Gaussian Processes for Sequential Bayesian Inference

S. Roberts[1], M. Osborne[1], M. Ebden[1], S. Reece[1], N. Gibson[2] & S. Aigrain[2].

1. Department of Engineering Science, 2. Department of Astrophysics.
University of Oxford.

March 13, 2012

Abstract

In this paper we offer a gentle introduction to Gaussian Processes for timeseries data analysis. The conceptual framework of Bayesian modelling for timeseries data is discussed and the conceptual framework of Bayesian non-parametric modelling presented for *Gaussian Processes*. We discuss how domain knowledge influences design of the Gaussian Process models and provide case examples to highlight the approaches.

Keywords: Gaussian Processes, timeseries analysis, Bayesian modelling.

1 Introduction

If we are to exploit the richness of scientific data available to us we must consider a principled framework under which we may reason and infer. To fail to do this is to ignore uncertainty and risk false analysis, decision making and forecasting. What we regard as a prerequisite for intelligent data analysis is ultimately concerned with the problem of computing in the presence of uncertainty. Considering data analysis under the mathematics of modern probability theory allows us to exploit a profound framework under which information, uncertainty and risk for actions, events and outcomes may be uniquely defined. Much recent research hence focuses on the principled handling of uncertainty for distributed modelling in complex environments which are highly dynamic and can be communication poor, observation costly and time-sensitive. The machinery of probabilistic inference brings to the field of timeseries analysis and monitoring robust, stable, computationally practical and principled approaches which naturally accommodate these real-world challenges. As a framework for reasoning in the presence of uncertain, incomplete and delayed information we appeal to Bayesian inference. This allows us to perform robust modelling even in highly uncertain situations, and has a long pedigree in inference. Being able to include measures of uncertainty allows, for example, us to actively select where and when we would like to observe samples and offers approaches by which we may readily combine information from multiple noisy sources.

This paper favours the conceptual over the mathematical[1]. We start in the next section with a short overview of why *Bayesian* modelling is important in timeseries analysis, culminating in arguments which provoke us to use non-parametric models. Section 3 presents a conceptual overview of a particular flavour of non-parametric

[1] Of course the mathematical details are important and elegant but would obscure the aims of this paper. The interested reader is encouraged to read the original material, or a canonical text such as [1].

model, the Gaussian Process, which we argue is well-suited to timeseries modelling. We discuss in more detail the role of *covariance functions*, the influence they have on our models and explore, by example, how the (apparently subjective) function choices we make are in fact motivated by domain knowledge. Section 4 presents real-world timeseries examples, from sensor networks, changepoint data and astronomy, to highlight the practical application of Gaussian Process models. The more mathematical framework of inference is detailed in section 5.

2 Bayesian time series analysis

We start by casting timeseries analysis into the format of a *regression* problem, of the form $y(x) = f(x) + \eta$, in which $f()$ is a (typically) unknown function and η is a (typically white) additive noise process. The goal of inference in such problems is two-fold; firstly to evaluate the putative form of $f()$ and secondly to evaluate the probability distribution of y for some x_*, i.e. $p(y|x_*)$. To enable us to perform this inference we assume the existence of a dataset of *observations*, typically obtained as input-output pairs, $\mathbb{D} = (x_i, y_i)$ for example. For the purposes of this paper we make the tacit assumption that the inputs x_i (representing, for example, time locations of samples) are known precisely, i.e. there is no *input noise*, but that observation noise is present on the y_i. When we come to analyse timeseries data there are two approaches we might consider. The first, which here we refer to as *instantaneous function mapping* and *curve fitting*.

Instantaneous function mapping approaches consider the inference of a function f which maps some x to the outcome variable y *without* explicit use of the (time) ordering of the data. This has the positive that x and y need not be defined as functions of time. However, it becomes more difficult to incorporate typical timeseries domain knowledge, such as belief of smoothness in y and the folding in of knowledge relating to gaps in the data is more difficult. The function mapping approach is typical when a static function $f : x \mapsto y$ is to be inferred from retrospective data. The mapping function f may be static, i.e. we believe that although the observed information x changes, the manner in which it maps to outcomes y is fixed.

Curve fitting on the other hand makes the tacit assumption that x and y are both ordered in time. Hence y is predicted based on observed past data. This has the benefit that the outcome variable is treated as lying close to a curve, which naturally takes into account the timing of observations. The relationship between x and y is hence not fixed, but conditioned on observed data which typically lies close, in time, to the point we are investigating In this paper we make the decision to concentrate on this approach, as we believe it offers a more profound model for much of the timeseries data we are concerned with.

As a simple example to introduce the canonical concepts of Bayesian modelling we consider a small set of data samples, located $x = 0, 1, 2$ and associated observed target values. Least-squares regression on this data using simple model (based on polynomial splines) gives rise to the curve shown as the line in the left panel of Figure 1. We see that, naturally, this curve fits our observed data very well. What about the credibility of the model in regions where we see no data, importantly $x > 2$? If we look at a larger set of examples of curves from the same model we obtain a family of curves which explain the observed data *almost identically* yet differ very significantly in regions where we have no observations, both interpolating between sample points, and in extrapolation. This simple example leads naturally to us considering a *distribution of curves*. Working with this distribution over curves, each of which offers an explanation for the observed data, is central to Bayesian modelling. We note that curves that lie towards the edges of this distribution have higher average curvature than those which lie close to the middle. There is an intimate relationship between curvature, complexity and Bayesian inference, leading naturally to posterior beliefs over models being a combination of how well observed data is explained and how complex the explanatory functions are. This elegant formalism encodes in a single mathematical framework such ideas as *Occam's razor*, such that simple explanations of observed data are favoured [].

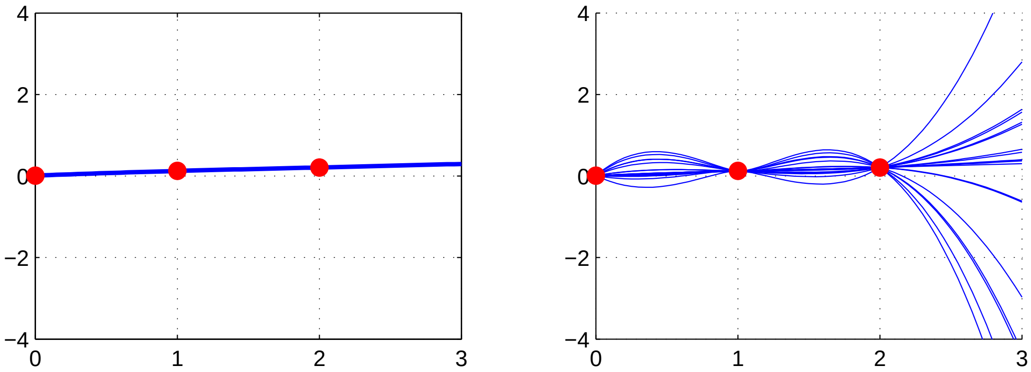

Figure 1: A simple example of curve fitting. The left panel represents the least-squares fit of a simple spline to the observed data (red dots). In the right panel shows example curves with almost identical fit to the data as the least-squares spline. These curves have high similarity close to the data yet high variability in regions of no observations, both interpolating and, importantly for time-series, as we extrapolate beyond $x = 2$.

2.1 Parametric and non-parametric models

The simple example from the previous section showed that there are many functions that can equally well explain data that we have observed. How should we choose from the bewildering array of mathematical functions that give rise to such explanatory curves? If we have strong prior knowledge regarding a system, then this (infinite) function space may be reduced to a single family; perhaps the family of quartic polynomials may be the right choice. Such models are considered to be *parametric*, in the sense that a finite number of unknown parameters (in our polynomial example, these are the coefficients of the model) need to be inferred as part of the data modelling process. Although there is a very large literature (rightly so) on such parametric modelling methods, there are many scenarios in which we have little, or no, prior knowledge regarding appropriate models to use. We may, however, have seemingly less specific domain knowledge; for example, we may know that our observations are visible examples from an underlying process which is smooth, continuous and variations in the function take place over characteristic time-scales (not too slowly yet not so fast) and have typical amplitude. Surprisingly we may work mathematically with the infinite space of all functions that have these characteristics. Furthermore, we may even contemplate probability distributions over this function space, such that the work of modelling, explaining and forecasting data is performed by refining these distributions, so focusing on regions of the function space that are excellent contenders to model our data. These functions are not of a simple pre-defined form, with sets of parameters to be inferred (unlike our simple polynomial example), this approach is referred to as a branch of *non-parametric* modelling. As the dominant machinery for working with these models is that of probability theory, they are often referred to as *Bayesian non-parametric models*. We now focus on a particular member, namely the *Gaussian Process* (GP).

3 Gaussian Processes

We start this introduction to Gaussian processes by considering a simple two-variable Gaussian distribution, which is defined for variables x_1, x_2 say, by a mean and a 2×2 covariance matrix, which we may visualise as a covariance ellipse corresponding to equal probability contours of the joint distribution $p(x_1, x_2)$. Figure 2 shows an example 2d distribution as a series of (blue) elliptical contours. The corresponding *marginal* distributions, $p(x_1)$ and $p(x_2)$ are shown as "projections" of this along the x_1 and x_2 axes (black). We now consider the effect of observing one of the variables such that, for example, we observe x_1 at the location of the dashed vertical line in the figure. The resultant *conditional distribution*, $p(x_2|x_1 = \text{known})$ indicated by the (black) dash-dot curve, now deviates significantly from the marginal $p(x_2)$. Because of the relationship between the variables implied by the covariance, knowledge of one shrinks our uncertainty in the other. To see the intimate link between this simple example and time-series analysis, we represent the same effect in a different format. Figure 3 shows the mean (black line) and $\pm\sigma$ (grey shaded region) for $p(x_1)$ and $p(x_2)$. The left panel depicts our initial state of ignorance and the right panel after we observe x_1. Note how the observation changes the location and uncertainty of the distribution over x_2. Why stop at only two variables? We can extend this example to arbitrarily large numbers of variables, the relationships between which are defined by an ever larger covariance. In principle we can extend this procedure to the limit in which the locations of the x_i are infinitely dense (here on the real line) and so the infinite joint distribution over them all is equivalent to a distribution over a function space. In practice we won't need to work with such infinite spaces, it is sufficient that we can choose to evaluate the probability distribution over the function at *any location on the real line* and that we incorporate any observations we may have at any other points. We note, crucially, that the locations of observations and points we wish to investigate the function are *not constrained* to lie on any pre-defined sample points; hence we are working in continuous time with a Gaussian Process. It is worth noting now, as we will see, that the probability distribution of a function drawn from a Gaussian process is *not necessarily Gaussian*; the distribution of a finite sample from the Gaussian Process is however, multivariate Gaussian.

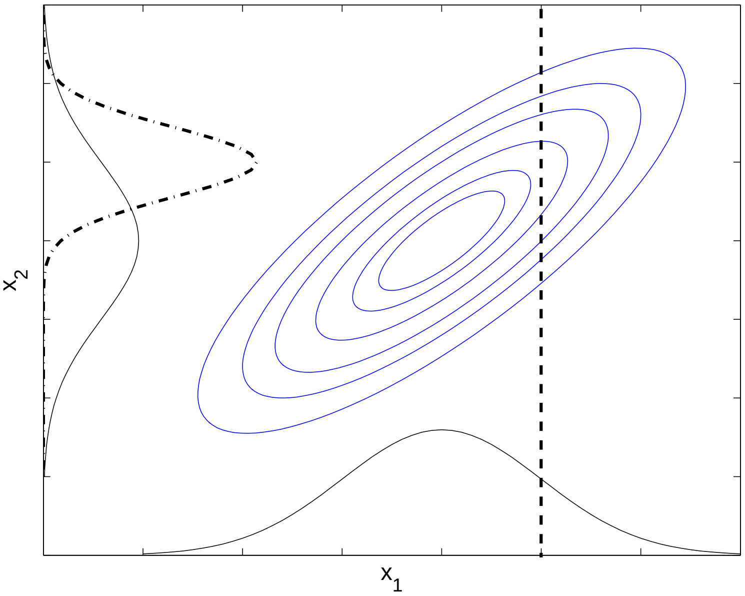

Figure 2: The conceptual basis of Gaussian Processes starts with an appeal to simple multivariate Gaussian distributions. A joint distribution (covariance ellipse) forms marginal distributions $p(x_1), p(x_2)$ which are vague (black solid). Observing x_1 at a value indicated by the vertical dashed line changes our beliefs about x_2, giving rise to a conditional distribution (black dash-dot). Knowledge of the covariance lets us shrink uncertainty in one variable based on observation of the other.

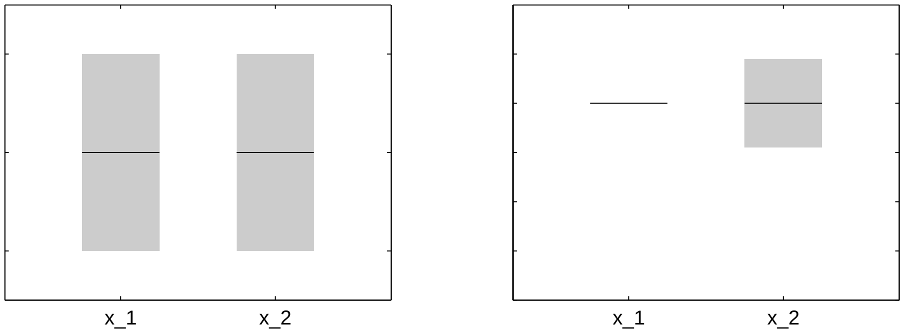

Figure 3: The change in distributions on x_1 and x_2 is here presented in a form more familiar to time-series analysis. The left panel shows the initial, vague, distributions (the black line showing the mean and the grey shading $\pm\sigma$) and the right panel subsequent to observing x_1. The distribution over x_2 has become less uncertain and the most-likely "forecast" of x_2 has also shifted.

3.1 Covariance functions

As we have seen, the covariance forms the beating heart of Gaussian Process inference. How do we formulate a covariance over arbitrarily large sets? The answer lies in defining a *covariance kernel function*, $k(x_i, x_j)$, which provides the covariance element between any two (arbitrary) sample locations, x_i and x_j say. For a set of locations, $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ we hence may define the *covariance matrix* as

$$\mathbf{K}(\mathbf{x},\mathbf{x}) = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \cdots & k(x_2,x_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \cdots & k(x_n,x_n) \end{pmatrix} \tag{1}$$

This means that the entire function evaluation, associated with the points in $\mathbf{x}$, is a draw from a multi-variate Gaussian (Normal) distribution,

$$p(\mathbf{y}(\mathbf{x})) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \mathbf{K}(\mathbf{x},\mathbf{x})) \tag{2}$$

where $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ are the dependent function values, evaluated at locations $x_1, ..., x_n$ and $\boldsymbol{\mu}$ is a *mean function*, again evaluated at the locations of the x variables (that we will briefly revisit later). If we believe there is noise associated with the observed function values, y_i, then we may fold this noise term into the covariance. As we expect noise to be uncorrelated from sample to sample in our data, so the noise term only adds to the diagonal of $\mathbf{K}$, giving a modified covariance for noisy observations of the form

$$\mathbf{V}(\mathbf{x},\mathbf{x}) = \mathbf{K}(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I} \tag{3}$$

where $\mathbf{I}$ is the identity matrix and σ^2 is a *hyperparameter* representing the noise variance.

How do we evaluate the Gaussian Process posterior distribution at some test datum, x_* say? We start with considering the joint distribution of the observed data $\mathbb{D}$ (consisting of $\mathbf{x}$ and associated values $\mathbf{y}$) augmented by x_* and y_*..

$$p\left(\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \mu(x_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x},\mathbf{x}) & \mathbf{k}(\mathbf{x},x_*) \\ \mathbf{k}(x_*,\mathbf{x}) & k(x_*,x_*) \end{bmatrix}\right) \tag{4}$$

where $\mathbf{k}(\mathbf{x}, x_*)$ is the column vector formed from $k(x_1, x_*), ..., k(x_n, x_*)$ and $\mathbf{k}(x_*, \mathbf{x})$ is its transpose. We find, after some manipulation, that the posterior distribution over y_* is Gaussian with mean and variance given by

$$m_* = \mu(x_*) + \mathbf{k}(x_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{x})), \tag{5}$$

$$\sigma_*^2 = k(x_*, x_*) - \mathbf{k}(x_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}\mathbf{k}(\mathbf{x}, x_*). \tag{6}$$

We may readily extend this to infer the GP at a set of locations outside our observations, at $\mathbf{x}_*$ say, to evaluate the posterior distribution of $\mathbf{y}(\mathbf{x}_*)$. The latter is readily obtained once more by extending the above equations and using standard results for multivariate Gaussians. We obtain a posterior mean and variance given by

$$p(\mathbf{y}_*) = \mathcal{N}(\mathbf{m}_*, \mathbf{C}_*) \tag{7}$$

where,

$$\mathbf{m}_* = \boldsymbol{\mu}(\mathbf{x}_*) + \mathbf{K}(\mathbf{x}_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{y}(\mathbf{x}) - \boldsymbol{\mu}(\mathbf{x})) \tag{8}$$

$$\mathbf{C}_* = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}(\mathbf{x}_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}\mathbf{K}(\mathbf{x}_*, \mathbf{x})^{\mathsf{T}} \,. \tag{9}$$

in which we use the shorthand notation for the covariance, $\mathbf{K}(\mathbf{a}, \mathbf{b})$, defined as

$$\mathbf{K}(\mathbf{a}, \mathbf{b}) = \begin{pmatrix} k(a_1, b_1) & k(a_1, b_2) & \cdots & k(a_1, b_n) \\ k(a_2, b_1) & k(a_2, b_2) & \cdots & k(a_2, b_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(a_n, b_1) & k(a_n, b_2) & \cdots & k(a_n b_n) \end{pmatrix} \tag{10}$$

If we believe (and in most situations we do) that the observed data are corrupted by a noise process, we would replace $\mathbf{K}$ above with, for example, $\mathbf{V}$ from Equation 3 above.

What should the functional form of the kernel function $k(x_i, x_j)$ be? To answer this we will start by considering what the covariance elements indicate. In our simple 2d example, the off-diagonal elements define the correlation between the two variables. By considering a time-series which we believe is locally smooth we expect, as $|x_i - x_j|$ increases, the resultant covariance element to decrease. This gives rise to a variety of well-known covariance functions, the most widely used perhaps being the *squared exponential,* given by

$$k(x_i, x_j) = h^2 \exp\left[-\left(\frac{x_i - x_j}{\lambda}\right)^2\right] \tag{11}$$

In the above equation we see two more *hyperparameters*, namely h, λ, which respectively govern the output scale of our function and the input, or time, scale. The role of inference in Gaussian process models is to refine vague distributions over many, very different curves, to more precise distributions which are focused on curves that explain our observed data. As the form of these curves is uniquely controlled by the hyperparameters so, in practice, inference proceeds by refining distributions over them. As h controls the gain, or magnitude, of the curves, we set this to $h = 1$ to generate Figure 4 which shows curves drawn from a Gaussian process (with squared exponential covariance function) with varying $\lambda = 0.1, 1, 10$ (panels from left to right). The important question of *how* we infer the hyperparameters is left until later in this paper, in section 5. We note that to be a valid covariance function, $k()$, implies only that the resultant covariance matrix, generated using the function, is guaranteed to be positive (semi-) definite. As a simple example, the left panel of Figure 5 shows a small sample of six observed data points, shown as dots, along with (red) error bars associated with each. The seventh datum, with green error bars and '?' beneath it, is unobserved. We fit a Gaussian Process with the squared exponential covariance kernel (Equation 11 above). The right panel shows the GP posterior mean (black curve) along with $\pm 2\sigma$ (the posterior standard deviation). Although only a few samples are observed, corresponding to the set of

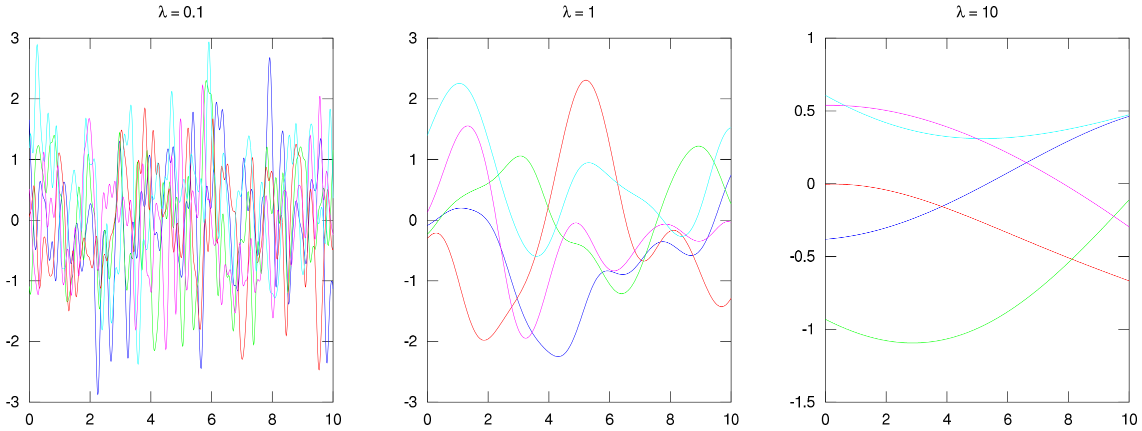

Figure 4: From left to right, functions drawn from a Gaussian process with a squared exponential covariance function with output-scale $h = 1$ and length scales $\lambda = 0.1, 1, 10$.

$\mathbf{x}, \mathbf{y}$ of equations 8 and 9, we here evaluate the function on a fine set of points, evaluating the corresponding y_* posterior mean and variance using these equations and hence providing interpolation between the noisy observations (this explains the past) and extrapolation for $x_* > 0$ which predicts the future. In this simple example we have used a "simple" covariance function. As the sum of valid covariance functions is itself a valid covariance function (more on this in section 3.3.1 later, so we may entertain more complex covariance structures, corresponding to our prior belief regarding the data. Figure 6 shows Gaussian Process modelling of observed (noisy) data for which we use slightly more complex covariances. The left panel shows data modelled using a sum of squared exponential covariances, one with a bias towards shorter characteristic timescales than the other. We see how this combination elegantly allows us to model a system with both long and short term dynamics. The right panel uses a squared exponential kernel, with bias towards longer timescale dynamics along with a periodic component kernel (which we will discuss in more detail in section 3.3.1). Note here how extrapolation outside the data indicates a strong posterior belief regarding the continuance of periodicity.

3.2 Sequential modelling and active data selection

We start by considering a simple example, shown in Figure 7. The left hand panel shows a set of data points and the GP posterior distribution *excluding* observation of the right-most datum. The right panel depicts the same inference *including* this last datum. We see how the posterior variance shrinks as we make the observation. The previous example showed how making an observation, even of a noisy timeseries, shrinks our uncertainty associated with beliefs about the function local to the observation. We can see this even more clearly if we successively extrapolate until we see another datum, as shown in Figure 8. Rather than observations coming on a fixed time-interval grid we can imagine a scenario in which observations are costly to acquire, and we wish to find a natural balance between sampling and reducing uncertainty in the functions of interest. This concept leads us naturally in two directions. Firstly for the active *requesting* of observations when our uncertainty has grown beyond acceptable limits[2] and secondly to dropping previously observed samples from our model. The computational cost of Gaussian Processes is dominated by the inversion of a covariance matrix (as in Equation 9) and hence scales with the cube of the number of retained samples. This leads to an adaptive *sample retention*. Once more the balance is problem specific, in that it relies on the trade-off between computational speed and

[2] Of course these limits are related to the cost of sampling and observation and the manner in which uncertainty in the timeseries can be balanced against this cost.

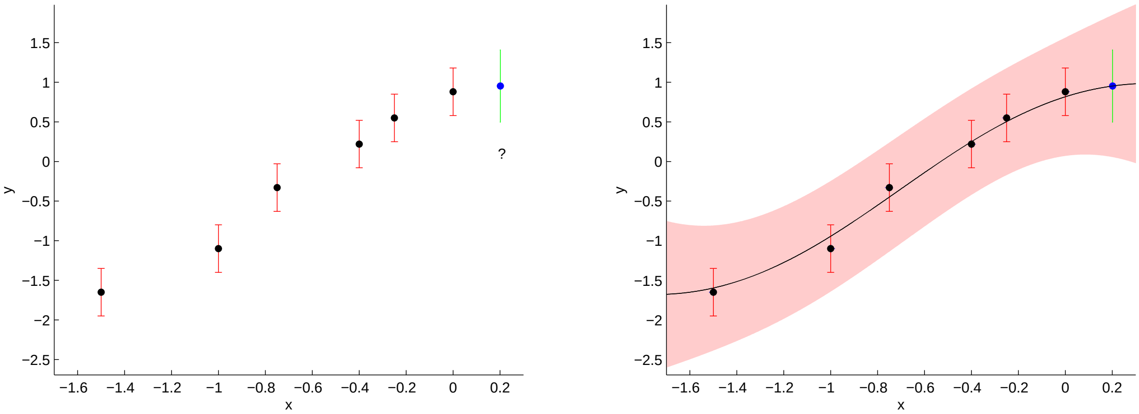

Figure 5: (left panel) Given six noisy data points (error bars are indicated with vertical lines), we are interested in estimating a seventh at $x_* = 0.2$. (right panel) The solid line indicates an estimation of y_* for x_* across the range of the plot. Along with the posterior mean, the posterior uncertainty, $\pm 2\sigma$ is shaded.

(for example) forecasting uncertainty. The interested reader is pointed to [2] for more detailed discussions. We provide some examples of active data selection in operation in real problem domains later in this paper.

3.3 Covariance and mean functions

The prior mean of a GP represents whatever we expect for our function before seeing any data. The covariance function of a GP specifies the correlation between any pair of outputs. This can then be used to generate a covariance matrix over our set of observations and predictants. Fortunately, there exist a wide variety of functions that can serve in this purpose [3, 4], which can then be combined and modified in a further multitude of ways. This gives us a great deal of flexibility in our modelling of functions, with covariance functions available to model periodicity, delay, noise and long-term drifts for example.

3.3.1 Covariance functions

In the following section we briefly describe commonly used kernels. We start with simple white noise, then consider common *stationary* covariances, both uni- and multi-dimensional. We finish this section with periodic and quasi-periodic kernel functions. The interested reader is referred to [1] for more details. Although this is not an exclusive list by any means, it provides most of the covariance functions suitable for timeseries analysis. We note once more that sums (and products) of valid covariance kernels give valid covariance functions (i.e. the resultant covariance matrices are positive (semi-) definite) and so we may entertain with ease multiple explanatory hypotheses. The price we pay lies in the extra complexity of handling the increased number of hyperparameters.

White noise with variance σ^2 is represented by:

$$k_{\text{WN}}(x_i, x_j) = \sigma^2 \mathbf{I}, \tag{12}$$

where $\mathbf{I}$ is the identity matrix. This kernel allows us to entertain uncertainty in our observed data and is so typically found added to other kernels (as we saw in Equation 3).

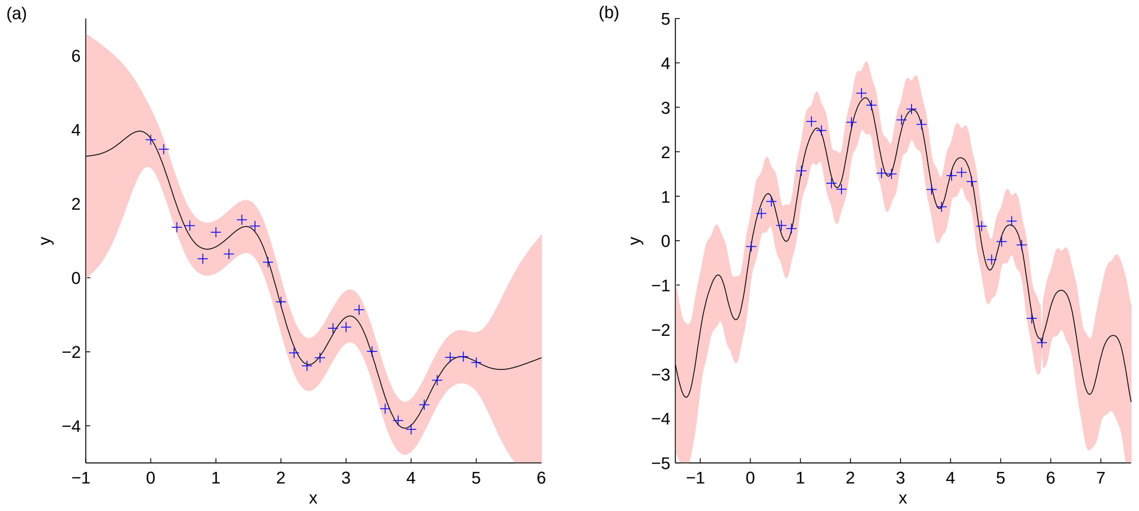

Figure 6: (a) Estimation of y_* (solid line) and $\pm 2\sigma$ posterior deviance for a function with short-term and long-term dynamics, and (b) long-term dynamics and a periodic component. Observations are shown as blue crosses. As in the previous example, we finely evaluate the posterior GP to show both interpolation and extrapolation.

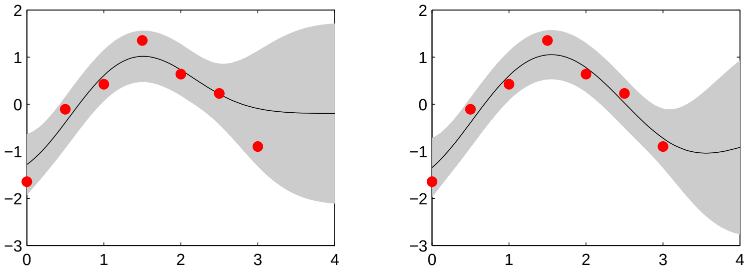

Figure 7: A simple example of a Gaussian process applied sequentially. The left panel shows the posterior mean and $\pm 2\sigma$ *prior* to observing the rightmost datum and the right panel *after* observation.

The squared exponential (SE) kernel is given by:

$$k_{\text{SE}} = h^2 \exp\left[-\left(\frac{x_i - x_j}{\lambda}\right)^2\right] \tag{13}$$

where h is an output-scale amplitude and λ is an input (length, or time) scale. This gives rather smooth variations with a typical time-scale of λ and admits functions drawn from the GP that are infinitely differentiable.

The rational quadratic (RQ) kernel is given by:

$$k_{\text{RQ}}(x_i, x_j) = h^2\left(1 + \frac{(x_i - x_j)^2}{\alpha\lambda^2}\right)^{-\alpha} \tag{14}$$

where α is known as the index. Rasmussen & Williams [1] show that this is equivalent to a scale mixture of squared exponential kernels with different length scales, the latter distributed according to a Beta distribution with parameters α and λ^{-2}. This gives variations with a range of time-scales, the distribution peaking around

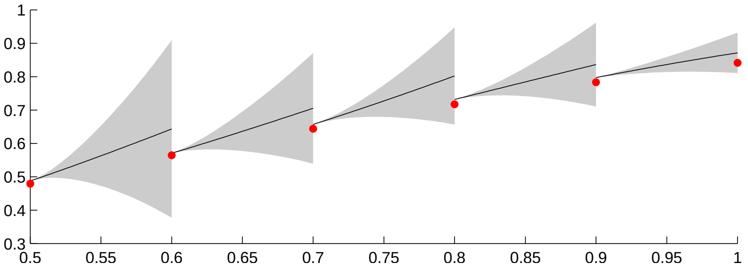

Figure 8: The GP is run sequentially making forecasts until a new datum is observed. Once we make an observation, the posterior uncertainty drops to zero (assuming noiseless observations).

λ but extending to significantly longer period (but remaining rather smooth). When $\alpha \to \infty$, the RQ kernel reduces to the SE kernel with length scale λ.

Matérn The Matérn class of covariance functions are defined by

$$k_{\mathrm{M}}(x_i, x_j) = h^2 \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(2\sqrt{\nu} \frac{|x_i - x_j|}{\lambda} \right) \mathbb{B}_\nu \left(2\sqrt{\nu} \frac{|x_i - x_j|}{\lambda} \right) \tag{15}$$

where h is the output-scale, λ the input-scale, $\Gamma()$ is the standard Gamma function and $\mathbb{B}()$ is the modified Bessel function of second order. The additional hyperparameter ν controls the degree of differentiability of the resultant functions modelled by a GP with a Matérn covariance function, such that they are only $(\nu + 1/2)$ times differentiable. As $\nu \to \infty$ so the functions become infinitely differentiable and the Matérn kernel becomes the squared exponential one. Taking $\nu = 1/2$ gives the exponential kernel,

$$k(x_i, x_j) = h^2 \exp \left(\frac{|x_i - x_j|}{\lambda} \right) \tag{16}$$

which results in functions which are only once differentiable, and correspond to the Ornstein-Ulenbeck process, the continuous time equivalent of a first order autoregressive model, AR(1). Indeed, as discussed in [1], timeseries models corresponding to AR(p) processes are discrete time equivalents of Gaussian process models with Matérn covariance functions with $\nu = p - 1/2$.

Multiple inputs and outputs The simple distance metric, $|x_1 - x_2|$, used thus far clearly only allows for the simplest case of a one dimensional input x, which we have hitherto tacitly assumed to represent a time measure. In general, however, we assume our input space has finite dimension and write $x^{(e)}$ for the value of the eth element in $\mathbf{x}$ and denote $x_i^{(e)}$ as the value of the eth element at the ith index point. In such scenarios we entertain multiple exogenous variables. Fortunately, it is not difficult to extend covariance functions to allow for these multiple input dimensions. Perhaps the simplest approach is to take a covariance function that is the product of one-dimensional covariances over each input (the *product correlation* rule [5]),

$$k(x_i, x_j) = \prod_e k^{(e)} \big(x_i^{(e)}, x_j^{(e)} \big) \tag{17}$$

where $k^{(e)}$ is a valid covariance function over the eth input. As the product of covariances is a covariance, so Equation (17) defines a valid covariance over the multi-dimensional input space. We can also introduce distance functions appropriate for multiple inputs, such as the Mahalanobis distance:

$$d^{(\mathrm{M})}(\mathbf{x}_i, \mathbf{x}_j; \mathbf{\Sigma}) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{18}$$

where $\mathbf{\Sigma}$ is a covariance matrix over the input variable vector $\mathbf{x}$. Note that this is a *hyperparameter* of the model, and should not be confused with covariances formed from covariance functions (which are always denoted by $\mathbf{K}$ in this paper). If $\mathbf{\Sigma}$ is a diagonal matrix, its role in Equation 18 is simply to provide an individual input scale $\lambda^e = \sqrt{\Sigma(e, e)}$ for the eth dimension. However, by introducing off-diagonal elements, we can allow for correlations amongst the input dimensions. To form the multi-dimensional kernel, we simply replace the scaled distance measure $|x_i - x_j|/\lambda$ of, e.g. Equation 13 with $d^{(\mathrm{M})}(\mathbf{x}_1, \mathbf{x}_2)$ from Equation 18 above.

For multi-dimensional outputs, we consider a multi-dimensional space of timeseries with a label l, which indexes the timeseries, and time denoted by x, hence forming the 2d input domain of $[l, x]$. We will then exploit the fact that a product of covariance functions is a covariance function in its own right, and write

$$k([l_m, x_i], [l_n, x_j]) = k_x(x_i, x_j)\, k_l(l_m, l_n)\,,$$

taking covariance function terms over both time and timeseries label. If the number of timeseries is not too large, we can arbitrarily represent the covariance matrix over the labels using the spherical decomposition [6]. This allows us to arbitrarily represent any possible covariance over labels. More details of this approach, which enables the dependencies between timeseries to be modelled, is found in [7] and we use this as the focus of one of our examples in Section 4 later in this paper.

Periodic and quasi-periodic kernels Note that a valid covariance function under any arbitrary (smooth) map remains a valid covariance function [8, 1]. For any function $u : x \to u(x)$, a covariance function $k()$ defined the range of x gives rise to a valid covariance $k'()$ over the domain of u. Hence we can use simple, stationary covariances in order to construct more complex (possibly non-stationary) covariances. A particularly relevant example of this,

$$u(x) = (u^{(a)}(x), u^{(b)}(x)) = \left(\cos\left(2\pi \frac{x}{T}\right), \sin\left(2\pi \frac{x}{T}\right)\right)\,, \tag{19}$$

allows us to modify our simple covariance functions above to model periodic functions. We can now take this covariance over u as a valid covariance over x. As a result, we have the covariance function, for the example of the squared exponential (13),

$$k_{\text{per-SE}}(x_j, x_j; h, w, T) = h^2 \exp\left(-\frac{1}{2w^2} \sin^2\left(\pi \left|\frac{x_j - x_j}{T}\right|\right)\right)\,. \tag{20}$$

In this case the output scale h serves as the amplitude and T is the period. The hyperparameter w is a “roughness” parameter that serves a similar role to the input scale λ in stationary covariances. With this formulation, we can perform inference about functions of arbitrary roughness and with arbitrary period. Indeed a periodic covariance functwion can be constructed from any kernel involving the squared distance $(x_i - x_j)^2$ by replacing the latter with $\sin^2[\pi(x_i - x_j)/T]$, where T is the period. The length scale w is now relative to the period, and letting $w \to \infty$ gives sinusoidal variations, whilst increasingly small values of w give periodic variations with increasingly complex harmonic content. Similar periodic functions could be constructed from any kernel. Other periodic functions could also be used, so long as they give rise to a symmetric, positive definite covariance matrix – $\sin^2$ is merely the simplest.

As described in [1], valid covariance functions can be constructed by adding or multiplying simpler covariance functions. Thus, we can obtain a *quasi-periodic* kernel simply by multiplying a periodic kernel with one

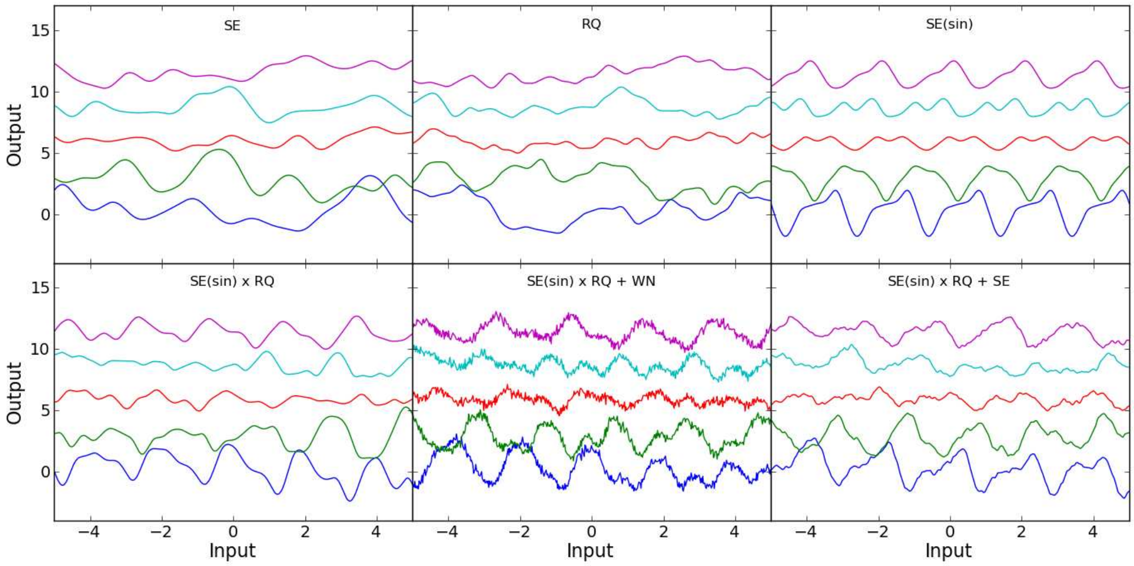

Figure 9: Random draws from Gaussian processes with different kernels. From left to right, the top row shows the squared exponential kernel, (Equation 13, with $h = 1$, $\lambda = 1$), the rational quadratic (Equation 14, with $h = 1$, $\lambda = 1$ and $\alpha = 0.5$), and a periodic kernel based on the squared exponential (Equation 20, with $h = 1$, $T = 2$ and $w = 0.5$). The bottom row left panel shows a quasi-periodic kernel constructed by multiplying the periodic kernel of Equation 13 (with $h = 1$, $T = 2$, $w = 1$) with the rational quadratic kernel of Equation 14 (with $\lambda = 4$ and $\alpha = 0.5$). The middle and right panel in the bottom row show noisy versions of this kernel obtained by adding, respectively, a white noise term (Equation 13, with $\sigma = 0.2$) and a squared exponential term (Equation 13, with $h = 0.1$, $\lambda = 0.1$). Each line consists of equally spaced samples over the interval $[-5, 5]$, and is offset from the previous one by 3 for clarity. The random number generated was initiated with the same seed before generating the samples shown in each panel.

of the basic stationary kernels described above. The latter then specifies the rate of evolution of the periodic signal. For example, we can multiply equation 20 with a squared exponential kernel:

$$k_{\text{QP,SE}}(x_i, x_j) = h^2 \exp\left(-\frac{\sin^2[\pi(x_i - x_j)/T]}{2w^2} - \frac{(x_i - x_j)^2}{\lambda^2}\right) \tag{21}$$

to model a quasi-periodic signal with a single evolutionary time-scale λ.

Examples of functions drawn from these kernels are shown in Figure 9. There are many more types of covariance functions in use, including some (such as the Matérn family above) which are better suited to model rougher, less smooth variations. However, the SE and RQ kernels already offer a great degree of freedom with relatively few hyper-parameters, and covariance functions based on these are often sufficient to model the data of interest.

Changepoints We now describe how to construct appropriate covariance functions for functions that experience sudden changes in their characteristics. This section is meant to be expository; the covariance functions we describe are intended as examples rather than an exhaustive list of possibilities. To ease exposition, we assume the (single) input variable of interest x is entirely temporal. If additional features are available, they may be readily incorporated into the derived covariances [1].

A drastic change in covariance: Suppose a function of interest is well-behaved except for a drastic change at the point x_c, which separates the function into two regions with associated covariance functions $k_1(\cdot,\cdot;\theta_1)$ before x_c and $k_2(\cdot,\cdot;\theta_2)$ after, where θ_1 and θ_2 represent the values of any hyperparameters associated with k_1 and k_2, respectively. If the change is so drastic that the observations before x_c are completely uninformative about the observations after the changepoint. The new set of hyperparameters for this covariance function contain knowledge about the original hyperparameters of the covariance functions as well as the location of the changepoint. This covariance function is easily seen to be semi-positive definite and hence admissible [9, 10].

A smooth drastic change in covariance: Suppose a *continuous function* of interest is best modelled by different covariance functions, before and after a changepoint x_c. The function values after the changepoint are conditionally independent of the function values before, given the value at the changepoint itself. This represents an extension to the drastic covariance described above; our two regions can be drastically different, but we can still enforce smoothness across the boundary between them. We call this covariance function the *continuous conditionally independent* covariance function. This covariance function can be extended to multiple changepoints, boundaries in multi-dimensional spaces, and also to cases where function derivatives are continuous at the changepoint. For proofs and details of this covariance function the reader is invited to see [11, 12].

A sudden change in input scale: Suppose a function of interest is well-behaved except for a drastic change in the input scale λ at time x_c, which separates the function into two regions with different degrees of long-term dependence.

Let λ_1 and λ_2 represent the input scale of the function before and after the changepoint at x_c, respectively. Suppose we wish to model the function with an isotropic covariance function $k()$, for example of SE form, that would be appropriate except for the change in input scale. We may model the function using the covariance function k_D defined by

$$k_D(x_i, x_j; \{h^2, \lambda_1, \lambda_2, x_c\}) = \begin{cases} k(x_i, x_j; \{h, \lambda_1\}) & (x_i, x_j < x_c) \\ k(x_i, x_j; \{h, \lambda_2\}) & (x_i, x_j \geq x_c) \\ h^2 \kappa\left(\frac{|x_c - x_i|}{\lambda_1} + \frac{|x_c - x_j|}{\lambda_2}\right) & (\text{otherwise}) \end{cases} . \tag{22}$$

in which $\kappa()$ represents, for example, the exponentiation of the square of the argument; hence forming a full covariance function.

A sudden change in output scale: Suppose a function of interest is well-behaved except for a drastic change in the output scale h at time x_c, which separates the function into two regions.

Let $y(x)$ represent the function of interest and let h_1 and h_2 represent the output scale of $y(x)$ before and after the changepoint at x_c, respectively. Suppose we wish to model the function with an isotropic covariance function $k()$ that would be appropriate except for the change in output scale. To derive the appropriate covariance function, we model $y(x)$ as the product of a function with unit output scale, $g(x)$, and a piecewise-constant scaling function, $a(x)$, defined by

$$a(x; x_c) = \begin{cases} h_1 & x < x_c \\ h_2 & x \geq x_c \end{cases} . \tag{23}$$

Given the model $y(x) = a(x)g(x)$, the appropriate covariance function for y is immediate. We may use the covariance function $k_E()$ defined by

$$k_E(x_i, x_j; \{h_1^2, h_2^2, \sigma, x_c\}) =$$

$$a(x_1; x_c)a(x_2; x_c)k(x_i, x_j; \{1, \lambda\}) = \begin{cases} k(x_i, x_j; \{h_1, \lambda\}) & (x_i, x_j < x_c) \\ k(x_i, x_j; \{h_2, \lambda\}) & (x_i, x_j \geq x_c) \\ k(x_i, x_j; \{(h_1 h_2)^{\frac{1}{2}}, \lambda\}) & \text{(otherwise)} \end{cases} . \tag{24}$$

The form of $k_E()$ follows from the properties of covariance functions, see [1] for more details.

A change in observation likelihood: Hitherto, we have taken the observation likelihood as being defined by a single GP. We now consider other possible observation models, motivated by fault detection and removal [11, 12]. A sensor fault essentially implies that the relationship between the underlying, or plant, process $y(x)$ and the observed values x is temporarily complicated. In situations where a model of the fault is known, the faulty observations need not be discarded; they may still contain valuable information about the plant process. We distinguish *fault removal*, for which the faulty observations are discarded, from *fault recovery*, for which the faulty data is utilised with reference to a model of the fault.

Perhaps the simplest fault mode is that of *bias*, in which the readings are simply offset from the true values by some constant amount (and then, potentially, further corrupted by additive Gaussian noise). Clearly knowing the fault model in this case will allow us to extract information from the faulty readings; here we are able to perform fault recovery [11, 12]. In this scenario the value of the offset and the start and finish times for the fault are additional hyperparameters to be included in the model.

Another simple fault model is that of a *stuck value*, in which our faulty readings return a constant (stuck) value regardless of the actual underlying true process. We consider the slightly more general model in which those faulty observations may also include a Gaussian noise component on top of the constant value. Here, of course, we can hope only for fault removal; the faulty readings are not at all pertinent to inference about the underlying variables of interest. This model has an extra hyperparameter, an indicator variable, which is unity if at time x_i we are within the faulty region, and is equal to zero otherwise. Here, as for the biased case, we also have additional hyperparameters corresponding to the stuck value and the start and finish times of the fault. Inference proceeds over the entire hyperparameter set, with the probability distribution over the indicator variable being of particular interest, as it encodes our posterior belief at each x_i that we are observing faulty data.

The final fault we consider is that of *drift*. Here our sensor readings undergo a smooth excursion from the plant process; that is, they gradually ‘drift’ away from the real values, before eventually returning back to normality. Unsurprisingly, the covariance kernel has an additional drift term. The model requires additional parameters that define the drift rate in the covariance function, as well as the fault start and finish times. With knowledge of this model, fault recovery is certainly possible, as shown in [11, 12].

Figure 10 shows some examples of the (non-fault) changepoint covariance functions (upper row), along with draws from the resultant GP (lower row). Each changepoint covariance function is drawn as a bold red line, with the standard squared exponential kernel shown as k_{SE} for comparison. For comparison we fix the location hyperparameter of all the functions to $x = 500$ and plot the functions over the interval from $460 \leq x \leq 560$.

3.3.2 Mean functions

As the mean function will dominate our forecasts in regions far from the data, the choice of the prior mean function can have a profound impact on our predictions and must be chosen with this in mind. In the majority of cases in the literature we find vague (i.e. high uncertainty) flat mean functions used. This choice is reinforced

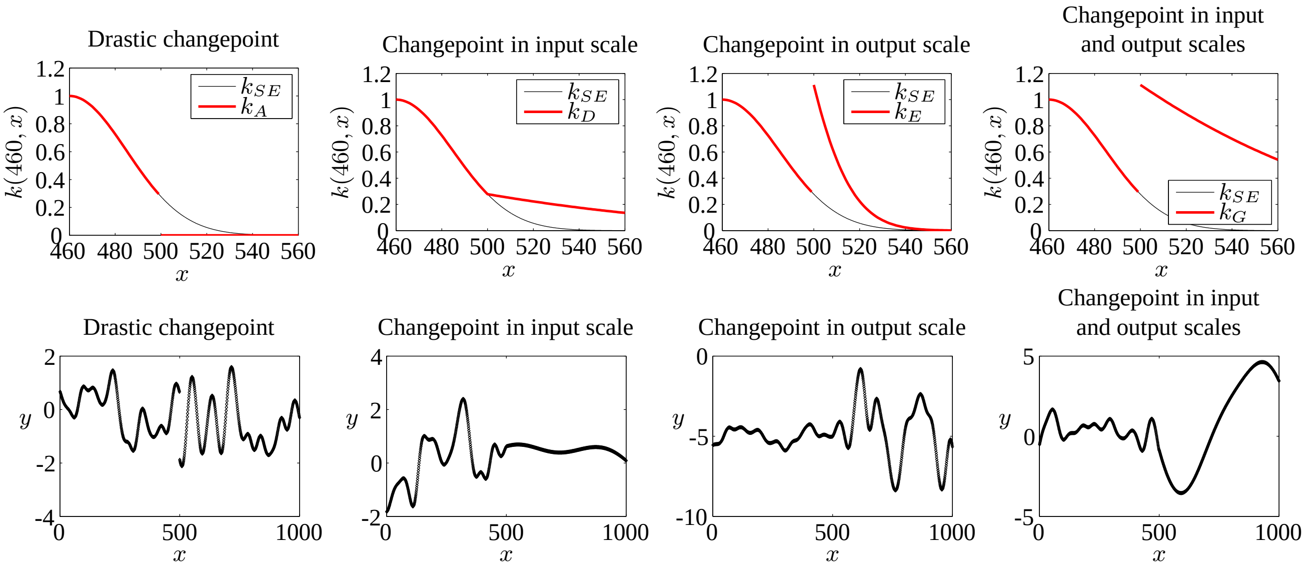

Figure 10: Example covariance functions (upper row) for the modelling of data with changepoints and associated draws (lower row) from the resultant GPs, indicating what kind of data that they might be appropriate for. Each changepoint covariance function is drawn as a bold red line, with the standard squared exponential kernel shown as k_{SE} for comparison.

by considering the prior mean function as the expectation function, prior to any observed data, of our domain beliefs. In the vast majority of situations the symmetry of our ignorance (i.e. we are equally unsure that a trend is up or down) leads to flat, often zero-offset, mean functions. As a simple example, we may have domain knowledge that our functions have a linear drift term, but we do not know the magnitude or direction. Whatever prior we place over the gradient of the drift will be necessarily symmetric and leads to a zero-mean with variance defined by the vagueness of our priors. If we do have such domain knowledge then we are free to incorporate this into our Gaussian Process models. For example, consider the case in which we know that the observed timeseries consists of a deterministic component and a n unknown additive component. Draws from our Gaussian Process are hence:

$$\mathbf{y}(\mathbf{x}) \sim \mathcal{N}\left(\mathbf{m}(\mathbf{x}; \boldsymbol{\theta}_M), \mathbf{K}(\mathbf{x}, \mathbf{x}; \boldsymbol{\theta}_C)\right) \tag{25}$$

in which the mean function, $\mathbf{m}$, has hyperparameters $\boldsymbol{\theta}_m$ that encode domain knowledge regarding the deterministic component and the covariance matrix $\mathbf{K}$ has hyperparameters $\boldsymbol{\theta}_C$. For example, we may know that our observations are obtained from an underlying exponential decay with an unknown additive function along with coloured noise. Our mean function will hence be of the form $m(x_*) = A\exp(-ax_*)$ where A, a are unknown hyperparameters. Figure 11 (left panel) shows a standard squared exponential covariance GP used to for a model for a small set of noisy data samples (red dots) drawn from a function with an underlying exponential decay. The GP models the observed data well but long-term predictions are naturally dominated by a flat prior mean function. In the right panel a GP with identical covariance is employed, but the mean function is that of an exponential decay with unknown hyperparameters. Even a few data points are sufficient for the probability distribution over the exponential hyperparameters to be inferred reasonably well leading to long-term forecasts that are dominated by a (albeit uncertain) decay function.

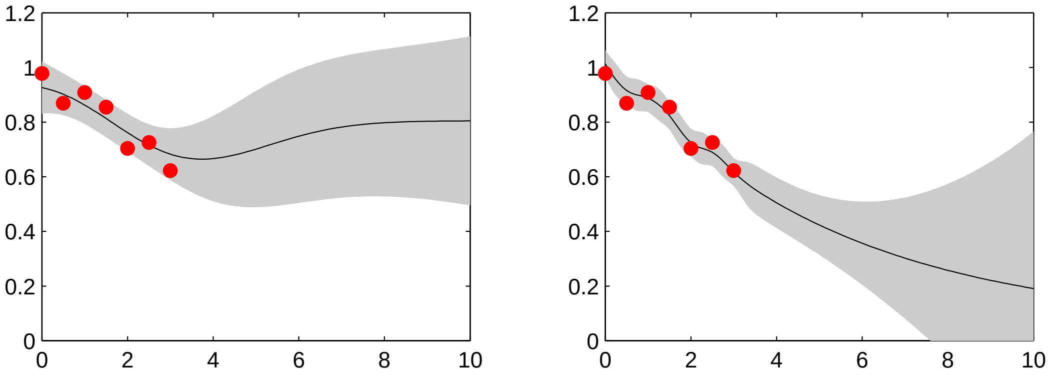

Figure 11: The effect of including a simple mean function. The left panel shows a GP model with a flat prior mean and SE covariance function. The noisy observations are indicated by (red) dots. The posterior from the GP is shown along with $\pm 2\sigma$. In the right panel the same covariance function is used, but now the mean function has extra hyperparameters corresponding to an exponential decay with unknown time-constant and scale. We see that the long-term forecasts in this example encode our prior belief in the decay function.

4 Examples

In the following examples we briefly illustrate the Gaussian Process approach to practical timeseries analysis, highlighting the use of a variety of covariance and mean functions.

4.1 Multi-dimensional weather sensor data

The first example we provide is based on real-time data which is collected by a set of weather, sea state and environment sensors on the south coast of the UK (see [7] for more details). The network (Bramblemet) consists of four sensors (named Bramblemet, Sotonmet, Cambermet and Chimet), each of which measures a range of environmental variables (including wind speed and direction, air temperature, sea temperature, and tide height) and makes up-to-date sensor measurements. We have two data streams for each variable at our disposal. The first is the real-time, but sporadic, measurements of the environmental variables; it is these that are presented as a multi-dimensional timeseries to the GP. Secondly we have access, retrospectively, to finer-grained data. We use this latter dataset for assessment only.

Figure 12 illustrates the efficacy our GP prediction for a tide height dataset. In order to manage the four outputs of our tide function (one for each sensor), we rewrite so that we have a single output and inputs t, time, and l, a sensor label, as discussed in Section 3.1 and the subsection above.

Note that our covariance over time is the sum of a periodic term and a *disturbance* term. Both are of the Matérn form with $\nu = \frac{5}{2}$. This form is a consequence of our expectation that the tides would be well modelled by the superposition of a simple periodic signal and an occasional disturbance signal due to exceptional conditions. Of course, for a better fit over the course of, say, a year, it would be possible to additionally incorporate longer-term drifts and periods.

The period T of the periodic covariance term was unsurprisingly learnt as being about half a day, whereas for the disturbance term the time scale w was found to be about two and a half hours. Note that this latter result is concordant with our expectations for the time scales of the weather events we intend our disturbance term to model.

Our algorithm learned that all four sensors were very strongly correlated, with spherical decomposition of

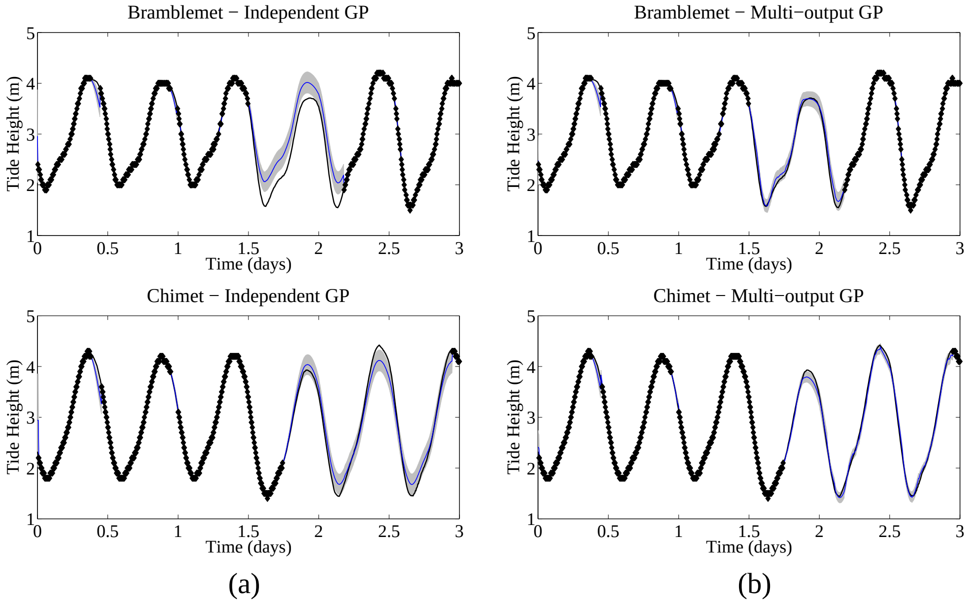

Figure 12: Prediction and regression of tide height data for (a) independent and (b) multi-output Gaussian processes.

the inferred correlation elements all very close to one. The hyperparameter matrix $\mathbf{\Sigma}$ of Equation 18 additionally gives an appropriate length scale for each sensor. Over this data set, the Chimet sensor was found to have a length scale of 1.4m, with the remainder possessing scales of close to 1m. From the inference we determined weather events to have induced changes in tide height on the order of 20%.

We also make allowances for the prospect of relative latency amongst the sensors by incorporating delay variables, introduced by a vector of delays in time observations [7]. We found that the tide signals at the Cambermet and Chimet stations were delayed by about 10 minutes relative to the other two. This makes physical sense – the Bramblemet and Sotonmet stations are located to the west of the Cambermet and Chimet stations, and the timing of high tide increases from west to east within the English channel.

Note the performance of our multi-output GP formalism when the Bramblemet sensor drops out at $t = 1.45$ days. In this case, the independent GP quite reasonably predicts that the tide will repeat the same periodic signal it has observed in the past. However, the GP can achieve better results if it is allowed to benefit from the knowledge of the other sensors' readings during this interval of missing data. Thus, in the case of the multi-output GP, by $t = 1.45$ days, the GP has successfully determined that the sensors are all very strongly correlated. Hence, when it sees an unexpected low tide in the Chimet sensor data (caused in this case by the strong northerly wind), these correlations lead it to infer a similarly low tide in the Bramblemet reading. Hence, the multi-output GP produces significantly more accurate predictions during the missing data interval, with associated smaller error bars. Exactly the same effect is seen in the later predictions of the Chimet tide height, where the multi-output GP predictions use observations from the other sensors to better predict the high tide height at $t = 2.45$ days.

Note also that there are two brief intervals of missing data for all sensors just after both of the first two peak tides. During the second interval, the GP's predictions for the tide are notably better than for the first – the greater quantity of data it has observed allows it to produce more accurate predictions. With time, the GP is able to build successively better models for the series.

Table 1: Predictive performances for five-day Bramblemet tide height dataset. We note the superior performance of the GP compared to a more standard Kalman filter model.

Algorithm	RMSE (m)	NMSE (dB)
Naïve	7.5×10^{-1}	-2.1
Kalman filter	1.7×10^{-1}	-15.2
Independent GPs	8.7×10^{-2}	-20.3
Multi-output GP	3.8×10^{-2}	-27.6

The predictive performances for our various algorithms over this dataset can be found in Table 1. For the Kalman filter comparison, a history length of 16 observations was used to generate each prediction, since this gave rise to the best predictive ability. However, note that our multi-output GP which exploits correlations between the sensors, and the periodicity in each individual sensors' measurements, significantly outperforms both the Kalman filter and the independent GP [7]. The naïve result is obtained by repeating the last observed sensor value as a forecast.

4.2 Active Data Selection

We now demonstrate our active data selection algorithm. Using the fine-grained data (downloaded directly from the Bramblemet weather sensors), we can simulate how our GP would have chosen its observations had it been in control. Results from the active selection of observations from all the four tide sensors are displayed in Figure 13. Again, these plots depict dynamic choices; at time t, the GP must decide when next to observe, and from which sensor, given knowledge only of the observations recorded prior to t, in an attempt to maintain the uncertainty in tide height below 10cm. The covariance function used was that described in the previous example, namely a sum of two $\nu = 5/2$ Matérn covariance functions, one stationary and the other of periodic form. Consider first the case shown in Figure 13(a), in which separate independent GPs are used to represent each sensor. Note that a large number of observations are taken initially as the dynamics of the sensor readings are learnt, followed by a low but constant rate of observation. In contrast, for the multi-output case shown in Figure 13(b), the GP is allowed to explicitly represent correlations and delays between the sensors. As mentioned above, this data set is notable for the slight delay of the tide heights at the Chimet and Cambermet sensors relative to the Sotonmet and Bramblemet sensors, due to the nature of tidal flows in the area. Note that after an initial learning phase as the dynamics, correlations, and delays are inferred, the GP chooses to sample predominantly from the undelayed Sotonmet and Bramblemet sensors[3]. Despite no observations of the Chimet sensor being made within the time span plotted, the resulting predictions remain remarkably accurate. Consequently only 119 observations are required to keep the uncertainty below the specified tolerance, whereas 358 observations were required in the independent case. This represents another clear demonstration of how our prediction is able to benefit from the readings of multiple sensors.

4.3 Changepoint Detection

In [9, 10] a fully Bayesian framework was introduced for performing sequential time-series prediction in the presence of changepoints. The position of a particular changepoint becomes a hyperparameter of the model

[3] The dynamics of the tide height at the Sotonmet sensor are more complex than the other sensors due to the existence of a 'young flood stand' and a 'double high tide' in Southampton. For this reason, the GP selects Sotonmet as the most informative sensor and samples it most often.

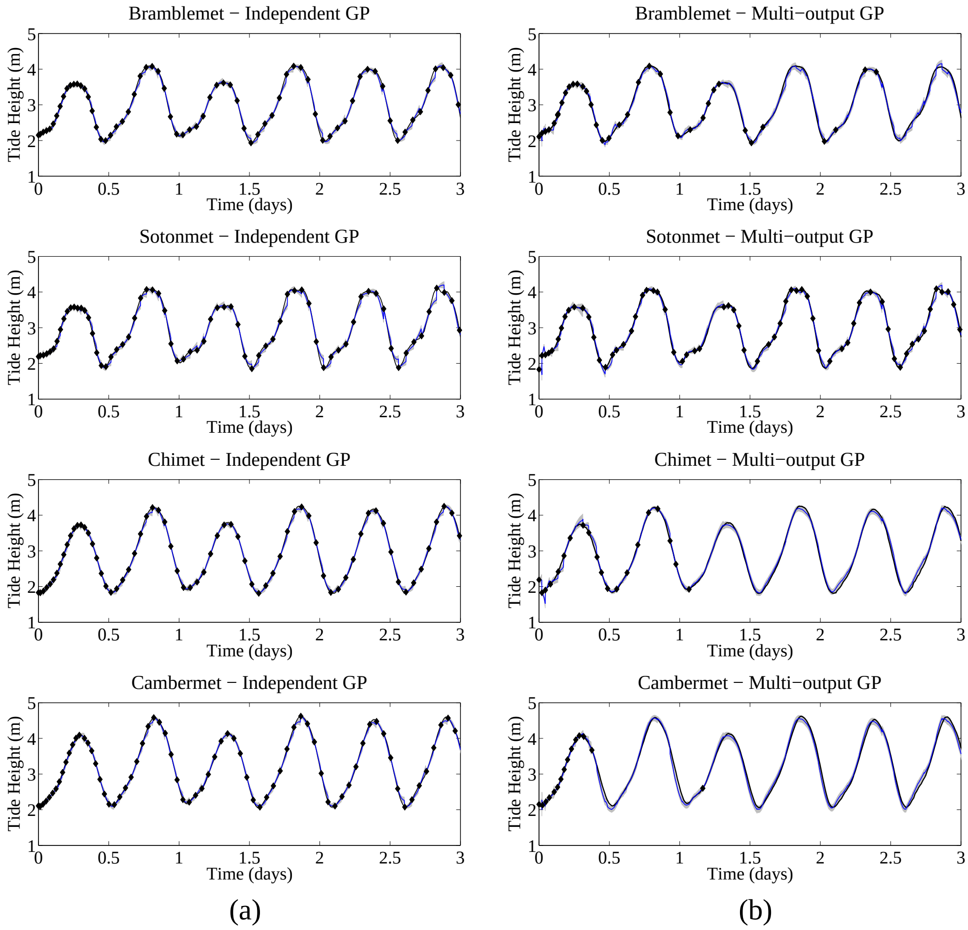

Figure 13: Comparison of active sampling of tide data using (a) independent and (b) multi-output Gaussian processes. Note that, in the case of multi-output GPs, one sensor reading (Sotonmet) slightly leads the other readings and is hence sampled much more frequently. In some cases, such as the Cambermet readings, only occasional samples are taken yet the GP forecasts are excellent.

which is obtained using Bayesian inference. If the locations of changepoints in the data are of interest, the full posterior distribution of these hyperparameters can be obtained given the data. The result is a robust time-series prediction algorithm that makes well-informed predictions even in the presence of sudden changes in the data. If desired, the algorithm additionally performs changepoint and fault detection as a natural byproduct of the prediction process. In this section we briefly present some exemplar data sets and the associated changepoint inference.

4.3.1 Nile data set

We first consider a canonical changepoint dataset, the minimum water levels of the Nile river during the period AD 622–1284 [13]. Several authors have found evidence supporting a change in input scale for this data around the year AD 722 [14]. The conjectured reason for this changepoint is the construction in AD 715 of a new device (a “nilometer”) on the island of Roda, which affected the nature and accuracy of the measurements.

We performed one-step lookahead prediction on this dataset using the input-scale changepoint covariance

K_D (22). The results can be seen in Figure 14. The upper plot shows our one-step predictions on the dataset, including the mean and $\pm\sigma$ error bars. The lower plot shows the posterior distribution of the number of years since the last changepoint. A changepoint around AD 720–722 is clearly visible and agrees with previous results [14].

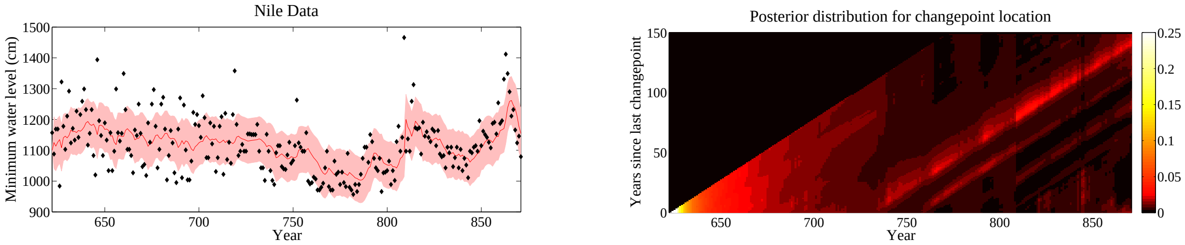

Figure 14: Prediction for the Nile dataset using input-scale changepoint covariance (left panel) and the corresponding posterior distribution for time since changepoint (right panel).

4.3.2 1972-1975 Dow-Jones industrial average

As a second canonical changepoint dataset we present the series of daily returns of the Dow-Jones industrial average between the 3rd of July, 1972 and the 30th of June, 1975 [15]. This period included a number of newsworthy events that had significant macroeconomic influence, as reflected in the Dow-Jones returns.

We performed sequential one-step prediction on this data using a GP with a diagonal covariance that assumed all measurements were IID (as under the efficient market hypothesis, returns should be uncorrelated). However, the variance of these observations was assumed to undergo changes, and as such we used a covariance that incorporated such changes in output scale. As such, we had three hyperparameters to marginalise: the variance before the changepoint, the variance after the changepoint and, finally, the location of that changepoint.

Our results are plotted in Figure 15. Our model clearly identifies the important changepoints that likely correspond to the commencement of the OPEC embargo on the 19th of October, 1973, and the resignation of Richard Nixon as President of the U.S.A. on the 9th of August, 1974. A weaker changepoint is identified early in 1973, which [15] speculate is due to the beginning of the Watergate scandal.

4.4 Quasi-periodic modelling of stellar light curves

Many Sun-like stars display quasi-periodic brightness variations on time-scales of days to weeks, with amplitudes ranging from a few parts per million to a few percent. These variations are caused by the evolution and rotational modulation of magnetically active regions, which are typically fainter than the surrounding photosphere. In this case, we may expect a range of both periodic covariance scales w and evolutionary time-scales λ, corresponding to different active region sizes and life-times respectively. This can be achieved by replacing one or both of the squared exponential (SE) kernels in equation 13 by rational quadratic (RQ) kernels (equation 14). Finally, we can also allow for short-term irregular variability or correlated observational noise by including a separate, additive SE or RQ kernel. For example, [16] used a Gaussian Process with such quasi-periodic kernels to model the total irradiance variations of the Sun in order to predict its radial velocity variations.

In Figure 16, we show the results of a quasi-periodic Gaussian Process regression to photometric observations of the well-known planet-host star HD 189733, taken from [17]. The kernel used consists of a periodic SE component (equation 21) multiplied by a RQ term (equation 14) to allow for a range of evolutionary time-scales, plus an additive white noise term (equation 12). Inference over the hyperparameters of interest yielded

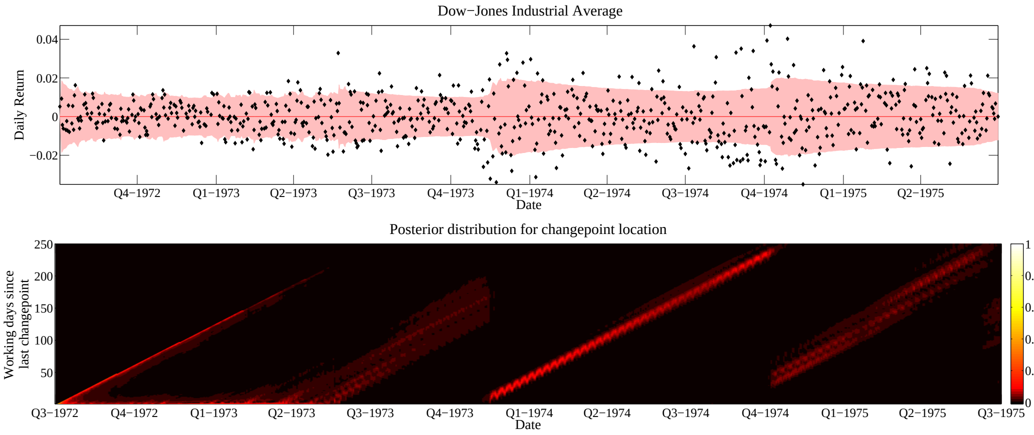

Figure 15: Online (sequential) one-step predictions (top panel) and posterior for the location of changepoint for the Dow-Jones industrial average data using an output-scale changepoint covariance (lower panel).

expected values of $h = 6.68$ mmag, $T = 11.86$ days, $w = 0.91$, $\alpha = 0.23$, $\lambda = 17.81$ days and $\sigma = 2.1$ mmag, where σ is the amplitude of the white noise term. Our period is in excellent agreement with [17]. The relatively long periodic length-scale w indicates that the variations are dominated by a small number of fairly large active regions. The evolutionary term has a relatively short time-scale, λ, but a shallow index α, which is consistent with the notion that the active regions on this star evolve relatively fast and/or that, as in the Sun, active regions located at different latitudes have different rotation rates (known as differential rotation).

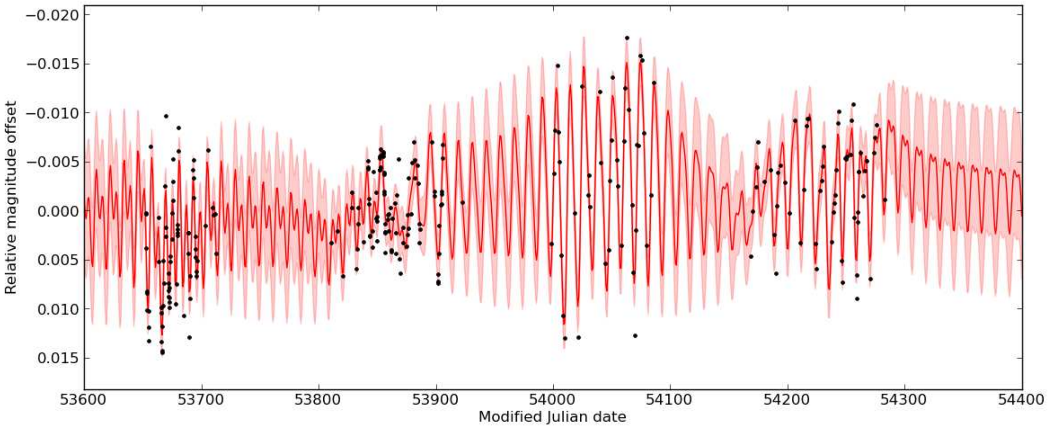

Figure 16: Predictive distribution for a quasi-periodic Gaussian process process model using a mixed SE and RQ kernel, trained and conditioned on observations made with the 0.8m APT telescope [17] using the Strömgren b and y filters. The black dots represent the observations, the red line is the mean of the predictive posterior distribution and the shaded region encompasses the $\pm\sigma$ interval.

4.5 Modelling light curves of transiting exoplanets

One of the most successful ways of discovering and characterising extra-solar planets (i.e. planets not in our solar system) is through observing transit light curves. A transit occurs when a planet periodically passes between its host star and the Earth blocking a portion of the stellar light, and produces a characteristic dip in the light curve. From this transit we can measure such physical parameters as the planet-to-star radius ratio and the inclination of the orbit. Whilst transit light curves are readily described by a deterministic parametric function, real observations are corrupted by systematic noise in the detector, external state variables (such as the temperature of the detector, orbital phase, position of the host star on the CCD array etc), as well as the underlying flux variability of the host star. As it is not possible to produce a deterministic model to account for all these systematics, a Gaussian Process may be used to place a distribution over possible artefact functions, modelling correlated noise as well as subtle changes in observed light curves due to external state variables. We hence encode the transit curve as the mean function of a GP. The covariance function has inputs given by time and external state variables (hence this is a multi-input, single output model. By integrating out our uncertainty (see Section 5) in the hyperparameters of the GP (which model all the systematic artefacts and noise processes), we can gain much more realistic inference of probability distribution of the transit function parameters (the hyperparameters of the mean function). For a detailed discussion of the application of Gaussian Processes to transit light curves see [18], in which the instrumental systematics are represented by a GP with a squared exponential kernel (Equation 13) and input parameters representing the external state variables. Robust inference of transit parameters is required to perform detailed studies of transiting systems, including the search for atomic and molecular signatures in the atmospheres of exoplanets. Figure 17 shows this GP model fitting to the timeseries of observations. More details are found in [18].

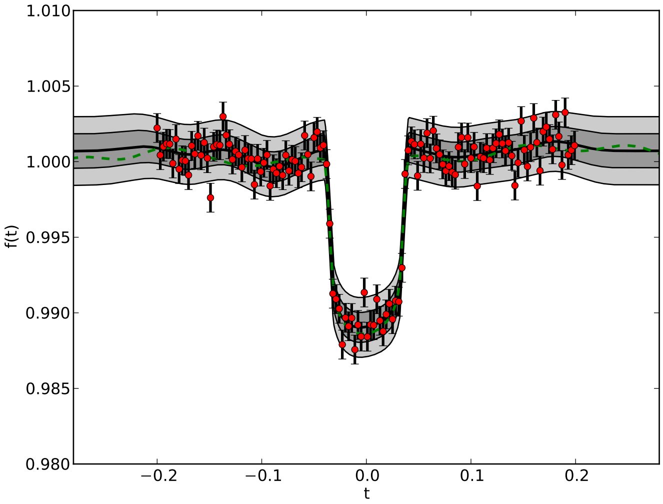

Figure 17: As an example of a complex mean function, we here model data from an exoplanet transit light curve. The data is fitted with a GP with an exoplanet transit mean function and a squared exponential covariance kernel to model the correlated noise process and the effects of external state variables. The shaded regions are at $\pm 1, 2\sigma$ from the posterior mean.

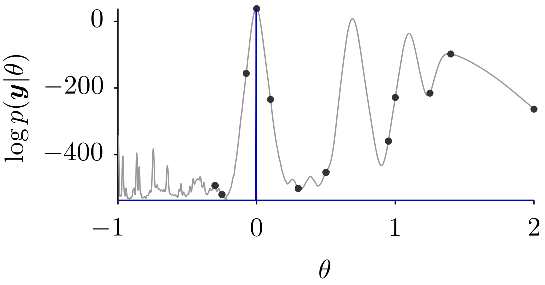

Figure 18: Samples (black dots) obtained by optimising the log-likelihood (grey) using a global optimiser, and in blue, the maximum likelihood approximation of the likelihood surface.

5 Marginalising Hyperparameters

As Gaussian Process models have a number of hyperparameters, in the covariance function (and the mean function) that we must *marginalise*[4] over in order to perform inference. That is, the quantity we are interested in is

$$p(y_\star|\boldsymbol{y}) = \frac{\int p(y_\star|\boldsymbol{y},\, \theta)\, p(\boldsymbol{y}|\theta)\, p(\theta) \mathrm{d}\theta}{\int p(\boldsymbol{y}|\theta)\, p(\theta) \mathrm{d}\theta} \tag{26}$$

which requires two integrals to be evaluated. These are both typically non-analytic, due to the complex form of the likelihood $p(\boldsymbol{y}|\theta)$ when considered as a function of hyperparameters θ. As such, we are forced to resort to approximate techniques.

Approximating an integral requires two problems to be solved. First, we need to make observations of the integrand, to explore it, and then those observations need to be used to construct an estimate for the integral. There are a number of approaches to both problems.

Optimising an integrand (see Figure 18) is one fairly effective means of exploring it: we will take samples around the maxima of the integrand, which are likely to describe the majority of the mass comprising the integral. A local optimiser, such as a gradient ascent algorithm, will sample the integrand around the peak local to the start point, giving us information pertinent to at least that part of the integrand. If we use a global optimiser, our attempts to find the global extremum will ultimately result in all the integrand being explored, as desired.

Maximising an integrand is most common when performing *maximum likelihood*. The integrands in (26) are proportional to the likelihood $p(\boldsymbol{y}|\theta)$: if the prior $p(\theta)$ is relatively flat, the likelihood will explain most of the variation of the integrands as a function of θ. Maximising the likelihood hence gives a reasonable means of integrand exploration, as above. Maximum likelihood, however, specifies a generally unreasonable means of integral estimation: the likelihood is approximated as a Dirac delta function located at the θ that maximised the likelihood. As per Figure 18, this completely ignores the width of the integrands, leading to potentially problematic features [19]. This approximation finds use when the likelihood is very peaked, as is the case when we have a great deal of data.

A slightly more sophisticated approach to integral estimation is to take a *Laplace approximation*, which fits a Gaussian around the maximum likelihood peak. This gives at least some representation of the width of the integrands. Yet further sophistication is displayed by the methods of *Variational Bayes* [20], which treat the fitting of probability distributions to the problematic terms in our integrands as an optimisation problem.

[4] The process of marginalisation refers to "integrating out" uncertainty. For example, given $p(y, \theta) = p(y|\theta)p(\theta)$ we may obtain $p(y)$ by marginalising over the unknown parameter θ, such that $p(y) = \int p(y|\theta)p(\theta)\mathrm{d}\theta$.

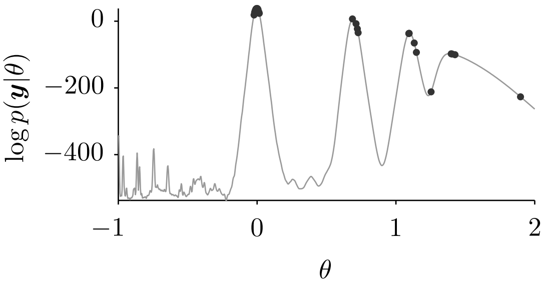

Figure 19: Samples obtained by taking draws from the posterior using an MCMC method.

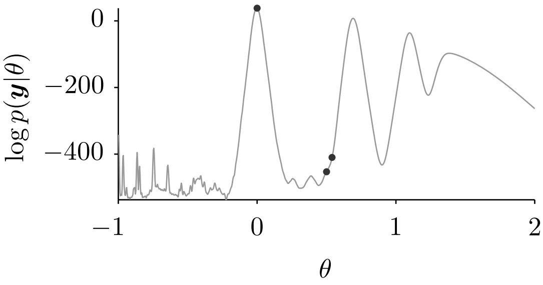

Figure 20: A set of samples that would lead to unsatisfactory behaviour from simple Monte Carlo.

Monte Carlo techniques represent a very popular means of exploring an integrand. *Simple Monte Carlo* draws random samples from the prior $p(\phi)$, to which our integrands are proportional. Note that (26) can be rewritten as

$$p(y_\star|\boldsymbol{y}) = \int p(y_\star|\boldsymbol{y},\, \theta)\, p(\theta|\boldsymbol{y})\, \mathrm{d}\theta\,. \tag{27}$$

More sophisticated *Markov Chain Monte Carlo* techniques [21] attempt to generate samples from the hyperparameter posterior

$$p(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)\, p(\theta)}{\int p(\boldsymbol{y}|\theta)\, p(\theta) \mathrm{d}\theta}\,, \tag{28}$$

to which (27) is proportional (Figure 19 illustrates samples drawn using such a method). Sampling in this way ensures that we have many samples where the prior/posterior is large, and hence, where our integrands are likely to be large. This is a particular concern for multidimensional integrals, where the problem is complicated by the ‘curse of dimensionality’ [22]. Essentially, the volume of space that could potentially be explored is exponential in its dimension. However, a probability distribution, which must always have a total probability mass of one, will be highly concentrated in this space; ensuring our samples are likewise concentrated is a great boon. Moreover, Monte Carlo sampling ensures a non-zero probability of obtaining samples from any region where the prior is non-zero. This means that we can achieve some measure of broader exploration of our integrands.

Monte Carlo, does not, however, provide a very satisfactory means of integral estimation: it simply approximates the integral as the average over the obtained samples. As discussed by [23], this ignores the information content contained in the locations of the samples, leading to unsatisfactory behaviour. For example, imagine that we had three samples, two of which were identical: $\theta_1 = \theta_2$. In this case, the identical value will receive $^2/_3$ of the weight, whereas the equally useful other value will receive only $^1/_3$. This is illustrated in Figure 20.

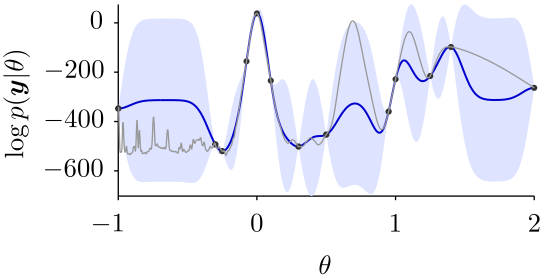

Figure 21: Bayesian quadrature fits a GP to the integrand, and thereby performs inference about the integral.

In attempt to address these issues, Bayesian quadrature [24, 25] provides a model-based means of integral estimation. This approach assumes Gaussian processes over the integrands, using the obtained samples to determine a distribution for the integrals (see Figure 21). This probabilistic approach means that we can use the obtained variance in the integral as a measure of our confidence in its estimate.

6 Conclusion

In this paper we have presented a brief outline of the conceptual and mathematical basis of Gaussian Process modelling of timeseries. As ever, a practical implementation of the ideas concerned requires jumping algorithmic rather than theoretical hurdles which we do not have space to discuss here. Some introductory code may be found at *ftp://ftp.robots.ox.ac.uk/pub/outgoing/mebden/misc/GPtut.zip* and more general code can be downloaded from *http://www.gaussianprocess.org/gpml*. Space has not permitted discussion of exciting recent trends in Gaussian Process modelling which allow for more explicit incorporation of differential equations governing the system dynamics (either observed or not), such as *Latent Force Models* [26]. Further extensions, using Gaussian Processes as building blocks in more complex probabilistic models are of course possible and recent research has also highlighted the use of GPs for numerical integration, global optimisation, mixture-of-experts models, unsupervised learning models and much more.

7 Acknowledgements

The authors would like to thank Alex Rogers, Roman Garnett, Richard Mann, Tom Evans, Mark Smith and Chris Hart. Part of this work was funded by the UK research councils, whose support is gratefully acknowledged.

References

[1] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[2] M. A. Osborne, A. Rogers, S. Ramchurn, S. J. Roberts, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 109–120, April 2008.

[3] P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 917, Norwegian Computing Center, Box 114, Blindern, N-0314 Oslo, Norway, 1997. 2nd edition.

[4] M.L. Stein. Space-Time Covariance Functions. *Journal of the American Statistical Association*, 100(469):310–322, 2005.

[5] M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002.

[6] J. Pinheiro and D. Bates. Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing*, 6:289–296, 1996.

[7] S. Roberts A. Rogers N. Jennings M. Osborne. Real-Time Information Processing of Environmental Sensor Network Data. *Transactions on Sensor Networks*, 9(1), 2012.

[8] D. J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 84–92. Springer-Verlag, 1998.

[9] R. Garnett, M. A. Osborne, and S.J. Roberts. Sequential Bayesian prediction in the presence of changepoints. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM New York, NY, USA, 2009.

[10] M. Osborne S. Reece A. Rogers S. Roberts R. Garnett. Sequential Bayesian Prediction in the Presence of Changepoints and Faults. *The Computer Journal*, 53(9):1430–1446, 2010.

[11] S. Reece, R. Garnett, M. A. Osborne, and S. J. Roberts. Anomaly detection and removal using non-stationary Gaussian processes. Technical report, University of Oxford, Oxford, UK, 2009.

[12] S. Reece, C. Claxton, D. Nicholson, and S. J. Roberts. Multi-Sensor Fault Recovery in the Presence of Known and Unknown Fault Types. In *Proceedings of the 12th International Conference on Information Fusion (FUSION 2009), Seattle, USA*, 2009.

[13] B. Whitcher, S.D. Byers, P. Guttorp, and D.B. Percival. Testing for homogeneity of variance in time series: Long memory, wavelets and the Nile River. *Water Resources Research*, 38(5):10–1029, 2002.

[14] B.K. Ray and R.S. Tsay. Bayesian methods for change-point detection in long-range dependent processes. *Journal of Time Series Analysis*, 23(6):687–705, 2002.

[15] Ryan Prescott Adams and David J.C. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge, UK, 2007. arXiv:0710.3742v1 [stat.ML].

[16] F. Pont, S. Aigrain, and S Zucker. A simple method to estimate radial velocity variations due to stellar activity using photometry. *Monthly Notices of the Royal Astronomical Society*, 419:3147, 2011.

[17] G.W. Henry and J.N. Winn. The Rotation Period of the Planet-Hosting Star HD 189733. *The Astronomical Journal*, 135:68, 2008.

[18] N.P. Gibson, S. Aigrain, S. Roberts, T. Evans, M. Osborne, and F. Pont. A Gaussian process framework for modelling instrumental systematics: application to transmission spectroscopy. *Monthly Notices of the Royal Astronomical Society*, 2012.

[19] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[20] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] J. Chen and AK Gupta. *Parametric Statistical Change Point Analysis*. Birkháuser Verlag, 2000.

[22] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, November 2000.

[23] Anthony O'Hagan. Monte Carlo is fundamentally unsound. *The Statistician*, 36:247–249, 1987.

[24] Anthony O'Hagan. Bayes-Hermite Quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

[25] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 15*, pages 489–496. The MIT Press, 2003.

[26] D. Luengo N. Lawrence M. Álvarez. Latent force models. In *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, pages 9–16, 2009.