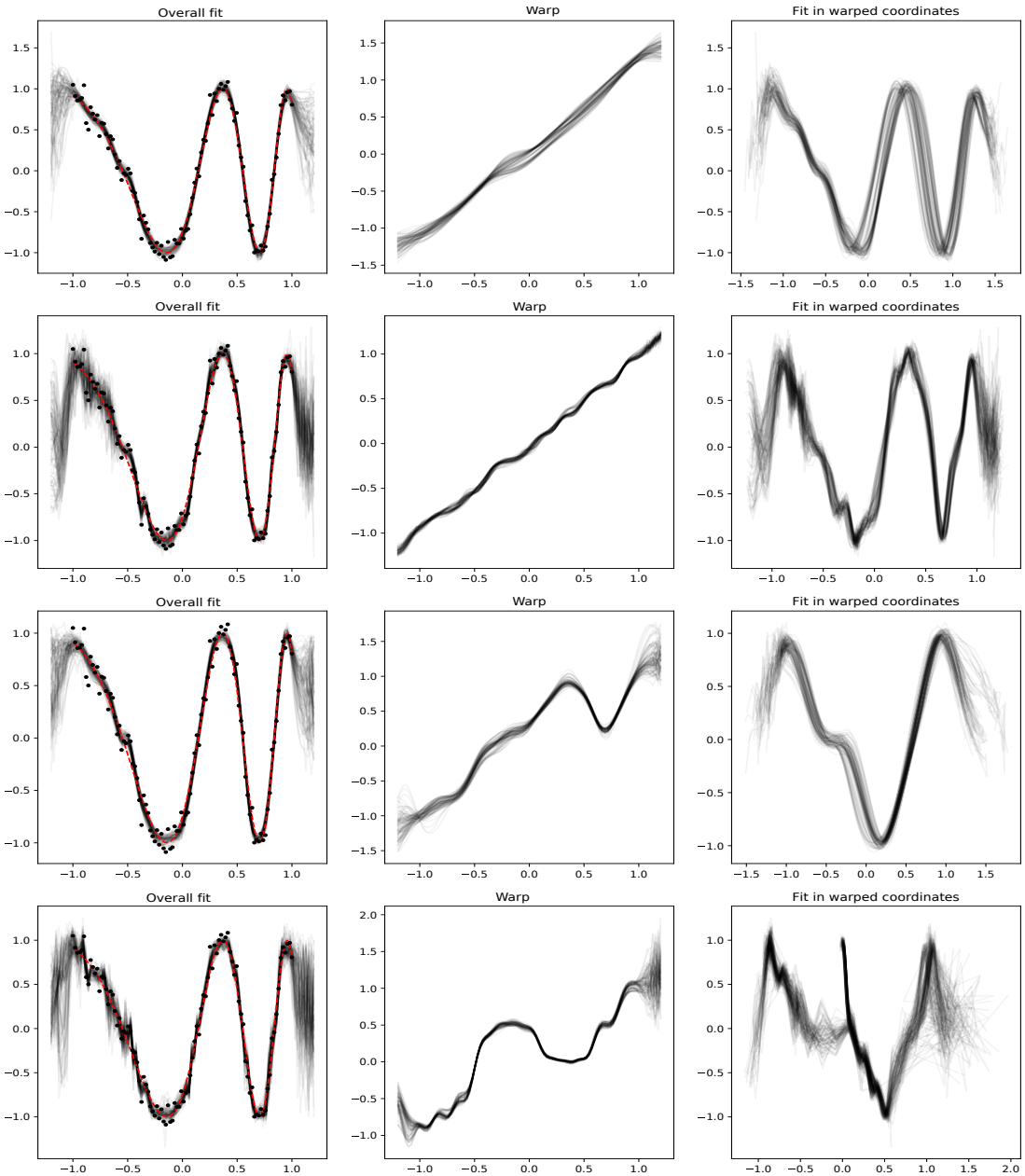


Figure A-2: A two-layer DGP fitted using SGHMC [Havasi et al., 2018]. From left to right, we plot the fitted function, the first layer of the model, and the fitted function in the warped coordinates. The rows correspond to random initialisations of SGHMC; the data is identical in all 4 cases.