

Latent Gaussian Process Regression

Erik Bodin

Department of Computer Science
University of Bristol
Bristol, United Kingdom
erik.bodin@bristol.ac.uk

Neill D. F. Campbell

Department of Computer Science
University of Bath
Bath, United Kingdom
n.campbell@bath.ac.uk

Carl Henrik Ek

Department of Computer Science
University of Bristol
Bristol, United Kingdom
carlhenrik.ek@bristol.ac.uk

Abstract

We introduce Latent Gaussian Process Regression which is a latent variable extension allowing modelling of non-stationary multi-modal processes using GPs. The approach is built on extending the input space of a regression problem with a latent variable that is used to modulate the covariance function over the training data. We show how our approach can be used to model multi-modal and non-stationary processes. We exemplify the approach on a set of synthetic data and provide results on real data from motion capture and geostatistics.

Introduction

Gaussian processes (GPs) are probabilistic objects that can be employed as priors to specify distributions over spaces of functions. This provides models with principled uncertainty specification and allows for Bayesian regularization to balance model complexity with model fit. The flexibility of GPs stems from their non-parametric structure where the characteristics of the prior is fully encapsulated in the choice of covariance and mean function. In all but few cases, the mean function is set to be constant leaving only the covariance function to be chosen.

Most covariance functions are stationary which means that there is a single structure of variations independent of location in the input space. However, for many types of data, the assumption of a stationary process is not suitable making non-stationary covariances desirable. Creating such covariances often leads to an explosion in the number of parameters, effectively removing the benefit of a non-parametric model.

An additional challenge with GPs is that they are limited to modelling a single function. Often we have data where, in certain parts of the input space, the data has been generated by several different functions. In such scenarios we desire a model that switches automatically between functions allowing the data to be represented by several different processes.

In this paper we present a unified framework that tackles both these problems. It allows modelling structures, such as non-stationary and multi-modal functions, using GPs without an explosion in the number of parameters. Specifically, we extend *any* covariance function with an additional latent space that encapsulates this structure in a non-parametric manner leading to a single GP with a specific covariance function.

During inference, we marginalise out these latent variables from the model using the variational approach of [19]. This

method depends on computing expectations over the covariance function of the GP. This is only analytically tractable for a subset of covariance functions, limiting the applicability of the approach. This motivates the second contribution of this paper. We show that these expectations can be approximated efficiently using Monte Carlo methods, yielding otherwise intractable covariances (such as ours) tractable. It is also beneficial compared to when the expectations need to be analytically tractable as it allows for rapid prototyping by removing challenging and time consuming derivations.

Background

An attractive property of Gaussian processes is that, through very simple means, it is possible to formulate priors that are both interpretable and expressive. Examples are covariance functions, such as the squared exponential, which with a single parameter encode a global smoothness structure. However, for many types of data these global assumptions are not valid. There has been significant interest in how to describe non-stationary covariances allowing for either changing function behaviour, as in [2], or for heteroscedastic noise, as in [14].

In multi-task learning, covariance functions have been defined that are able to model variations between output dimensions such as [3]. As positive-semidefinite kernels are closed under several different operations, there has also been work on how to combine covariances to generate more expressive models. In [8], the authors present an approach where a class of additive covariances is described. In a continuation of this work, the choice of covariance function was formulated as a search problem [9] where a set of base covariances could be combined using additions and multiplications. Using these techniques it is possible to create far more expressive priors while still retaining the benefits of the GP framework.

When moving from modelling stationary to non-stationary covariances, the prior assumption changes from that of a global structure to one of input dependent, local structures. When there are multiple global and/or local trends, these may be modelled by operations on the covariances; for example, as being generated from a sum of globally varying processes in [16] or as a sum of (potentially infinitely many) local experts in [17]. However, all of these models use a single generating mapping from input space to output space. When there are multiple processes generating the data independently, the

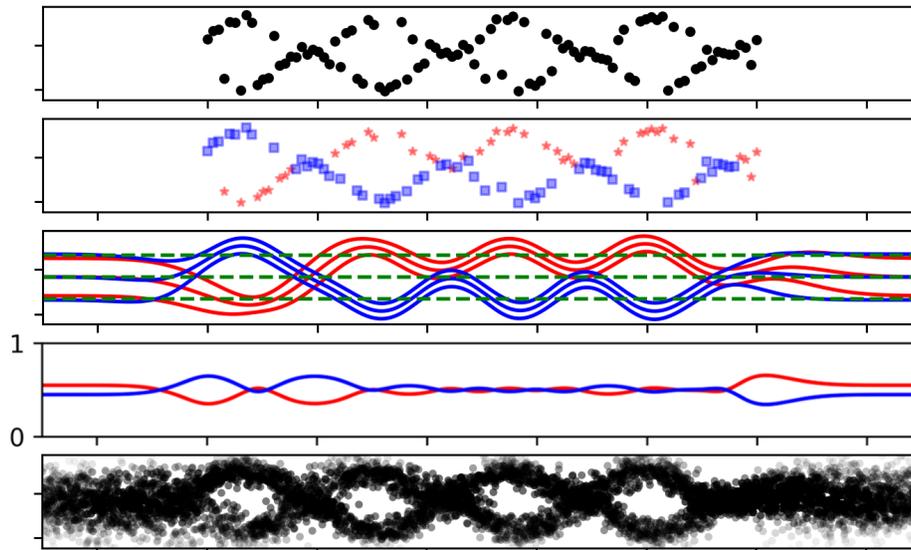


Figure 1: *Factorization.* In this experiment the factorizing kernel (8) is used to disambiguate two non-uniformly sampled superimposed sinusoids. The factorizing kernel is parameterized by two components, each one a squared exponential kernel with inferred hyper parameters. Upper plot: Synthesized data. Second plot: The component association of each observation. Third plot: Posterior predictions from each component with a standard deviation on each side of the mean. The prediction from a standard GP is shown in green. Fourth plot: Estimated component probabilities (10). Bottom plot: Posterior samples.

observations are still forced to be explained in terms of their covariances with respect to all other observations based on their position in the input space. When the underlying data generating processes give rise to multiple modes in the output space, at a single location in the input space, a single-process GP model must resort to explaining the data as noise - failing to capture the density of the data and failing to generalize from it. The strength of the GP to model smooth functions is not being utilized in this case as the covariances within each independent *partition* of the observations, produced by their respective underlying data generating process, cancel each other out.

In [15] the authors present an approach to model overlapping GPs, exemplified as a method for e.g. multi-target tracking scenarios and modelling of heteroscedasticity. In their approach, they model data-association of the observations to independent GPs via a latent association matrix. However, since the structure of the latent subspaces created by the association matrix is not explicitly modelled the approach only allows modelling of structures where there is no interdependency between groups. The method we propose in this paper includes this approach as a special case.

The creation of more complicated covariance structures presents a particular challenge during inference. GPs, in their simplest form, scale cubically with the data which has led to a significant amount of work on reducing this computational complexity. In [7] a factorised approach is presented which represents the GP as a product-of-experts that allows for massively distributed computations. However, creating a factorised model such as this relies on making independence assumptions that are not necessarily straight forward or are restrictive on modelling power.

Another approach is to sparsify the GP and use a smaller set of points, referred to as inducing points, to approximate the full model [5]. This then presents a challenge on how to select the inducing points. In [18] a variational approach was presented that learns these inducing points by viewing them as variational parameters specifying the bound. Using the same approach, the authors showed that an extension of the same idea applies to unsupervised learning with GPs [19] facilitating approximative integration of the latent variables. The downside of the variational approach presented in [18, 19] is that it requires calculations of expectations over the covariance function. These calculations cumbersome and sometimes intractable which limits scope of applicable covariance functions.

In this paper, we describe a simple extension to any covariance function that allows modelling of non-stationary multi-modal behaviour. Our formulation can model both non-stationary functions, i.e. when the behaviour of the function is different in different parts of the input domain, and also multi-modal functions where a single input location can be associated with several different outputs. Our approach is based on combining any covariance function with an additional covariance over a latent input space. Using this approach we create a non-parametric model for non-stationary and multi-modal data. We approximately integrate out the latent space using the variational approach in [19]. We show empirically that it is possible to approximate the challenging expectations using efficient sampling. This makes our approach applicable to any type of covariance independent of whether or not the expectations can be computed in closed form.

Latent Gaussian Process Regression

Consider a set of N input output pairs $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ generated from a non-stationary process $\mathbf{y}_i = g(\mathbf{x}_i) + \epsilon$. We can describe this function as an expansion of processes,

$$\mathbf{y}_i = \sum_j \alpha^{(j)}(\mathbf{x}_i) f^{(j)}(\mathbf{x}_i) + \epsilon \quad (1)$$

where $f^{(j)}$ is modelled by a GP. The functions $\alpha^{(j)}(\mathbf{x}_i)$ modulate the base functions $f^{(j)}$ and encode non-stationary multi-modal behaviour by smoothly segmenting the different functions over the training data. Modelling a sum of functions as a GP is straight forward so the main challenge is how to parametrise $\alpha^{(i)}(\mathbf{x}_i)$. The approach that we take in this paper is to extend the input domain with a latent variable $\mathbf{x}^{(c)}$, such that we have

$$\mathbf{y}_i = \sum_j f^{(j)}(\mathbf{x}_i, \mathbf{x}_i^{(c)}) + \epsilon. \quad (2)$$

Now, rather than directly modulating the output of the function, the latent variable can be used to modulate the covariance function in a non-parametric manner. This allows for modelling of non-functions by differentiating between several outputs at the same input location \mathbf{x} by altering $\mathbf{x}^{(c)}$. This leads to the following latent regression model,

$$p(\mathbf{Y}, \mathbf{F}, \mathbf{X}^{(c)} | \mathbf{X}) = p(\mathbf{Y} | \mathbf{F}) p(\mathbf{F} | \mathbf{X}, \mathbf{X}^{(c)}) p(\mathbf{X}^{(c)}), \quad (3)$$

where $p(\mathbf{F} | \mathbf{X}, \mathbf{X}^{(c)})$ is a GP prior over additive functions. By marginalising out the latent variables $\mathbf{X}^{(c)}$ and the GP prior we can recover the standard marginalised likelihood for GP regression,

$$p(\mathbf{Y} | \mathbf{X}) = \int p(\mathbf{Y} | \mathbf{F}) p(\mathbf{F} | \mathbf{X}, \mathbf{X}^{(c)}) p(\mathbf{X}^{(c)}) d\mathbf{F} d\mathbf{X}^{(c)}. \quad (4)$$

An intuition behind this approach is to think of the marginalisation as a projection, where multiple single-modal processes over the extended input space becomes multi-modal when projected onto the subspace of the original data. We will now describe how we achieve this by using a simple class of covariance functions that we will refer to as juxtaposition kernels.

Juxtaposition Covariance

In order to achieve the desired characteristics described above, we will study covariances that are defined as sums of products of different covariance functions evaluated in both the original and the extended input space as,

$$k_{\text{juxta}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=0}^L w_l(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}) k_l(\mathbf{x}_i, \mathbf{x}_j), \quad (5)$$

where $w_l(\cdot, \cdot)$ is a kernel function over latent inputs $\mathbf{X}^{(l)}$, a subset of $\mathbf{X}^{(c)}$, and $k_l(\cdot, \cdot)$ is a kernel function over the observed inputs \mathbf{X} . This can be viewed as a GP-prior consisting of a weighted sum of L different kernels $k_l(\cdot, \cdot)$ where

$w_l(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})$ describes the influence of $k_l(\cdot, \cdot)$ for the input pair \mathbf{x}_i and \mathbf{x}_j . In this paper we will evaluate a special case of this kernel that encourages a factorised representation such that the covariance of each pair of observed data points is modelled by either a single kernel function $k_l(\cdot, \cdot)$ or considered independent. This is achieved by using a linear kernel over the latent space,

$$w_l(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}) = \mathbf{x}_i^{(l)\top} \mathbf{x}_j^{(l)}, \quad (6)$$

and constraining the latent locations to reside in the corners of a simplex in the unified latent space $\mathbf{X}^{(c)}$ such that $\|\mathbf{x}_i^{(c)}\|_1 = 1, \forall i$. In this special case, the compound kernel (5) can be interpreted as a continuous convex association of each observation to a respective component kernel. To achieve this, a transformation is deployed as,

$$\varphi(x^{(l)}) = \frac{x^{(l)\alpha}}{\sum_{l'=0}^L x^{(l')\alpha}}, \quad (7)$$

where $\alpha \in \mathbb{R}$ encodes the strength of the discretisation. Since w_l is linear and $\mathbf{x}^{(l)}$ is constrained to a simplex it is sufficient for it to be one-dimensional, denoted $x^{(l)}$. To remove effects of initialisation, we use a simple annealing scheme to set α starting with a small value that increases each iteration of the optimisation. The motivation behind this is that with a small value, the associations can easily be altered while it is associated with a significantly higher ‘‘cost’’ for large alphas. This yields a *factorising* variant of the juxtaposition kernel, summarized as,

$$k_{\text{factorising}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=0}^L \varphi(x_i^{(l)}) \varphi(x_j^{(l)}) k_l(\mathbf{x}_i, \mathbf{x}_j). \quad (8)$$

Using the factorisation above we aim to address the same problem formulation as in [15] however our method is capable of generalising to any structures over the latent space which can be encoded using different latent space priors and is not limited to the linear kernel as explained above. Further, by removing the transformation Eq. 7 we can allow for continuous mixtures of processes rather than a discretely factorised.

Predictive inference for novel input locations can be done through the posterior of the model. However, during prediction the latent locations are not known and need to be marginalised from the model,

$$p(\mathbf{y}_* | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}) \approx \sum_{l=0}^L \int p(\mathbf{y}_* | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}, \mathbf{x}_*^{(l)}) p(\mathbf{x}_*^{(l)} | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}) d\mathbf{x}_*^{(l)}. \quad (9)$$

When the latent locations are confined to lie on the corners of a simplex the integral in Eq. 9 reduces to a sum over those corners. This means that the posterior becomes a Gaussian mixture where the distribution over the latent locations $p(\mathbf{x}_*^{(l)} | \mathbf{x}_*, \mathbf{Y}, \mathbf{X})$ can be interpreted as mixture coefficients. This distribution is not analytically tractable hence we proceed with an assumption. The predictive uncertainty of each

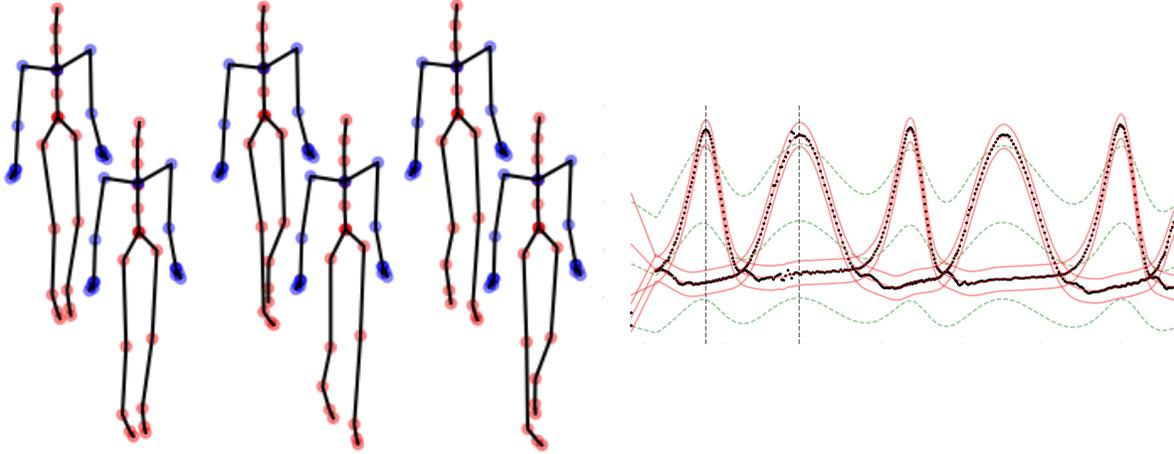


Figure 2: Disambiguating motion. In this experiment the factorizing kernel (8) is used to disambiguate joint positions in motion data. The character is walking through a room twice; once starting with the left leg and once starting with the right leg. Thus, the joint locations for the legs are ambiguous at each step, causing a 'single-process' GP to predict the mean of the two legs. The presented kernel successfully disambiguates the legs and thus preserves their motions. The training data $\mathbf{X} \in \mathbf{R}^{718 \times 54}$, $\mathbf{Y} \in \mathbf{R}^{718 \times 60}$ is 718 frames of 18 joints (chest, shoulders, arms and hands) as input and 20 joints (legs, spine and head) as output in 3D-space. Note that some of the joints are superimposed for this particular skeleton. The same squared exponential kernel with inferred lengthscale was used to parameterize the factorizing kernel as two kernel components. The upper figure shows the predicted joint locations (in red) at the input joint locations (in blue) for the character at two frames corresponding to consecutive steps using a single-process GP and using the factorizing kernel for the two components, respectively. The lower figure shows the predicted height location for the left heel given a current length-wise location across the room for the left hand, with the single-process GP in green and the two factorized components in red. The black dots in the lower figure corresponds to training data and the vertical lines mark the length-wise location for the two frames in the upper figure. The motion sequence is subject 35 sequence 1 from CMU Graphics Lab Motion Capture Database [11], with a duplicate of flipped leg motions concatenated with the original.

component is comparable. Therefore an expression of the relative certainty can be recovered given a coordinate in the input space,

$$\rho_l(\mathbf{x}_*) = \frac{\sum_{l'=0}^L \sigma_{l'}(\mathbf{x}_*)}{\sigma_l(\mathbf{x}_*)} \quad (10)$$

$$\hat{p}(x_*^{(l)} = 1 | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \frac{\rho_l(\mathbf{x}_*)}{\sum_{l'=0}^L \rho_{l'}(\mathbf{x}_*)},$$

where $\sigma_l(\mathbf{x}_*)$ is the square root of the predictive variance of the model for component l . This means that outputs can then be generated from the model at any given input location by first sampling a component according to the probabilities in (10) and then sampling from the drawn component. Examples of samples and $\hat{p}(x_*^{(l)} = 1 | \mathbf{x}_*, \mathbf{Y}, \mathbf{X})$ are provided in Figures 1, 4 and 5.

Variational Gaussian Process Latent Variable Model

The GP-LVM framework used throughout this paper, introduced in [6], deploys *auxiliary inducing variables* as a mean of marginalizing out the input. For our model the input is \mathbf{X} and $\mathbf{X}^{(c)}$, which we jointly denote $\mathbf{X}^{(s)}$. The objective function is specified as a lower bound on the data evidence. In evaluating the lower bound, the following expectations,

referred to as *sufficient statistics*, need to be evaluated:

$$\begin{aligned} \xi &= \langle \text{Tr}(\mathbf{K}_{\text{ff}}) \rangle_{q(\mathbf{X}^{(s)})} \\ \Psi &= \langle \mathbf{K}_{\text{fu}} \rangle_{q(\mathbf{X}^{(s)})} \\ \Phi &= \langle \mathbf{K}_{\text{uf}} \mathbf{K}_{\text{fu}} \rangle_{q(\mathbf{X}^{(s)})}, \end{aligned} \quad (11)$$

where $q(\mathbf{X}^{(s)}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$ and $\boldsymbol{\mu}$ and \mathbf{S} are variational parameters. These expectations are only analytically tractable for *some* covariance kernels, e.g. the linear and the squared exponential. Thus, the variational framework remains limited to the class of kernels where these expectations are tractable, significantly suppressing its modelling power. Critically, the class of kernels where these expectations are analytically tractable does not include our juxtaposition kernel.

Stochastic Approximations of Expectations

For many kernels, including ours, evaluating these expectations is not tractable; to make progress, we will proceed with a Monte Carlo approach. Besides enabling the variational framework to be used with a vastly larger set of kernels, it is easy to implement, computationally fast and, as we will show, capable of yielding virtually equivalent lower bounds. Below we provide an intuition for why this is.

Within the Variational GP-LVM framework [19, 6], the form of the approximate posterior over the latent space

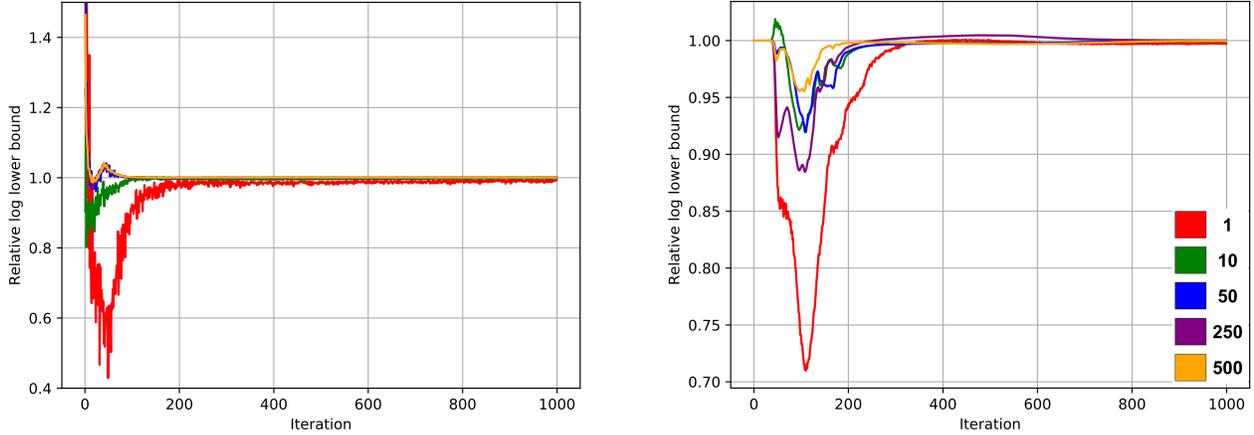


Figure 3: Comparison of the log lower bounds obtained using Monte Carlo sampling relative to the analytical expectation throughout 1000 iterations for two data sets. Sampling using 1, 10, 50, 250 and 500 samples per iteration are represented by red, green, blue, purple and orange respectively. The exact and the approximated log lower bounds have been normalized with respect to the exact starting and final log lower bound. Left: $\mathbf{x} \in \mathbf{R}^1$, $\mathbf{Y} \in \mathbf{R}^{100 \times 50}$ is a toy data set comprising 50 draws from a GP with a squared exponential kernel. Right: $\mathbf{x} \in \mathbf{R}^{40}$, $\mathbf{Y} \in \mathbf{R}^{46 \times 2048}$ is a data set of font splines used in [4].

$q(\mathbf{X}^{(s)})$ is selected to be a known parametric distribution (Gaussian) and each observation’s input location is parameterized with an egocentric, independent Gaussian for every input dimension. The expectation approximation accuracy over each entry (i, j) in $\mathbf{X}^{(s)}$ is thus independent of dimensionality. In addition the expectations of ξ and Φ are aggregates over $\mathbf{X}^{(s)}$, which further reduces the approximation error. Further, when used as part of an iterative optimization, each expectation approximation error over the course of the optimization procedure is independent. Thus the approximation error of the expectations results in a ‘noisy’ gradient which is correct on average (cf. stochastic gradient descent). In summary, the empirical means can be expressed as,

$$\begin{aligned} \xi &\approx \frac{1}{T} \sum_{t=0}^{T-1} \text{Tr} \left(\mathbf{K}_{\text{ff}}^{(t)} \right) \\ \Psi &\approx \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{K}_{\text{fu}}^{(t)} \\ \Phi &\approx \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{K}_{\text{uf}}^{(t)} \mathbf{K}_{\text{fu}}^{(t)} \end{aligned} \quad (12)$$

where $\mathbf{K}_{\text{ff}}^{(t)}$ and $\mathbf{K}_{\text{fu}}^{(t)}$ are obtained using $X_{i,j}^{(s)(t)} \sim \mathcal{N}(\mu_{i,j}, S_{i,j})$.

Implementation We implement the stochastic approximation using the ‘reparameterization trick’ as discussed in [12] and [13] to ensure we obtain low variance estimates for the expectations. The entire architecture is implemented using the Tensorflow framework [1]. This allows us to propagate gradients through the sampling procedure as if they were analytically calculated.

We will now proceed with the experiments where we provide empirical evidence that the number of samples T can be small enough to be practical to compute while still obtaining accurate enough approximations of the expectations.

Experiments

Here, we demonstrate the suitability of Monte Carlo methods for approximating the expectations of (11) as in (12). We show this with respect to the effect on the resulting lower bound. In all experiments in this section the squared exponential kernel is used to provide a comparison to the analytical expectation for a common case. Throughout the rest of the paper, the juxtaposition kernel (8) used is *analytically intractable* and we rely entirely on the sampling method.

Lower Bound In Fig. 3, the log lower bounds obtained by using the exact analytical expectations are compared to using Monte Carlo sampling. As can be seen, the lower bound obtained using approximations with 10 or more samples per iteration follows the one obtained by the exact analytical expectation closely. Furthermore, even using just one sample to approximate the expectation the lower bound converges to a value close to the analytical. This is in agreement with the findings in [12] and [13], and makes intuitive sense since the approximation error at every iteration is independent of the error at other iterations and has zero mean; resulting in a ‘noisy’ gradient of the cost function which is correct on average.

Complexity In terms of computation time, there is an important difference between the sampling method we deploy and the use of analytic expectations. The complexity of the analytic expectation, and its derivatives, can increase greatly,

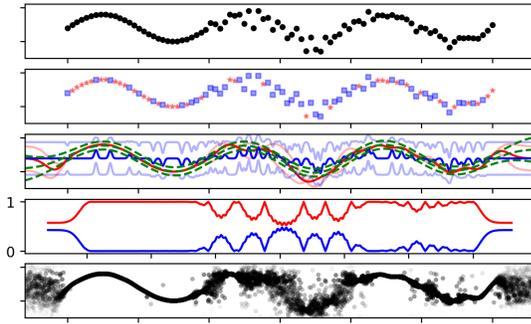


Figure 4: Heteroscedastic noise via the non-stationary mixing density. The kernel is parameterized by two components; a squared exponential kernel as well as the sum of a squared exponential and a diagonal noise term. All hyper parameters are inferred. Upper plot: Synthesized data. Second plot: The component association of each observation. Third plot: Posterior predictions from each component with a standard deviation on each side of the mean. The prediction from a standard GP is shown in green. Fourth plot: Estimated component probabilities (10). Bottom plot: Posterior samples.

even to the point of making it intractable, to that of directly evaluating the covariance matrix that is needed, regardless, for both methods. This issue is magnified particularly in the case of compound kernels.

In the best case, when using the analytical expectation the time complexity remains unchanged. In the worst case, using e.g. a simple linear kernel, the sampling method would only increase the complexity by a constant factor T (evaluating the covariance matrix T times) assuming that the expectations are the computational bottleneck. However, the covariance matrix evaluations can be run in parallel. If the constant T can be kept low without negatively impacting the obtained lower bound, this can lead to substantial speed-ups (depending on kernel and data size). In our specific environment the total computation time of the lower bound using Monte Carlo sampling with $T = 1$ was less than half compared to using the analytical expectation for the squared exponential kernel (and twice with $T = 10$). Throughout the rest of the paper $T = 1$ is used for the analytically intractable presented kernel.

Juxtaposition kernel

Multi-Modality The ability of the model to disambiguate distinct modes in the output space, at the same locations in the input space, is illustrated in Fig. 1. Since the two partitions of the observations are conditionally independent they can be explained by different covariance functions and a significantly better data fit can be obtained. Given the regularizing properties of the used variational framework, balance between model complexity and data fit is recovered automatically. The data in the example cannot be represented by a Gaussian likelihood satisfactory; this forces a standard ('single-process') GP to explain the data using a high noise variance. The result is poorly explained observations and a model with low predictive power. By allowing observations to become conditionally independent of observations close in the input space, via the latent $\mathbf{X}^{(c)}$ space, a more probable



Figure 5: 2-D data as 1-D regression. The kernel is parameterized by three squared exponential kernels. Left plot: Synthesized 2-D. Second plot: The component association of each observation. Third plot: Posterior predictions from each component with a standard deviation on each side of the mean. Right plot: Posterior samples.

and useful explanation of the data is obtained. A real world example of this is illustrated in Fig. 2, where joint positions are disambiguated in motion data.

Posterior predictions of mean and variance are provided by the individual components allowing for prediction in the input space. We compare the results with those of a normal covariance that is forced to explain all of the variations in the signal as noise. Another view-point is to think of our model as a means to cast a multi-modal problem as a regression problem. An extreme case of this is shown in Fig. 5 where the letter S is decomposed into three different functions. Importantly, we can predict three different outputs from a single input space, in effect we have decomposed this multimodal regression problem into a regression problem with one free latent variable that differentiates between the modes.

Non-Stationarity An example of modelling single-modal non-stationarity in the form of heteroscedastic noise is found in Fig. 4. Here the factorisation kernel is used as a means of forming a GP prior where individual observations are explained by either a noiseless or noisy smooth function. In this model, the covariance of observations are dependent on the local properties of the data partitioning in the input space (such as relative density) in addition to the hyper parameters governing their respective components covariance function. As a result, the compound covariance function can model local noise characteristics of the data non-parametrically. By comparison, the standard squared exponential covariance overestimates the variance in the noiseless region while underestimating (being overconfident) in the noisy regions.

Jura Geostatistics

In Fig. 6a the effects of the properties of the presented kernel is illustrated on a real world data set comprised of measured element concentrations throughout the geographical region of Jura, Switzerland [10]. The geographical distance between neighbouring data points in the data set are around a kilometer. We believe it is fair to assume that the concentrations within the area around each data point are 'sporadically mixed' rather than 'homogeneously blended'. In other words, the soil within an area is not necessarily blended such that individual samples of it contain a representative concentration for the region. An analogy would be trying to locate a suitable location for a gold mine; a single measurement of gold concentration within a square kilometer, even in the most

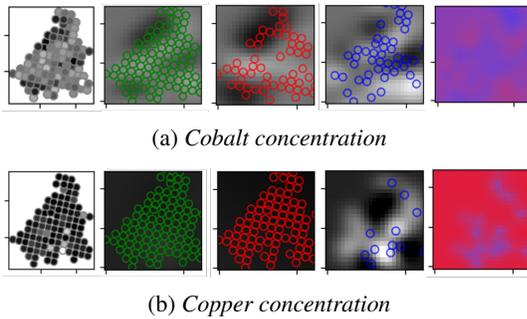


Figure 6: The factorizing kernel parameterized with two squared exponential kernels to model geostatistics data as a result of a non-stationary probabilistic mixing of two stationary gaussian processes. The data set is from [10]. Colored rings illustrate observation component associations. Left column: Element concentrations measured throughout Jura, Switzerland. Second column: Posterior means using a standard GP with a squared exponential kernel (a ‘single-component’ GP). Third and fourth column: Posterior means for the first and second component respectively when used within the factorizing kernel. Right column: Estimated component probabilities (10), where the relative mixture of red and blue illustrate the probability for respective component.

gold rich areas, can result in both a low concentration and ‘hitting the motherlode’. We model this as that any individual measurement (or sample) within a given area is drawn from any of L probability distributions. The probability for a given distribution depends on the location, which we model as $\hat{p}(\mathbf{x}_*^{(l)} = 1 | \mathbf{x}_*, \mathbf{Y}, \mathbf{X})$. For the given data, we make the conservative assumption that $L = 2$ and that independently each of the two generating processes are stationary and smoothly varying; this we model by parameterizing the factorizing kernel using squared exponential kernels. As can be seen in the respective figures, the non-stationary process is not forced to explain the local high concentrations as high levels of global noise but can capture the multi-modal smoothly varying trends of the data set.

Conclusion

We have presented Latent Gaussian Process Regression, a natural extension to GP regression that allows modelling of non-stationary multi-modal processes using a simple combination of any covariance functions. We show how our method can be used to factorise a signal into several different processes which allows modelling of multi-modal data. We also show how non-stationary single-modal data can be modelled using the same approach. Our approach builds on a latent variable extension of the input domain which is approximatively marginalised out. We provide empirical evidence which highlights that a simple sampling based approach can be used to replace expensive, and sometimes intractable, expectations in the traditional variational formulation of GP-LVMs. In its current form the number of components is a free parameter which we, in later work, hope to directly infer from data.

References

- [1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.
- [2] Ryan P. Adams and Oliver Stegle. “Gaussian Process Product Models for Nonparametric Nonstationarity”. In: *Proc. International Conference on Machine Learning (ICML)*. 2008.
- [3] Mauricio A. Álvarez and Neil D. Lawrence. “Sparse Convolved Gaussian Processes for Multi-output Regression.” In: *Neural Information Processing Systems (NIPS)*. 2008.
- [4] Neill D. F. Campbell and Jan Kautz. “Learning a Manifold of Fonts”. In: *ACM Transactions on Graphics (SIGGRAPH)* 33.4 (2014).
- [5] Joaquin Q. Candela and Carl E. Rasmussen. “A Unifying View of Sparse Approximate Gaussian Process Regression”. In: *Journal of Machine Learning Research (JMLR)* 6 (2005).
- [6] Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. “Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes”. In: *Journal of Machine Learning Research (JMLR)* 2 (2015).
- [7] Marc P. Deisenroth and Jun Wei Ng. “Distributed Gaussian Processes”. In: *International Conference on Machine Learning (ICML)*. 2015.
- [8] David K. Duvenaud, Hannes Nickisch, and Carl E. Rasmussen. “Additive Gaussian Processes”. In: *Neural Information Processing Systems (NIPS)*. 2011.
- [9] David K. Duvenaud et al. “Structure Discovery in Non-parametric Regression through Compositional Kernel Search”. In: *International Conference on Machine Learning (ICML)*. 2013.
- [10] Pierre Goovaerts. *Geostatistics for Natural Resources Evaluation*. Oxford University Press on Demand, 1997.
- [11] Ralph Gross and Jianbo Shi. “The cmu motion of body (mobo) database”. In: (2001).
- [12] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *International Conference on Representation Learning, (ICLR)*. 2014.
- [13] Diederik P. Kingma et al. “Semi-Supervised Learning with Deep Generative Models”. In: *Neural Information Processing Systems (NIPS)*. 2014.
- [14] Miguel Lázaro-Gredilla and Michalis K. Titsias. “Variational Heteroscedastic Gaussian Process Regression”. In: *International Conference on Machine Learning (ICML)*. 2011.
- [15] Miguel Lázaro-Gredilla, Steven Van Vaerenbergh, and Neil D Lawrence. “Overlapping mixtures of Gaussian processes for the data association problem”. In: *Pattern Recognition* 45.4 (2012), pp. 1386–1395.

- [16] Antoine Liutkus, Roland Badeau, and Gäel Richard. “Gaussian Processes for Underdetermined Source Separation”. In: *IEEE Transactions on Signal Processing* 59.7 (2011).
- [17] Carl E. Rasmussen and Zoubin Ghahramani. “Infinite Mixtures of Gaussian Process Experts”. In: *Neural Information Processing Systems (NIPS)*. 2002.
- [18] Michalis Titsias. “Variational learning of inducing variables in sparse Gaussian processes”. In: *International Conference on Artificial Intelligence and Statistical Learning*. 2009, pp. 567–574.
- [19] Michalis Titsias and Neil D. Lawrence. “Bayesian Gaussian Process Latent Variable Model”. In: *Int. Conf. on Artificial Intelligence and Statistical Learning (AISTATS)*. 2010.