AIMS Course 1: Data, Estimation And Inference

Jake Levi

October 2022

1 Introduction

This lab report investigates the use of Gaussian Processes (GPs), a type of machine learning model motivated by Bayesian probability theory, for modelling a meteorological dataset called Sotonmet. In a GP model, given a mean function μ, kernel function K, vaiance of observation noise σ^2, training inputs x (represented as a vector), and prediction inputs x^*, the noisy training labels y (which we assume are noisy observations of unknown labels f) and noiseless prediction labels f^* are assumed to have a joint Gaussian distribution:

$$p\left(\begin{bmatrix} y \\ f^* \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} y \\ f^* \end{bmatrix} \middle| \begin{bmatrix} \mu(x) \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} K(x,x) + \sigma^2 I & K(x,x^*) \\ K(x,x^*)^T & K(x^*,x^*) \end{bmatrix}\right) \tag{1}$$

Where $K(x, x^*)$ is a matrix whose (i, j)th element is given by $K(x, x^*)_{i,j} = K(x_i, x_j^*)$. The predictive distribution $p(f^* \mid y)$ follows from the formula for the conditional distribution of a jointly Gaussian random variable [1]:

$$p(f^* \mid y) = \mathcal{N}(f^* \mid \mu^*, \Sigma^*) \tag{2}$$

$$\text{where} \quad \mu^* = \mu(x^*) + K(x^*, x)\left(K(x,x) + \sigma^2 I\right)^{-1}(y - \mu(x)) \tag{3}$$

$$\Sigma^* = K(x^*, x^*) - K(x^*, x)\left(K(x,x) + \sigma^2 I\right)^{-1} K(x, x^*) \tag{4}$$

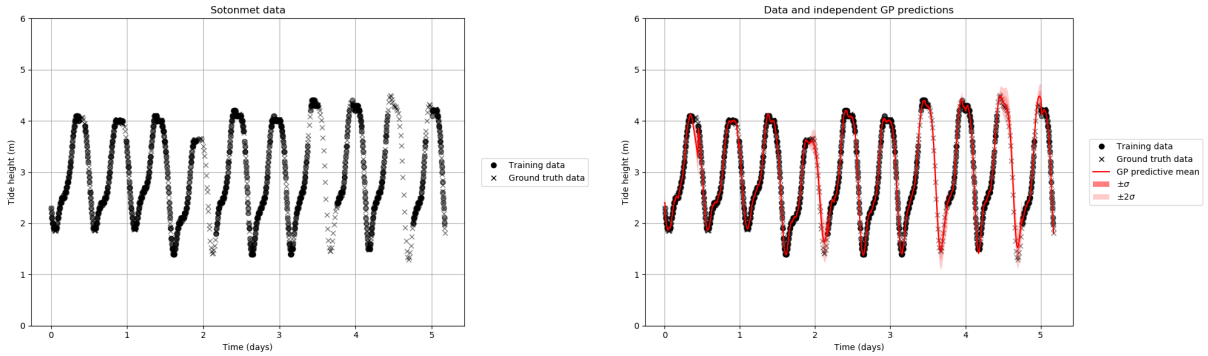
The log marginal likelihood (LML) of the noisy training labels y given training input data x (and also implicitly given any hyperparameters of the model) is given by:

$$\log\left(p(y \mid x)\right) = -\frac{1}{2}\log\left(\det\left(2\pi\Sigma_y\right)\right) - \frac{1}{2}(y - \mu(x))^T \Sigma_y^{-1}(y - \mu(x)) \tag{5}$$

$$\text{where} \quad \Sigma_y = K(x,x) + \sigma^2 I \tag{6}$$

This expression implies that maximising the LML encourages $\mu(x)$, $K(x, x)$ and σ to fit the data accurately and with calibrated uncertainty. Maximising the LML can also be motivated from a Bayesian perspective, which is discussed further in appendix A.

The focus in this coursework submission is on predicting tide height as a function of time, for which the training and ground truth data is shown in figure 1a, alongside an independent GP prediction in figure 1b.



(a) Training and ground truth data

(b) Independent GP predictions

Figure 1: The Sotonmet dataset

2 Results

2.1 Squared Exponential Kernels

We start off by considering GPs with constant mean function and squared exponential kernel, whose mean and kernel functions are related to the hyperparameters c, k, and λ as follows (σ is also considered to be a hyperparameter, although it is not explicitly part of the mean or kernel functions):

$$\mu(x) = c \tag{7}$$

$$K_{\text{sqe}}(x, x') = k \exp\left(-\left(\frac{x - x'}{\lambda}\right)^2\right) \tag{8}$$

Initially we consider two GPs denoted by sqe_1 and sqe_2, whose hyperparameter values are described in table 1a. Samples from the prior distributions of sqe_1 and sqe_1 are shown in figures 5a and 5b, which show that the prior distribution of sqe_1 looks subjectively like a much more plausible explanation for the data. The predictive distributions of these GPs are shown in figures 5c and 5d, which show that, although sqe_1 produced a *prior* distribution which looks like a more plausible explanation for the training data, sqe_2 produces a *predictive* distribution which looks like a much better fit to the training data. Furthermore, the predictive distribution of sqe_1 is "confidently wrong" (the mean is far away from the ground truth labels and with high certainty/low standard deviation) in regions containing ground truth labels but no training data, which would be a very undesirable property of a machine learning prediction model in a safety-critical context. Samples from the predictive distribution of both GPs are shown in figures 5e and 5f, which show that neither GP produces samples that reflect the data distribution particularly well.

The evaluation of GPs sqe_1 and sqe_2 according to the metrics RMSE (square root of the mean-squared-error between labels and the GP's predictive mean), LML, and log predictive likelihood (LPL, which is just the logarithm of equation 2 given the expression for a multivariate Gaussian probability density function [1]), is summarised in table 2. Note that although sqe_1 has a very low RMSE evaluated on the training data, it has an RMSE which is 30× higher when evaluated on the ground truth data, which is to say that sqe_1 overfits the training data very badly, which is reflected in its relatively bad LML and LPLs. sqe_2 has worse RMSE on the training data than the sqe_1, but better RMSE on the ground truth data, which is to say that sqe_2 generalises better to unseen data (although it does so with low confidence/high uncertainty, as seen in sqe_2's predictive samples in figure 5f), and this is reflected in the better LML and LPLs of sqe_2.

As mentioned in section 1, the LML can be used as an objective function to optimise the hyperparameters of a GP. Starting with sqe_2 (because this GP has greater LML than sqe_1), the parameters of this GP can be optimised using the L-BFGS-B algorithm [4], leading to a GP referred to as sqe_opt, whose hyperparameter values are described in table 1a, and whose evaluation metrics are described in table 2. Remarkably, sqe_opt has worse RMSE on *training* data than sqe_1, but better RMSE on *unseen* ground truth data. This implies that optimising RMSE on training data (which is often performed in machine learning) is not always a good approach, because an improvement in the RMSE on training data might lead to decreased predictive performance on unseen data (which is generally of greater importance in machine learning), and that where possible, a better approach might be to optimise LML instead. The sensitivity of the LML of sqe_opt to its hyperparameters is shown in figure 6, which shows that sqe_opt is very sensitive to large values of λ and small values of σ. The predictive distribution of sqe_opt is shown in figure 7a.

2.2 Epistemic And Aleatoric Uncertainty

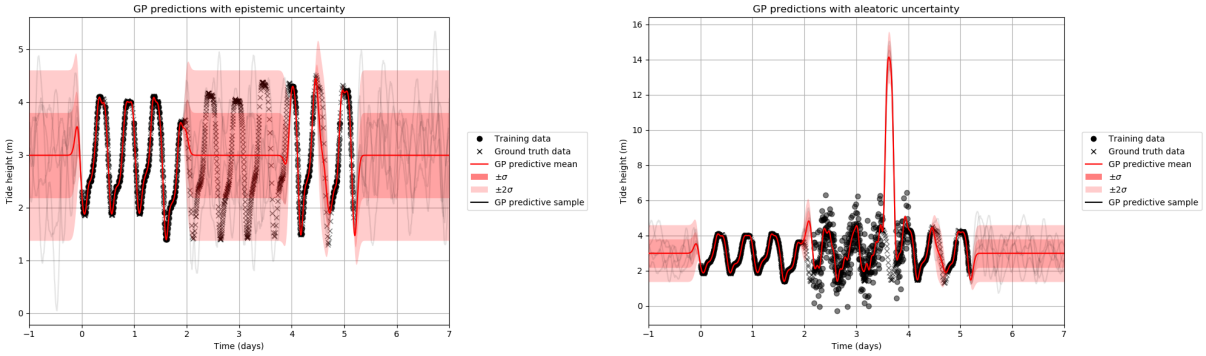
To quote Alex Kendall and Yarin Gal in [2]:

> There are two major types of uncertainty one can model. Aleatoric uncertainty captures noise inherent in the observations. On the other hand, epistemic uncertainty accounts for uncertainty in the model - uncertainty which can be explained away given enough data.

We can model the performance of a GP in the presence of epistemic or aleatoric uncertainty by either removing a subsection of the data or by artificially adding noise to a subsection of the data respectively. In the case of sqe_opt (which was optimised to have high LML), the results of two such experiments are shown in figure 2.

Although this GP performs well in the presence of epistemic uncertainty, reverting to a larger predictive standard deviation when far from the vicinity of any training data, we see that this GP does not perform well in the presence of aleatoric uncertainty, making confidently wrong predictions (predictions which are far away from the ground truth labels and with high certainty/low standard deviation) in the vicinity of training data which has a high degree of noise.

We can understand this behaviour by looking at the expression for the predictive variance of a GP in equation 4, which depends only on the input locations of the training data and predictions, and not on the labels of the training



(a) Epistemic

(b) Aleatoric

Figure 2: The performance of sqe_opt with epistemic and aleatoric uncertainty

data. Of course, the predictive variance of sqe_opt considered here depends indirectly on the training labels, as a result of its hyperparameters having been optimised with respect to the LML of the training data, however this model has no capacity to increase its predictive uncertainty in the presence of unseen noisy training labels.

This could be a very undesirable property for the model to have in a safety-critical prediction context, for example if one of the input sensors failed and started producing very noisy measurements, we would not want the model to produce wildly incorrect predictions with a high degree of certainty, rather we would prefer the model to increase its predictive uncertainty to fit the noise in the newly observed data. One possible solution to this problem would be to model the observation noise (which does directly affect the predictive uncertainty of a GP) using a second GP, which predicts observation noise as a function of the same input data as the original GP, and conditions on the estimated noise of the training data, however we leave this as a direction for future work.

2.3 Periodic Kernels

A periodic kernel for a GP can be defined as follows:

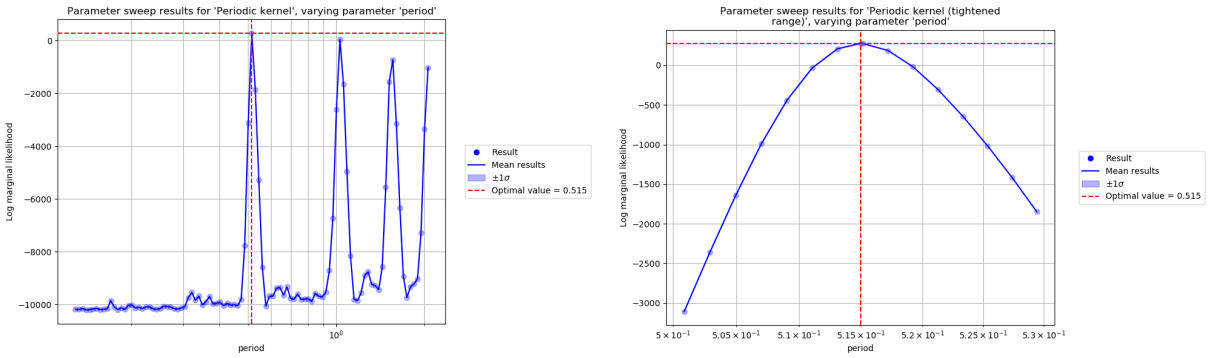
$$K_{\text{per}}(x, x') = k \exp\left(-\frac{2}{\lambda}\left(\sin\left(\pi\frac{x - x'}{T}\right)\right)^2\right) \tag{9}$$

We define a new GP with a periodic kernel, set its hyperparameters to be identical to those of sqe_opt, except that $T = 0.5$, and denote this GP as per_1. We then optimise the LML of this GP with respect to its hyperparameters as before, leading to a GP denoted by per_opt. The hyperparameters of both GPs are summarised in table 1b, evaluation metrics are summarised in table 2, and the predictive distribution of per_opt is shown in figure 7b. We see that per_1 has a very bad LML (worse than sqe_2, but not as bad as sqe_1), and that per_opt has a LML which is better than sqe_2, but not as good as sqe_opt.

We note per_opt learns a much greater value of the observation noise σ than did sqe_opt, and also, looking at figure 7b, that the predictive standard deviation of per_opt (shown as the red shaded region, which predicts the standard deviation of the unknown label $f*$ without observation noise) is very low, whereas the samples from the predictive distribution of per_opt (which do include observation noise) are much noisier. We can understand this behaviour by considering the expression for the periodic kernel function (equation 9), and interpreting the behaviour in terms of epistemic uncertainty, described in section 2.2. To the periodic kernel, two training points might be separated by a long time lag, but if they are separated by an exact (or close to exact) integer multiple of the period T then they will be interpreted by the periodic kernel as occuring at the same (or at a nearby) input location.

For the Sotonmet data considered here, the training data is spread over ten periods of high/low tide, but using the periodic kernel, this data would be processed in exactly the same way if all 917 training points were shifted by an integer multiple of the period to lie in the first half day, in which case we would would have a large number of training points all in a very close vicinity to each other. In section 2.2, we saw that the predictive uncertainty increases when the number of nearby data points decreases (this is epistemic uncertainty), and here we see the inverse effect, that the predictive uncertainty decreases when, from the perspective of the kernel, the effective number of nearby data points increases. This explains why per_opt learns a very high value of σ: to compensate for the high certainty it has about the value of the underlying noiseless variable f, as a result of effectively observing a larger number of training points in the nearby vicinity.

The sensitivity of the LML of per_opt to the period, T, is shown in figure 3, which shows that the LML is very sensitive to and sharply peaked around the optimal value of T. In general, it might be difficult for a gradient-based



(a) Zoomed out

(b) Zoomed in

Figure 3: Sensitivity of the LML of per_opt to the period, T

optimisation algorithm to find the optimal value of a hyperparameter when the LML is so sharply peaked. A possible solution to this problem would be to model the LML as a function of the hyperparameters using a second GP, and use Bayesian Optimisation [3] to optimise the hyperparameters, but this raises the question of how to optimise the hyperparameters of the second GP. Clearly one cannot use a separate Bayesian Optimisation routine to optimise the hyperparameters of every GP without requiring an infinite number of GPs.

2.4 Sum And Product Kernels

Given any finite set $\mathcal{K}$ of valid GP kernel functions, we can define a new function $K'(x, x')$ whose output is equal the sum or the product of the outputs of all GP kernel functions in $\mathcal{K}$, and $K'(x, x')$ will also be a valid GP kernel function [1]. Sums and products of kernels correspond to the assumptions that covariance is dependent on similarity under *any* kernel in $\mathcal{K}$ or on similarity under *all* kernels in $\mathcal{K}$, respectively. This naturally leads us to consider two new kernel functions K_{sum} and K_{prod}, equal to the sum and the product respectively of squared exponential and periodic kernels. These kernel functions are defined below in terms of their hyperparameters:

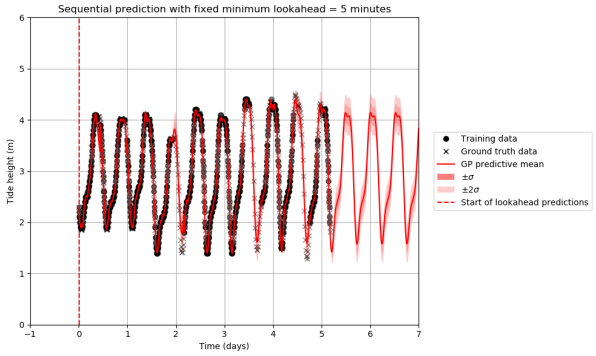
$$K_{\text{sum}}(x, x') = k_{\text{sqe}} \exp\left(-\left(\frac{x - x'}{\lambda_{\text{sqe}}}\right)^2\right) + k_{\text{per}} \exp\left(-\frac{2}{\lambda_{\text{per}}}\left(\sin\left(\pi\frac{x - x'}{T}\right)\right)^2\right) \tag{10}$$

$$K_{\text{prod}}(x, x') = k_{\text{sqe}} \exp\left(-\left(\frac{x - x'}{\lambda_{\text{sqe}}}\right)^2\right) \times k_{\text{per}} \exp\left(-\frac{2}{\lambda_{\text{per}}}\left(\sin\left(\pi\frac{x - x'}{T}\right)\right)^2\right) \tag{11}$$

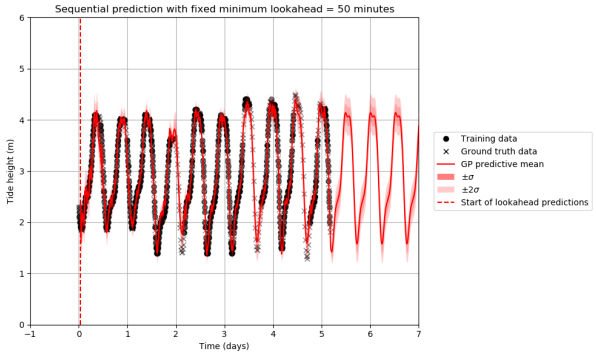
We define GPs sum_1 and prod_1, having the same hyperparameter values as the corresponding hyperparameters in sqe_opt and per_opt (tables 1a and 1b), except that we use the value of σ used by per_opt, since this is larger than the value of σ used by sqe_opt, and the LML of a GP is generally more sensitive to small values of σ than large values, as demonstrated in figure 6d. We then optimise the hyperparameters of GPs sum_1 and prod_1, leading to GPs sum_opt and prod_opt, respectively. The hyperparameter values of these four GPs are shown in tables 1c and 1d, their evaluation metrics are shown in table 2, and the predictive distributions of sum_opt and prod_opt are shown in figures 7d and 7c, respectively. GP sum_opt has the best LML out of all the GPs considered in this report, although interestingly prod_opt has a better RMSE evaluated on unseen ground truth data. Regarding sqe_1, the first GP considered in this report, no other GP considered has better RMSE evaluated on training data, or worse RMSE evaluated on unseen ground truth data, which highlights the problems associated with overfitting the training data, as discussed in section 2.1. Regarding predictive distributions, sum_opt has the best combination of predicting the periodic nature of tide heights far from the training data, while also accurately modelling local variations in tide height.

2.5 Sequential Prediction

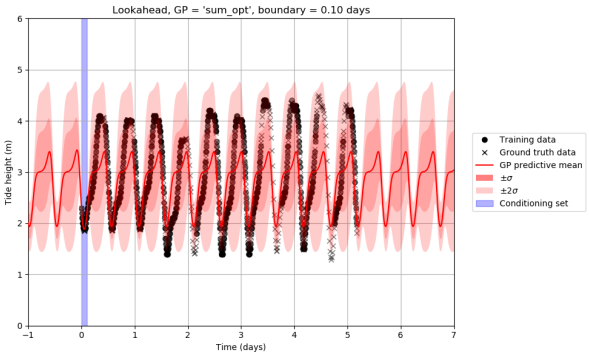
In this section we consider sequential prediction, in particular how well a GP can predict future tide heights given only historical data, by restricting the data available to the GP in different ways. Initially we consider sequential prediction with a fixed lookahead period, $t_{\text{lookahead}}$, in which the prediction at any time t^* uses only training data with input time values less than or equal to $t^* - t_{\text{lookahead}}$, such that predictions at subsequent values of t^* use progressively larger datasets. The results of sequential predictions with lookahead periods of 5 minutes and 50 minutes using GP sum_opt are shown in figures 4a and 4b, demonstrating the model's ability to make predictions at a fixed time period in the future. As shown in the figures, for larger $t_{\text{lookahead}}$, the predictions become more noisy, especially at earlier values of t^*.



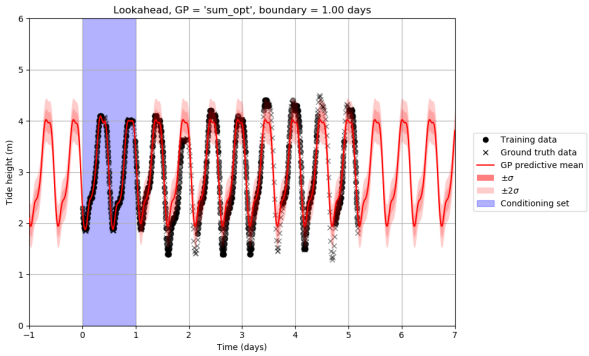
(a) 5 minute fixed lookahead



(b) 50 minute fixed lookahead



(c) Predictions after 0.1 days



(d) Predictions after 1 day

Figure 4: Sequential predictions of GP sum_opt (see table 1c) using different fixed and varying lookaheads

Lastly, we consider future predictions in which the size of the subset of data on which the GP is conditioned remains fixed, using data points whose time values are between $t = 0$ days and some limit t_{lim} to form the entire predictive distribution, and consider how that predictive distribution as a function of time varies with t_{lim}. Figures 4c and 4d show the predictive distribution of sum_opt as a function of time for $t_{\text{lim}} = 0.1$ days and $t_{\text{lim}} = 1$ days respectively, with the region $[0, t_{\text{lim}}]$ on the time axis shaded in blue. As shown in the figures, for $t_{\text{lim}} = 0.1$, the predictive distribution is highly uncertain, but for $t_{\text{lim}} = 1$, the predictive distribution for all future days has reasonably high accuracy and calibrated uncertainty, given that sum_opt has only oserved one day's worth of data. These statements could be quantified by considering how the RMSE and LPL on unseen ground truth data vary as a function of t_{lim}, however this is left as a direction for future work. Animations of GP predictive distributions for varying t_{lim} are included in the GitHub repository for this project[1], in particular for GPs sqe_opt, prod_opt, and sum_opt.

3 Conclusions

In this report, Gaussian Processes (GPs) were introduced, and expressions for the GP joint and predictive distributions and the log marginal likelihood (LML) were stated. The Sotonmet dataset of meteorological data was also introduced, in particular the data for tide height as a function of time, shown in figure 1a, which was used to investigate the properties of regression with GPs. Three GPs with squared exponential kernel were considered and evaluated according to different metrics, and the problem of overfitting was discussed, as was the sensitivity of the LML to different hyperparameters. Epistemic and aleatoric uncertainty were discussed, and it was found that a simple GP with optimised hyperparameters performs well under epistemic uncertainty, but not under aleatoric uncertainty, which could be problematic in a safety-critical prediction context. Two GPs with periodic kernels were considered and evaluated, and the tendency of the periodic kernel to assume high certainty about underlying variables and also to assume high observation noise was interpreted in terms of epistemic uncertainty. The problem of optimising the LML with respect to the period of the kernel (to which the LML is very sensitive) was considered, as was the possibility of applying Bayesian optimisation to this problem. Sum and product kernels were discussed, two GPs with each type of kernel were considered and evaluated, and it was found that a GP whose kernel function was the sum of a squared exponential kernel and a periodic kernel achieved the best LML, as well as the qualitatively most plausible looking predictive distribution. Lastly sequential prediction was discussed, in the two cases of prediction with a fixed lookahead period, and prediction using a fixed time-bounded subset of the training data.

[1] `https://github.com/jakelevi1996/aims/tree/main/scripts/course_1_dei`

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[2] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[3] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[4] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

A Motivation For Maximising The Log Marginal Likelihood

In a Bayesian regression problem, we generally have a set of weights w, a set of data $\mathcal{D}$, and a set of hyperparameters θ, for which the posterior, likelihood, prior, and marginal likelihood distributions are related by Bayes' rule:

$$p(w \mid \mathcal{D}, \theta) = \frac{p(\mathcal{D} \mid w, \theta)p(w \mid \theta)}{p(\mathcal{D} \mid \theta)} \tag{12}$$

The predictive distribution of an unknown target y^* given the dataset and hyperparameters is then found by marginalising with respect to the posterior distribution of the weights:

$$p(y^* \mid \mathcal{D}, \theta) = \int dw \left[p(y^*, w \mid \mathcal{D}, \theta)\right] \tag{13}$$

$$= \int dw \left[p(y^* \mid w, \mathcal{D}, \theta)p(w \mid \mathcal{D}, \theta)\right] \tag{14}$$

Specifically this gives us the predictive distribution of the target y^* given that we know the "correct" values of the hyperparameters θ, however in general this is not the case, and in a "truly" Bayesian approach, we should marginalise over the hyperparameters as well:

$$p(y^* \mid \mathcal{D}) = \int d\theta \left[p(y^*, \theta \mid \mathcal{D})\right] \tag{15}$$

$$= \int d\theta \left[p(y^* \mid \mathcal{D}, \theta)p(\theta \mid \mathcal{D})\right] \tag{16}$$

$$= \int d\theta \left[p(y^* \mid \mathcal{D}, \theta)\left(\frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}\right)\right] \tag{17}$$

We denote value of the hyperparameters which maximise the marginal likelihood by $\hat{\theta}$, and if we assume that the marginal likelihood is so sharply peaked around $\hat{\theta}$ that it can be approximated by a delta function, as well as assuming that the prior distribution over hyperparameters is uninformative, then we obtain the following simplification:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}[p(\mathcal{D} \mid \theta)] \tag{18}$$

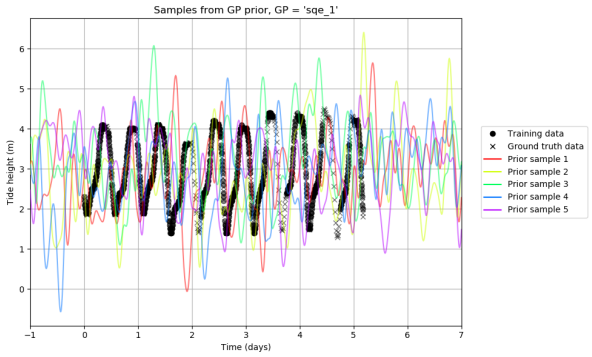
$$p(\mathcal{D} \mid \theta) \approx \delta(\theta - \hat{\theta}) \tag{19}$$

$$\Rightarrow p(y^* \mid \mathcal{D}) \approx p(y^* \mid \mathcal{D}, \hat{\theta}) \tag{20}$$

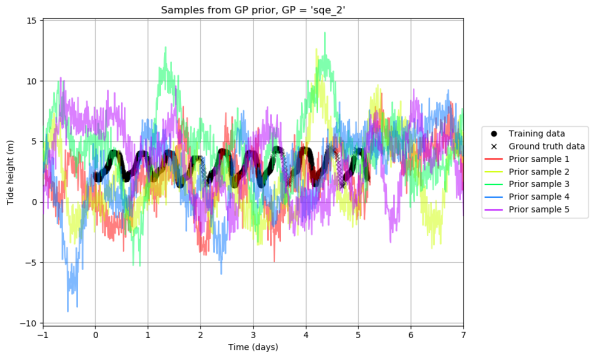
Therefore, assuming we have found $\hat{\theta}$ which maximises the marginal likelihood, then $\hat{\theta}$ is the best point estimate of the hyperparameters for approximating the marginalised predictive distribution $p(y^* \mid \mathcal{D})$.

The situation is the same for GPs, except that instead of defining distributions over weights, we define a joint distribution over predictions and observations directly, and derive the predictive distribution using the product rule:

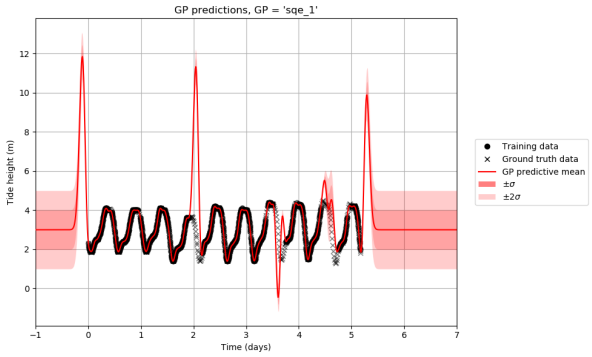
$$p(y^* \mid \mathcal{D}, \theta) = \frac{p(y^*, \mathcal{D} \mid \theta)}{p(\mathcal{D} \mid \theta)} \tag{21}$$



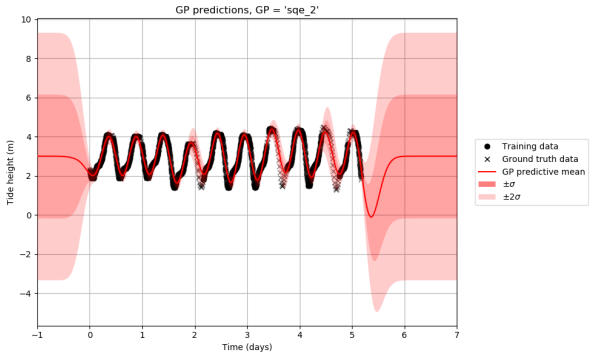
(a) Samples from the prior of sqe_1



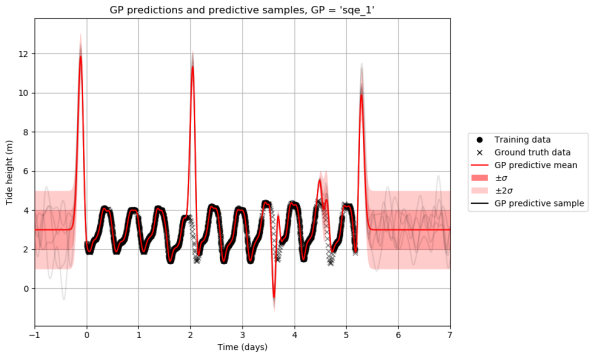
(b) Samples from the prior of sqe_2



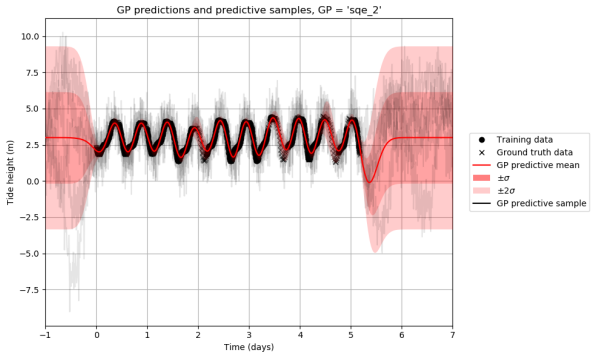
(c) Predictive distribution of sqe_1



(d) Predictive distribution of sqe_2



(e) Samples from the predictive distribution of sqe_1



(f) Samples from the predictive distribution of sqe_2

Figure 5: Prior and predictive distributions of GPs with square exponential kernels

Hyperparameter	sqe_1	sqe_2	sqe_opt
c	3.0000	3.0000	2.9905
λ	0.1000	0.3000	0.0867
k	1.0000	10.0000	0.6522
σ	0.0010	1.0000	0.0293

(a) Squared exponential

Hyperparameter	per_1	per_opt
c	2.9905	2.9945
λ	0.0867	1.2264
k	0.6522	1.0346
T	0.5000	0.5149
σ	0.0293	0.1733

(b) Periodic

Hyperparameter	sum_1	sum_opt
c	2.9945	3.0000
λ_{sqe}	0.0867	0.0692
k_{sqe}	0.6522	0.0299
λ_{per}	1.2264	0.6453
k_{per}	1.0346	0.5962
T	0.5149	0.5150
σ	0.1733	0.0287

(c) Sum

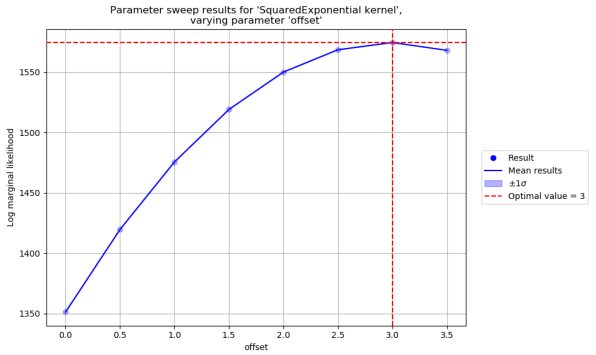
Hyperparameter	prod_1	prod_opt
c	2.9945	2.9286
λ_{sqe}	0.0867	0.7881
k_{sqe}	0.6522	5.5389
λ_{per}	1.2264	0.7271
k_{per}	1.0346	0.0975
T	0.5149	0.5082
σ	0.1733	0.0291

(d) Product

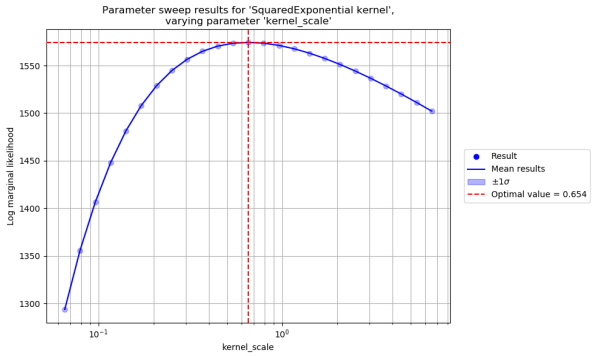
Table 1: Values of the hyperparameters of GPs with different kernel functions

Metric	sqe_1	sqe_2	sqe_opt	per_1	per_opt	sum_1	prod_1	sum_opt	prod_opt
RMSE (train)	<u>0.0268</u>	0.2246	0.0275	0.5211	0.1724	0.0299	0.0286	0.0270	0.0272
RMSE (truth)	0.8040	0.2573	0.1587	0.5355	0.1751	0.0402	0.1783	0.0397	<u>0.0319</u>
LML	-327743.8	-942.0	1574.4	-142748.1	276.3	544.8	499.3	<u>1672.9</u>	1621.6
LPL (train)	-321611.9	-875.4	1954.9	-142566.3	307.6	720.7	715.7	<u>1971.2</u>	1965.6
LPL (truth)	-87596.3	-894.3	3366.3	-205771.5	723.7	1294.7	1277.4	<u>3432.4</u>	3411.4

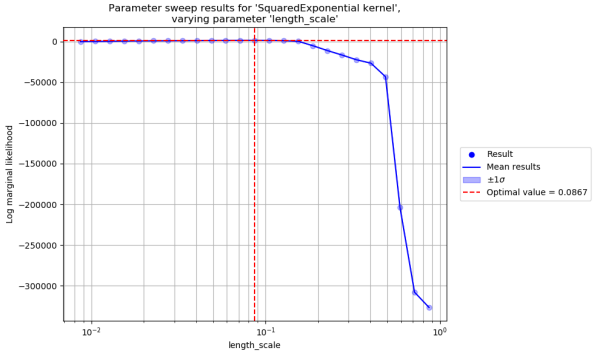
Table 2: Evaluations of GPs according to different metrics



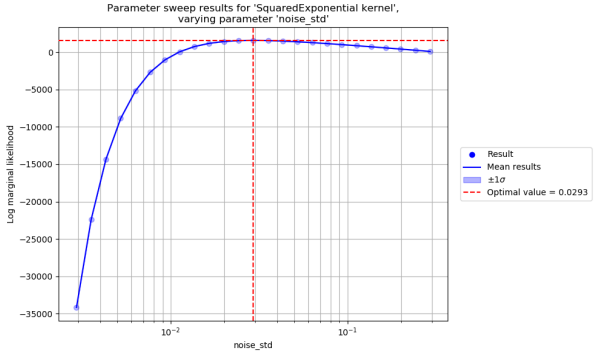
(a) c



(b) k

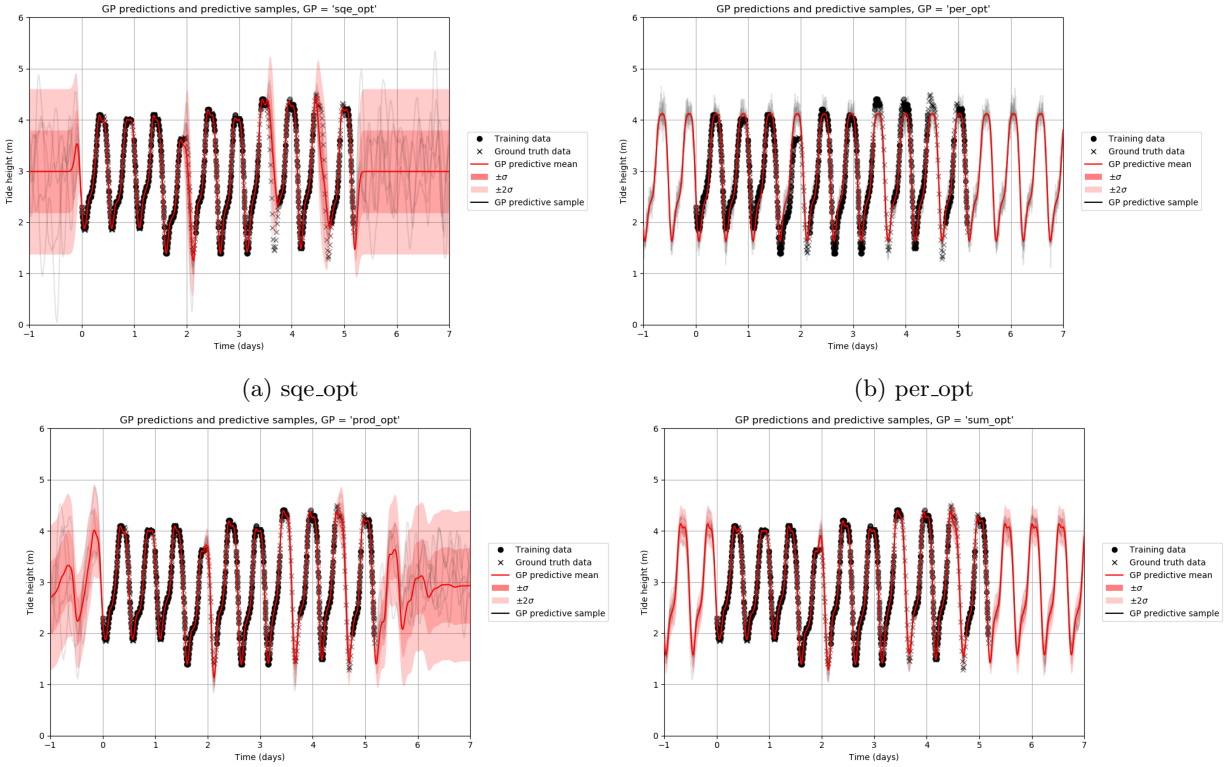


(c) λ



(d) σ

Figure 6: Sensitivity of the LML of sqe_opt to its hyperparameters



(a) sqe_opt

(b) per_opt

(c) prod_opt

(d) sum_opt

Figure 7: Predictive distributions of GPs with different kernel functions and optimised hyperparameters