Regression with Gaussian processes

Freddie Bickford Smith
University of Oxford

October 2020

Gaussian processes (GPs) offer a flexible and tractable way of performing nonlinear regression with uncertainty estimates. We introduce the core ideas of GP regression and apply them to make predictions of missing time-series data.[1]

1 A motivating example

Consider the data in Figure 1. We have tide-height readings for many points in time over a roughly five-day period, but some readings are missing. How should we go about estimating them?

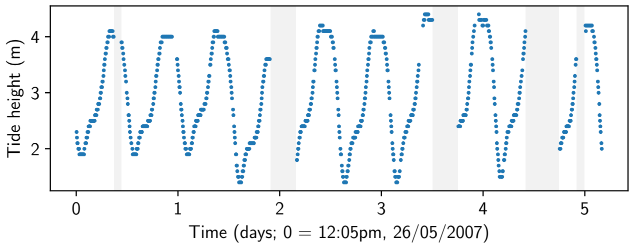

Figure 1 Given the tide-height readings we have (blue dots), we need to infer the tide height at the times where there are no readings. This is especially hard where many consecutive readings are missing (grey regions).

According to one approach, we first assume each reading, y, is generated by a model of the form

$$y = f_\theta(x) + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma_y^2) \tag{1}$$

where f_θ is an unknown parametric function, x is time and ε is measurement noise. Then the task is simply to find the f_θ that best explains the data we have.

Three sets of questions naturally arise. The first concerns how we constrain the search for a model. What is an appropriate class of functions to explore? How do we convert this into a parametric form? The second set of questions relates to selecting a model. How do we determine the best within a class of functions? How can we trade off the prediction performance on existing data against the expected performance on unseen data? Last are questions surrounding uncertainty. How confident are we in the predictions of our model? How could these predictions change if we had access to more data?

There are a number of ways of addressing these questions by modifying the approach described above (formally we would call it parametric modelling with maximum-likelihood estimation of parameters). But perhaps the most comprehensive and yet simplest solution for the problem at hand is to try a new method altogether. Instead of thinking about parametric forms, why not model functions directly?

2 Introducing Gaussian processes

A Gaussian process (GP) defines a probability distribution over functions, denoted by

$$\begin{aligned} f(x) &\sim \mathcal{GP}(m(x), k(x, x')) \\ m(x) &= \mathbb{E}[f(x)] \\ k(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \end{aligned} \tag{2}$$

where $m(x)$ is a mean function and $k(x, x')$ is a covariance function. In the same way that a mean vector and covariance matrix fully specify a multivariate Gaussian distribution, $m(x)$ and $k(x, x')$ are all that is needed to define the probability density of functions under a GP. Examples of their influence are shown in Figure 2. For demonstration, we use the exponentiated-quadratic (EQ) covariance,

$$k_{\text{EQ}}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \tag{3}$$

where σ^2 and l are variance and lengthscale parameters.

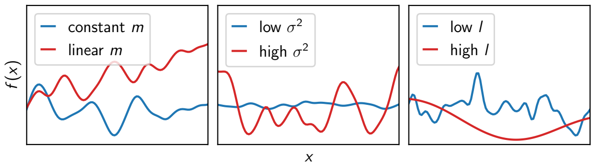

Figure 2 Varying a GP's mean function (left) and covariance function (centre, right) changes the distribution over functions, illustrated by representative sampled functions (blue and red lines). On average, for the exponentiated-quadratic covariance used here, greater σ^2 gives greater deviation from the mean and greater l gives more slowly varying functions.

3 Using a Gaussian process for regression

For simplicity, we consider a zero-mean GP. According to the definition of a GP, for a vector of test points, $\mathbf{x}_*$, the distribution of function values at those points is jointly Gaussian:

$$\begin{aligned} \mathbf{f}_* &\sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{**}) \\ \mathbf{f}_*[i] &= f\left(x_*^{(i)}\right) \\ \mathbf{K}_{**}[i, j] &= k\left(x_*^{(i)}, x_*^{(j)}\right) \end{aligned} \tag{4}$$

This is a prior over $\mathbf{f}_*$: it does not explicitly incorporate the data we are trying to model. For regression, we want to form a posterior that assigns high probability density to functions that are compatible both with the prior and the data.

[1] Code and data are available at `https://github.com/fbickfordsmith/aims-data-estimation-inference`.

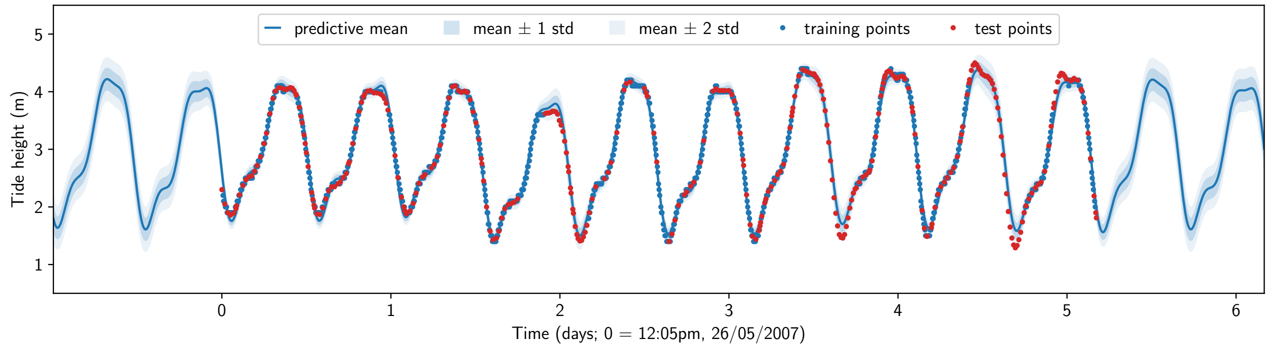

Figure 3 Our best model of the data is a GP with zero mean and the covariance function defined in Equation 9. The parameters of this covariance function were tuned to maximise the log marginal likelihood defined in Equation 7.

To do this, we first form a joint distribution that incorporates our training data, comprised of vectors **x** and **y**:

$$\begin{aligned}\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix} &\sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}_y & \mathbf{K}_*\\ \mathbf{K}_*^T & \mathbf{K}_{**}\end{bmatrix}\right)\\ \mathbf{K}_y[i,j] &= k\left(x^{(i)}, x^{(j)}\right) + \sigma_y^2 \mathbb{I}(i=j)\\ \mathbf{K}_*[i,j] &= k\left(x^{(i)}, x_*^{(j)}\right)\end{aligned} \tag{5}$$

where $\mathbf{K}_{**}$ is defined as above and we have assumed that each reading, y, is corrupted by noise as in Equation 1 (but, notably, here we do not assume f is parametric). Following standard results for conditioning Gaussians, the posterior is

$$\begin{aligned}\mathbf{f}_* \mid \mathbf{x}, \mathbf{y}, \mathbf{x}_* &\sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*))\\ \bar{\mathbf{f}}_* &= \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{y}\\ \mathrm{cov}(\mathbf{f}_*) &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_*\end{aligned} \tag{6}$$

All that remains is to find the covariance function that gives us the best posterior. To do this, we maximise the log marginal likelihood,

$$\log p(\mathbf{y} \mid \mathbf{x}) = -\frac{1}{2}\mathbf{y}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log(2\pi) \tag{7}$$

where N is the number of examples in the training data.

4 A practical application

Having covered the basics of GP regression, we return to the regression problem shown in Figure 1, where **x** is a vector of reading times and **y** is a vector of tide-height readings.

As in Section 3, we assume a mean function of zero.[2] This allows us to focus on choosing a covariance function. As a baseline, we use the EQ function (Equation 3), which encodes an assumption that the covariance of two points falls smoothly as the distance between them increases. With this, the optimised posterior (with respect to the objective in Equation 7) provides reasonable interpolation (Figure 4). Notably, though, extrapolation beyond the range of reading times in the training data is poor. From inspection, we expect the tide height to always cycle up and down as it does in the training data; instead, the predictive mean is flat.

Noting the need for periodicity in the regression function for this data, we try the following covariance function:

$$k_{\text{periodic}}(x, x') = \sigma_p^2 \exp\left(-\frac{2\sin^2(\pi|x-x'|/p)}{l_p^2}\right) \tag{8}$$

where σ_p^2 and l_p are variance and lengthscale parameters analogous to those controlling the EQ function, and p is a period parameter that determines the spacing of function repetitions. This gives more plausible extrapolation than with the EQ function (Figure 5). But the uncertainty estimates are poorly calibrated: the model is often confidently wrong in its predictions of the test data.

A solution is to form another covariance function by simply summing the two we have already tried:

$$k(x, x') = k_{\text{EQ}}(x, x') + k_{\text{periodic}}(x, x') \tag{9}$$

This gives a distribution over regression functions that, while not perfect, both captures the underlying trend in the data and gives reasonable uncertainty estimates (Figure 3). The posterior mean oscillates as we expect, and the posterior variance is high where the model makes false predictions.

We have seen how extrapolation by a GP's predictive posterior is affected the covariance function. How is it affected by the data the posterior is conditioned on? One way of looking at this is the sequential-prediction setting, where we predict future data based only on past data (Figures 8-11).

5 Conclusion

GP regression addresses each of three issues raised in Section 1. First, the mean and covariance functions provide a straightforward way of encoding prior knowledge about a problem to constrain the search for models. We showed this by stepping through the process of selecting a covariance function in Section 4. Second, the model-selection problem is solved by the use of the log marginal likelihood (Equation 7) as our objective function: it automatically trades off how well the model fits the training data against how complex the model is. Third, the probabilistic approach to regression allows us to quantify uncertainty in the model and thus know how much confidence to place in its predictions.

References

Murphy (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

Solentmet Support Group (2007). Sotonmet: weather reports from Southampton Dockhead.

[2] Since there is no obvious trend in the mean value of the data, this is a reasonable assumption as long as we normalise **y** to have zero mean. We do this by subtracting $\bar{\mathbf{y}}$ from the elements of **y** before conditioning the GP on training data. When we make predictions, we simply add $\bar{\mathbf{y}}$ back.

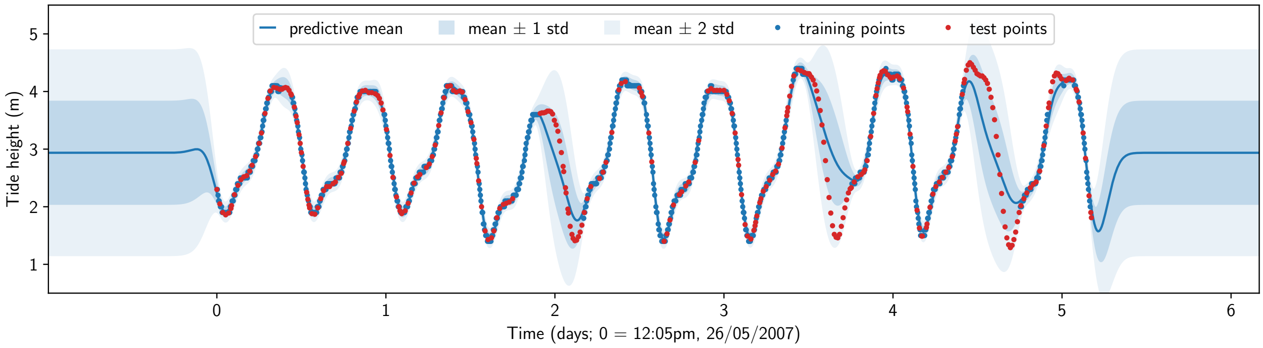

Figure 4 A GP with exponentiated-quadratic covariance (Equation 3) gives good interpolation and uncertainty estimation for our data. But it fails to capture the oscillatory nature of the tide height.

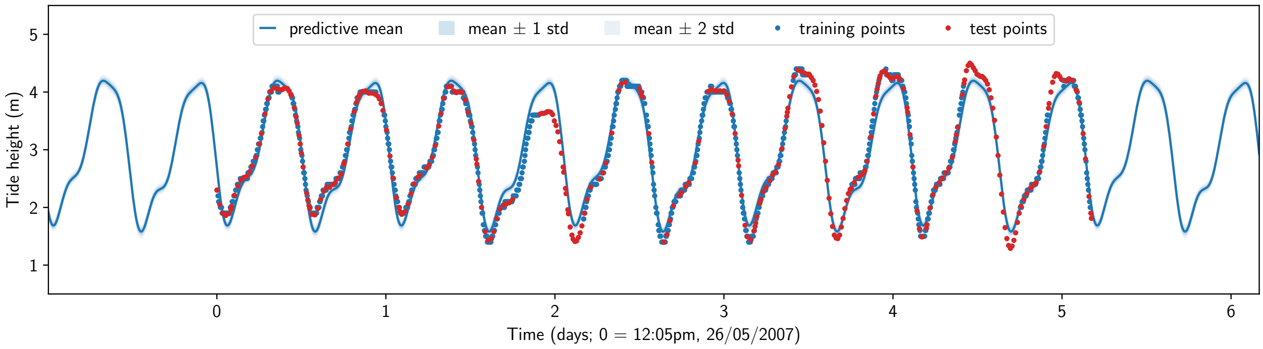

Figure 5 A GP with periodic covariance (Equation 8) gives reasonable extrapolation outside the range of reading times in the training data. But it is overconfident, with low predictive variance at all points in time, even when its predictions are wrong.

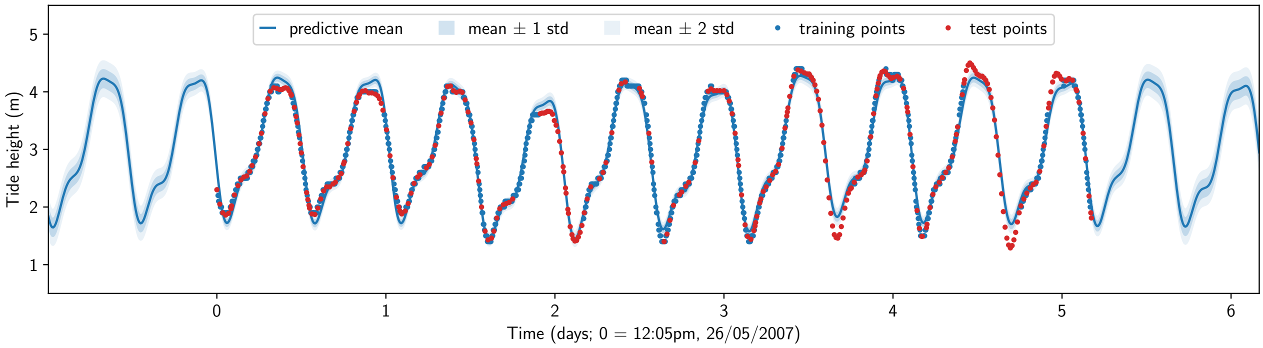

Figure 6 In order to check our implementation of GP regression is correct, we use GPflow (`https://www.gpflow.org`) to construct the same GP as shown in Figure 3. The agreement between the posterior distributions suggests there are no significant issues with our implementation.

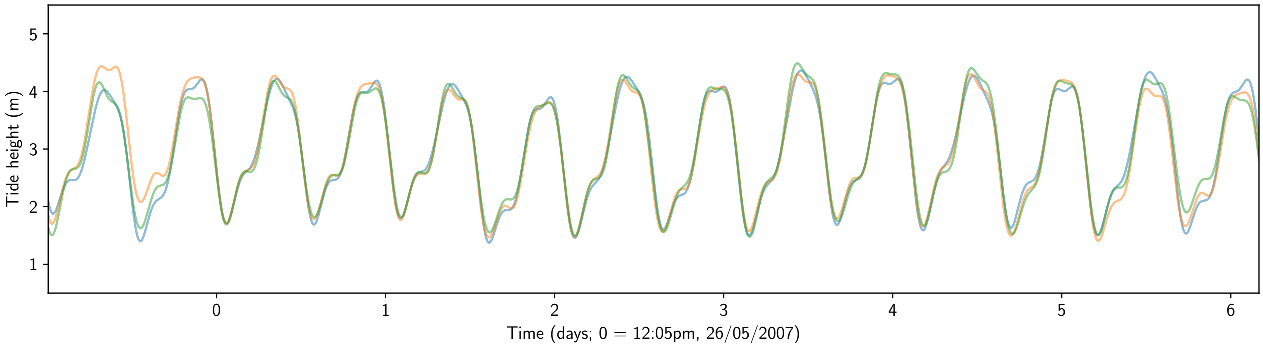

Figure 7 Another way of visualising the posterior distribution shown in Figure 3 is to plot functions sampled from the distribution.

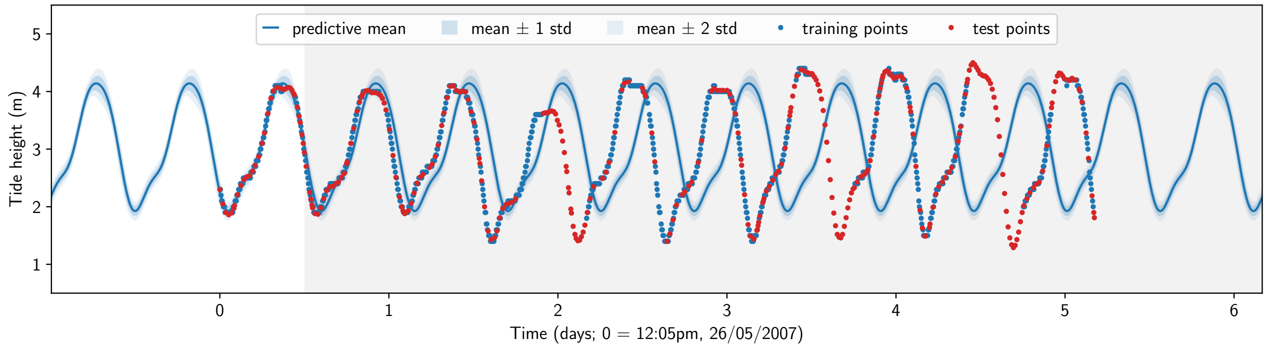

Figure 8 Here we condition a GP on the training data up to a time of 0.5 days (ie, excluding all training data in the grey region). The posterior is a poor fit to the data: neither the period nor the amplitude of the posterior mean matches that of the data.

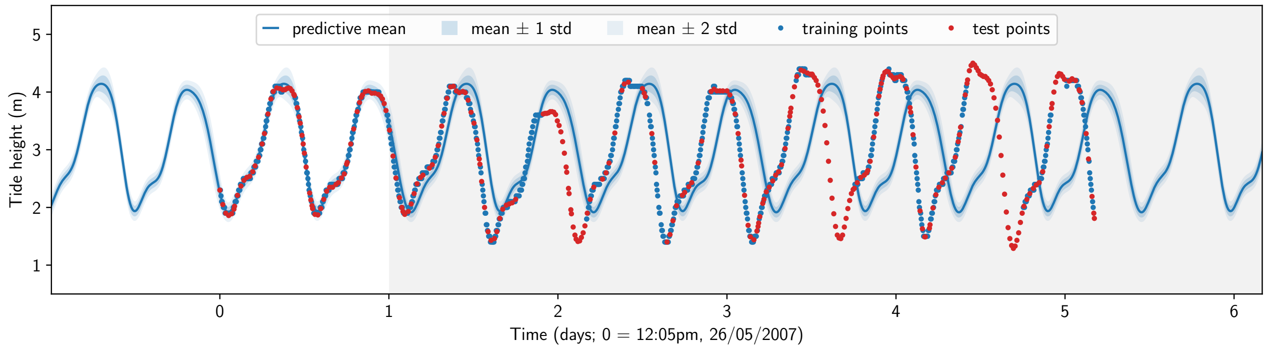

Figure 9 We now include the training data up to a time of 1 day. Compared to the case in Figure 8, there is double the amount of training data. Yet the posterior is not substantially better than before.

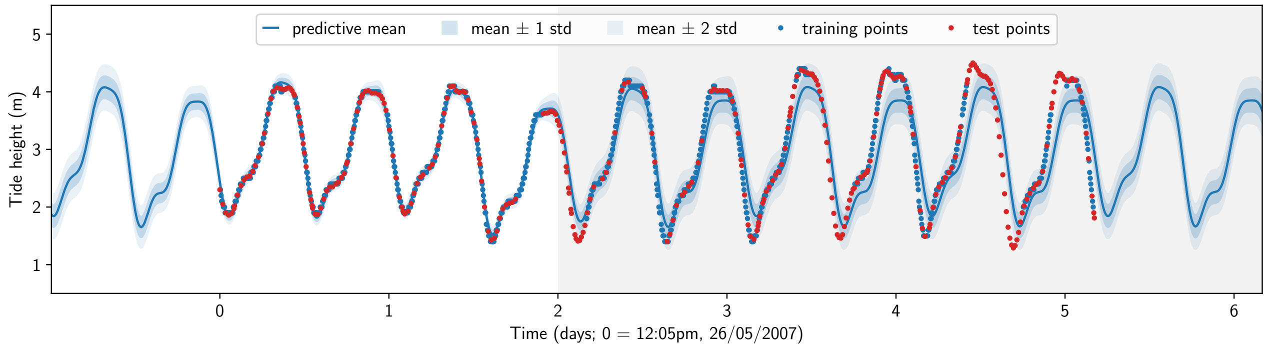

Figure 10 Once 2 days' worth of training data is used to compute the posterior, we see better correspondence between the posterior mean and the patterns in the data, and the posterior uncertainty is better calibrated.

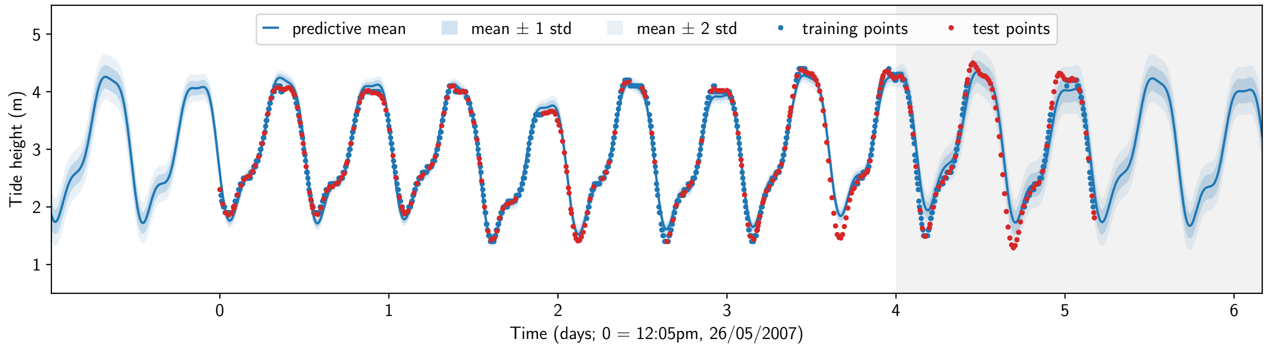

Figure 11 Finally we condition a GP on the first 4 days of training data. At this point, the posterior is almost the same as that in Figure 3.