

Parameter estimation and prediction using Gaussian Processes

Yiannis Andrianakis and Peter G. Challenor

August 20, 2009

Abstract

This is a report on methods of parameter estimation and prediction for Gaussian Process emulators. The parameters considered are the regression coefficients β , the scaling parameter σ^2 and the correlation lengths δ . Starting from a maximum likelihood estimate of the three parameters we show that by marginalising β and σ^2 out we can derive REML and what we call here the ‘Toolkit’ approach. Focusing on correlation lengths we then investigate the effect of the number of design points to their estimation, as a function of the input’s dimensionality and the inherent correlation length of the data. A comparison between the ML and REML methods is then given, before discussing methods for accounting for our uncertainty about δ . We propose the use of two correlation length priors that result in proper posterior distributions. The first is the reference prior and the second is based on a transformation of the inverse Gamma distribution. Finally, we discuss two methods for marginalising the correlation lengths.

1 Parameter estimation

1.1 Model setup

Before starting, let us define the framework on which we will base the subsequent analysis. We define the following

- n design points $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ with outputs $y = [y_1, y_2, \dots, y_n]$
- A point \mathbf{x} where we want to predict the emulator’s output, denoted by $\eta(\mathbf{x})$
- A correlation function $c(\mathbf{x}, \mathbf{x}')$ between \mathbf{x} and \mathbf{x}'
- The correlation between the design and prediction points

$$\mathbf{c}(\mathbf{x}) = [c(\mathbf{x}, \mathbf{x}_1), c(\mathbf{x}, \mathbf{x}_2), \dots, c(\mathbf{x}, \mathbf{x}_n)]^T$$

- The design points correlation matrix

$$A = \begin{pmatrix} 1 & c(\mathbf{x}_1, \mathbf{x}_2) & \dots & c(\mathbf{x}_1, \mathbf{x}_n) \\ c(\mathbf{x}_2, \mathbf{x}_1) & 1 & \dots & c(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ c(\mathbf{x}_n, \mathbf{x}_1) & \dots & \dots & 1 \end{pmatrix}$$

- A $(q \times 1)$ vector of regression functions for \mathbf{x} , denoted as $h(\mathbf{x})$
- A matrix of regression functions for the design points

$$H = [h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_n)]^T$$

The model we assume for our data is

$$y = h(\mathbf{x})^T \beta + f \quad (1)$$

In this equation, β is a vector of the regression coefficients and f is a Gaussian process. More specifically we assume that f is zero mean and its members have covariance function $\sigma^2 c(\mathbf{x}, \mathbf{x}')$, where σ^2 is a scale hyperparameter. In the following we will assume a specific form for the correlation function $c(\mathbf{x}, \mathbf{x}')$, but much of the discussion will be applicable in the case a different correlation function is used. The correlation function we consider here, is the Gaussian, which is given by

$$c(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p \exp \left\{ -\frac{(x_i - x'_i)^2}{\delta_i^2} \right\} \quad (2)$$

x_i is the i^{th} element (input) of \mathbf{x} and similarly, δ_i is the i^{th} element of the vector of hyperparameters δ , also known as correlation lengths. The number of inputs is denoted by p .

We now discuss different approaches to estimating the hyperparameters β , σ^2 and δ and give the respective equations for predicting the simulator's output in an arbitrary input point \mathbf{x} , once the y_1, \dots, y_n have been observed.

1.2 Maximum Likelihood

The likelihood of the parameters given the observed data is

$$p(y|\beta, \sigma^2, \delta) = \frac{|A|^{-1/2}}{(2\pi\sigma^2)^{n/2}} \exp \left\{ -\frac{1}{2\sigma^2} (y - H\beta)^T A^{-1} (y - H\beta) \right\} \quad (3)$$

The three hyperparameters can be estimated by maximising the above expression. The estimates for β , σ^2 can be found in a closed form and are

$$\hat{\beta}_{ML} = (H^T A^{-1} H)^{-1} H^T A^{-1} y \quad (4)$$

$$\hat{\sigma}_{ML}^2 = \frac{(y - H\hat{\beta}_{ML})^T A^{-1} (y - H\hat{\beta}_{ML})}{n} \quad (5)$$

An estimate for δ can be found by maximising the likelihood after plugging in the ML estimates for β and σ^2 .

$$\hat{\delta}_{ML} = \arg \max_{\delta} [p(y|\hat{\beta}_{ML}, \hat{\sigma}_{ML}^2, \delta)] \quad (6)$$

The distribution of the output $\eta(\mathbf{x})$ given the observations and the parameters, will be

$$p(\eta(\mathbf{x})|\hat{\beta}_{ML}, \hat{\sigma}_{ML}^2, \hat{\delta}_{ML}, y) = \mathcal{N}(m_{ML}(\mathbf{x}), \hat{\sigma}_{ML}^2 u_{ML}(\mathbf{x}, \mathbf{x})) \quad (7)$$

where

$$m_{ML}(\mathbf{x}) = h(\mathbf{x})^T \hat{\beta}_{ML} + \mathbf{c}(\mathbf{x})^T A^{-1} (y - H\hat{\beta}_{ML})$$

$$u_{ML}(\mathbf{x}, \mathbf{x}) = c(\mathbf{x}, \mathbf{x}) - \mathbf{c}(\mathbf{x})^T A^{-1} \mathbf{c}(\mathbf{x})$$

Note that A , $c(\mathbf{x}, \mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are all functions of $\hat{\delta}_{ML}$.

1.3 Restricted Maximum Likelihood

According to Harville [1], the Restricted Maximum Likelihood (REML) can be obtained from Maximum Likelihood, by integrating β using a uniform prior (i.e. $p(\beta) \propto \text{constant}$). The resulting expression is

$$p(y|\sigma^2, \delta) \propto \frac{|A|^{-1/2}|H'A^{-1}H|^{-1/2}}{(2\pi\sigma^2)^{\frac{n-q}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(y-H\hat{\beta})'A^{-1}(y-H\hat{\beta})\right\} \quad (8)$$

The value of σ^2 that maximises the above expression is

$$\hat{\sigma}_{RL}^2 = \frac{(y-H\hat{\beta})'A^{-1}(y-H\hat{\beta})}{n-q} \quad (9)$$

An estimate for δ is then given by

$$\hat{\delta}_{RL} = \arg \max_{\delta} [p(y|\hat{\sigma}_{RL}^2, \delta)] \quad (10)$$

In the above we also define

$$\hat{\beta} = (H^T A^{-1} H)^{-1} H^T A^{-1} y \quad (11)$$

Strictly speaking, $\hat{\beta}$ cannot be considered as an estimate for β , because β has been integrated out. We could have used any symbol for the above expression, but because it is identical to $\hat{\beta}_{ML}$ we call it $\hat{\beta}$ (but not $\hat{\beta}_{RL}$).

The distribution of the output $\eta(\mathbf{x})$ given the observations and the parameters, will be

$$p(\eta(\mathbf{x})|\hat{\sigma}_{RL}^2, \hat{\delta}_{RL}, y, \cdot) = \mathcal{N}(m_{RL}(\mathbf{x}), \hat{\sigma}_{RL}^2 u_{RL}(\mathbf{x}, \mathbf{x})) \quad (12)$$

where

$$\begin{aligned} m_{RL}(\mathbf{x}) &= h(\mathbf{x})^T \hat{\beta} + \mathbf{c}(\mathbf{x})^T A^{-1} (y - H\hat{\beta}) \\ u_{RL}(\mathbf{x}, \mathbf{x}) &= c(\mathbf{x}, \mathbf{x}) - \mathbf{c}(\mathbf{x})^T A^{-1} \mathbf{c}(\mathbf{x}) \\ &\quad + (h(\mathbf{x})^T - \mathbf{c}(\mathbf{x})^T A^{-1} H) (H^T A^{-1} H)^{-1} (h(\mathbf{x})^T - \mathbf{c}(\mathbf{x})^T A^{-1} H)^T \end{aligned}$$

Note that $m_{RL}(\mathbf{x}) = m_{ML}(\mathbf{x})$. Note also that the terms in the first line of $u_{RL}(\mathbf{x}, \mathbf{x})$ are the same as those in $u_{ML}(\mathbf{x}, \mathbf{x})$, except that A , $c(\mathbf{x}, \mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are now calculated with $\hat{\delta}_{RL}$ instead of $\hat{\delta}_{ML}$. Other than that, $u_{RL}(\mathbf{x}, \mathbf{x})$ can be considered as $u_{ML}(\mathbf{x}, \mathbf{x})$ augmented by the expression of the last line in the above equation, that accounts for the increase in the variance, due to our uncertainty about β . Both predictions for the output $\eta(\mathbf{x})$ however, are Gaussian.

1.4 The Toolkit approach

The toolkit approach can be obtained from REML if we account for the uncertainty in σ^2 , by integrating it out, using a prior $p(\sigma^2) \propto \sigma^{-2}$. The marginal likelihood is then

$$p(y|\delta) \propto \left[(y-H\hat{\beta})'A^{-1}(y-H\hat{\beta}) \right]^{-\frac{n-q}{2}} |A|^{-1/2} |H'A^{-1}H|^{-1/2} \quad (13)$$

where $\hat{\beta}$ is the same as in REML. An estimate for δ can be found by

$$\hat{\delta}_{TL} = \arg \max_{\delta} [p(y|\delta)] \quad (14)$$

The distribution of the output $\eta(\mathbf{x})$ given the observations and $\hat{\delta}_{TL}$ is given by

$$p(\eta(\mathbf{x})|\hat{\delta}_{TL}, \mathbf{y}) \propto \left(1 + \frac{(\eta(\mathbf{x}) - m_{TL}(\mathbf{x}))^2}{(n - q)\hat{\delta}^2 u_{TL}(\mathbf{x}, \mathbf{x})}\right)^{-\frac{n-q+1}{2}} \quad (15)$$

which is a t-student distribution with $n - q$ degrees of freedom. In other words

$$\frac{\eta(\mathbf{x}) - m_{TL}(\mathbf{x})}{\sqrt{\hat{\delta}^2 u_{TL}(\mathbf{x}, \mathbf{x})}} \sim t_{n-q} \quad (16)$$

In the above,

$$\hat{\sigma}^2 = \frac{(y - H\hat{\beta})^T A^{-1}(y - H\hat{\beta})}{n - q} \quad (17)$$

which implies that $\hat{\sigma}^2 = \hat{\sigma}_{RL}^2$. Again, $\hat{\sigma}^2$ cannot strictly be considered as an estimate of σ^2 , as it was marginalised. (See also the discussion about $\hat{\beta}$ and $\hat{\beta}_{ML}$.) The expressions for $m_{TL}(\mathbf{x})$ and $u_{TL}(\mathbf{x}, \mathbf{x})$ are the same as $m_{RL}(\mathbf{x})$ and $u_{RL}(\mathbf{x}, \mathbf{x})$.

1.4.1 Comparison between REML and the Toolkit approach

The Toolkit approach is very closely related to REML. We will now show that they are only different with respect to the prediction variance, and in the limit $(n - q) \rightarrow \infty$ they become identical.

If we substitute $\hat{\sigma}_{RL}^2$ from 9 in 8, we can see that the REML and Toolkit cost functions become proportional. This implies that the estimates for δ are not affected by the integration of σ^2 , or in other words, accounting for our uncertainty in σ^2 does not influence our estimates of δ .

We therefore establish that $\hat{\delta}_{RL} = \hat{\delta}_{TL}$. Consequently, it will also hold that $m_{TL} = m_{RL}$ and $u_{TL} = u_{RL}$. As a result the prediction means will be identical between the Toolkit and the REML approaches. The variance of the Toolkit prediction will be larger, because for the same values of σ^2 and u , the t-distribution in (15) has larger variance than the Gaussian in (12). More precisely, the variance of the t-distribution will be $\frac{n-q}{n-q-2}$ times the variance of the Gaussian. Therefore, as the term $n - q$ increases, or in other words as the observations become a lot more than the regression coefficients, the t-distribution in (15) will converge to the Gaussian in (12) and the predictions will be the same both in terms of the mean and the variance.

1.5 Estimation of the correlation lengths

1.5.1 Reparameterisation

We previously gave the expressions that need to be maximised for estimating the correlation lengths δ . Here we propose a reparameterisation that can help with the maximisation process. This is $\tau = -2 \ln(\delta)$. The expressions that need to be maximised are identical, the only difference being that the correlation function now reads

$$c(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p \exp \{ -(x_i - x'_i)^2 e^{\tau_i} \} \quad (18)$$

Note also that A and $\mathbf{c}(\mathbf{x})$ are also functions of $c(\mathbf{x}, \mathbf{x}')$. An advantage of this reparameterisation is that the optimisation problem is now unconstrained, since δ can only take values in $(0, \infty)$, while τ can take values in $(-\infty, \infty)$. A second benefit is that the likelihood comes closer to a Gaussian distribution after the transformation at least in low dimensions, as it is claimed in [2].

1.5.2 Derivatives

While the maximisation of the various cost functions can be achieved with derivative-free optimisation methods (e.g. Nelder-Mead), the existence of closed form expressions for the derivatives of the cost functions could benefit the maximisation process. We now give the two first derivatives of the Toolkit's log likelihood (13) w.r.t. τ . Incidentally, since REML (8) becomes identical to (13) after substitution of the $\hat{\sigma}_{RL}^2$, the expressions for the derivatives will be applicable to REML as well. The two first derivatives are:

$$\frac{\partial(\ln(p(y|\tau)))}{\partial\tau_k} = -\frac{1-n+q}{2}\text{tr}\left[P\frac{\partial A}{\partial\tau_k}\right] - \frac{n-q}{2}\text{tr}\left[R\frac{\partial A}{\partial\tau_k}\right]$$

$$\frac{\partial(\ln(p(y|\tau)))}{\partial\tau_l\partial\tau_k} = -\frac{1-n+q}{2}\text{tr}\left[P\frac{\partial A}{\partial\tau_l\partial\tau_k} - P\frac{\partial A}{\partial\tau_l}P\frac{\partial A}{\partial\tau_k}\right] - \frac{n-q}{2}\text{tr}\left[R\frac{\partial A}{\partial\tau_l\partial\tau_k} - R\frac{\partial A}{\partial\tau_l}R\frac{\partial A}{\partial\tau_k}\right]$$

with

$$P \equiv A^{-1} - A^{-1}H(H^T A^{-1}H)^{-1}H^T A^{-1}$$

and

$$R \equiv P - Py(y^T Py)^{-1}y^T P$$

To find the derivatives of the correlation matrix A w.r.t. τ , we first denote its $(\mu, \nu)^{\text{th}}$ element as

$$A(\mu, \nu) = c(\mathbf{x}_\mu, \mathbf{x}_\nu) = \prod_{i \in p} \exp\{-(x_{i,\mu} - x_{i,\nu})^2 e^{\tau_i}\} \quad (19)$$

In the following, the subscripts μ, ν denote design points and the subscripts i, k, l index the inputs. The first derivative is

$$\begin{aligned} \frac{\partial A(\mu, \nu)}{\partial\tau_k} &= \prod_{i \in p} \exp\{-(x_{i,\mu} - x_{i,\nu})^2 e^{\tau_i}\} [-(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] \\ &= A(\mu, \nu) [-(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] \end{aligned} \quad (20)$$

The second derivative

$$\begin{aligned} \frac{\partial^2 A(\mu, \nu)}{\partial\tau_l\partial\tau_k} &= \prod_{i \in p} \exp\{-(x_{i,\mu} - x_{i,\nu})^2 e^{\tau_i}\} [(x_{l,\mu} - x_{l,\nu})^2 e^{\tau_l}] [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] \\ &= A(\mu, \nu) [(x_{l,\mu} - x_{l,\nu})^2 e^{\tau_l}] [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] \end{aligned} \quad (21)$$

and finally

$$\begin{aligned} \frac{\partial^2 A(\mu, \nu)}{\partial\tau_k\partial\tau_k} &= \prod_{i \in p} \exp\{-(x_{i,\mu} - x_{i,\nu})^2 e^{\tau_i}\} [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k} - 1] \\ &= A(\mu, \nu) [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k}] [(x_{k,\mu} - x_{k,\nu})^2 e^{\tau_k} - 1] \end{aligned} \quad (22)$$

An implementation of a Gaussian quadrature algorithm using the above derivatives typically converged to the maximum in less than 10 iterations, while the Nelder-Mead algorithm that did not use derivative information, with the same initialisation conditions needed more than 300. Additionally, even though the calculation of the derivatives involves some computational overhead, the Gaussian quadrature took overall less time to converge. Finally, we should mention that the expressions for the derivatives will be also useful when we will be considering the uncertainty in δ in section 4.

2 The effect of the number of design points to parameter estimation

In this section we investigate the effect of the number of design points to the accuracy of estimating the model parameters. It is generally known that the likelihood function gets flatter as the number of design points decreases ([3], §5.4). In two dimensions, this phenomenon manifests itself with the replacement of the peak that should appear in the vicinity of the correct parameter values, with a ridge that runs across the axis of one of the parameters. This is illustrated in Figure 1. For $n = 20$, the maximisation algorithm returned $\tau_2 = 0.1$ and $\tau_2 = 0.08$, which is the location of the main peak presented in the left panel of Figure 1. For $n = 10$, we can notice the ridge that exists at $\tau_2 \sim 3$ and runs across the axis of τ_1 . In this case the maximisation algorithm returned $\tau_1 = -32$ and $\tau_2 = 3.4$, which correspond to $\delta_1 = 1.2 \times 10^7$ and $\delta_2 = 0.18$. The interpretation of this solution is that when there are very few data available, the model can explain them with only one input, while the second is essentially rendered inactive, by setting its correlation length to an unrealistically high value.

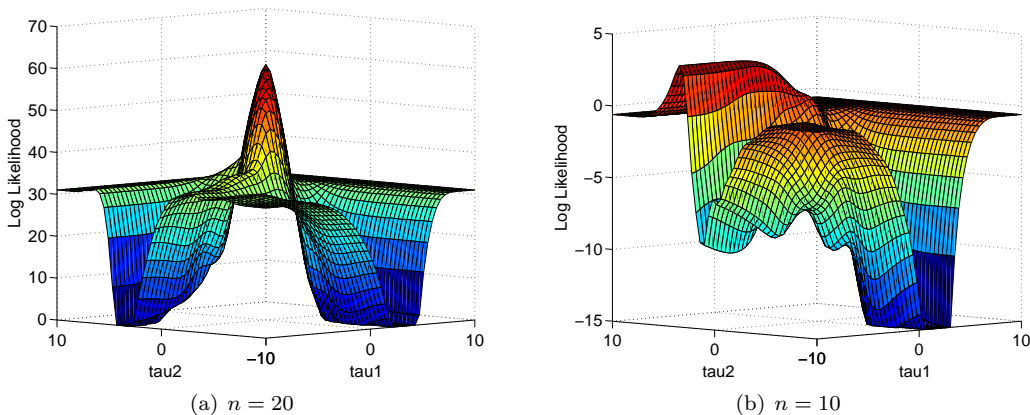


Figure 1: Log Likelihood (8) of data generated from a Gaussian process with $\tau = 0$. For the left panel $n = 20$ design points were used and for the right $n = 10$.

A question that then arises, is what is the minimum number of design points needed so as to avoid the above ‘degeneracy’. In our experiments the maximisation algorithm is initialised with the correlation lengths used for generating the data. We therefore make the assumption that if one of the estimated δ ’s is above a certain threshold, then the respective output is rendered inactive, as a result of not having used sufficient design points. To obtain the minimum number of points needed for the model to explain all the available inputs we drew samples from a Gaussian process for several combinations of inputs and design points. For each case, the experiment was run 1000 times, and we counted the number of realisations for which all the estimated δ ’s were smaller than a threshold, which was set to 5. The results for an input δ of 1 are shown in table 1.

The rows of this table correspond to the number of design points per input dimension (n/p) and the columns correspond to the number of input dimensions (p). The left part of the table shows the results for the REML method and the right for ML. Focusing on REML we see that for small dimensions using the rule of 10 point per input seems adequate for allowing all the runs to produce estimates within a reasonable range of δ . Using the same rule in 10 dimensions, almost 20% of the runs resulted in very large estimates for the correlation lengths.

Examining the right part of table 1, which corresponds to the ML method, similar conclusions can be drawn. An important observation however, is that for the same number of design points and inputs, fewer runs resulted in estimates of δ within the predetermined range. This implies that

more design points might be generally needed for estimating the correlation lengths when using the ML method. We will attempt to explain this in section 3.

Table 2 shows the same results when the correlation length used for generating the data was $\delta = 0.3$. A striking difference is that for the same number of inputs, significantly more design points are needed for estimating the model parameters when the correlation between the observations is smaller. This is particularly evident for high dimensional inputs. Note also the different n/p ratios used in tables 1 and 2.

| n/p | REML | | | | | ML | | | | |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ |
| 5 | 93 | 81 | 46 | 42 | 39 | 94 | 77 | 23 | 12 | 11 |
| 10 | 100 | 100 | 99 | 92 | 81 | 100 | 100 | 99 | 69 | 39 |
| 15 | 100 | 100 | 100 | 99 | 97 | 100 | 100 | 100 | 98 | 88 |
| 20 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99 |
| 25 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 30 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

Table 1: Percentage of realisations for which the estimated correlation lengths were less than 5. Input δ was 1.

| n/p | REML | | | | | ML | | | | |
|-------|------------|------------|------------|---------|----------|------------|------------|------------|---------|----------|
| | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ |
| 5 | 92 | 75 | 8 | 0 | 0 | 82 | 57 | 6 | 0 | 0 |
| 10 | 100 | 99 | 59 | 1 | 0 | 100 | 95 | 38 | 0 | 0 |
| 20 | 100 | 100 | 93 | 12 | 0 | 100 | 100 | 89 | 2 | 0 |
| 30 | 100 | 100 | 99 | 28 | 1 | 100 | 100 | 99 | 17 | 1 |
| 50 | 100 | 100 | 100 | 58 | 7 | 100 | 100 | 100 | 50 | 6 |

Table 2: Percentage of realisations for which the estimated correlation lengths were less than 5. Input δ was 0.3.

2.1 Evaluation using the Mahalanobis distance

In this section we investigate the effect of the number of design points on parameter estimation using the Mahalanobis distance. The Mahalanobis distance for both ML and REML methods follows a Chi squared distribution with n_p degrees of freedom, where n_p is the number of points used for prediction. In our experiments we used $n_p = n/2$. The above Chi Squared distribution has mean n_p and standard deviation $\sqrt{2n_p}$. If we denote by \bar{M} the average Mahalanobis distance from the successful runs out of 100, we define the normalised Mahalanobis distance as

$$M_n = \frac{\bar{M} - n_p}{2\sqrt{2n_p}} \quad (23)$$

The normalised version of the Mahalanobis distance has the benefit of being independent from the number of evaluation points used, and when its absolute value is less than one, then the mean

| n/p | REML | | | | | ML | | | | |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------|----------|
| | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ |
| 10 | 1.50 | 1.86 | 2.72 | 2.07 | 1.93 | 3.53 | 1.84 | 2.04 | 2.93 | 3.16 |
| 15 | 0.73 | 0.97 | 1.62 | 1.42 | 1.43 | 1.70 | 1.58 | 1.21 | 1.98 | 2.29 |
| 20 | 0.38 | 0.62 | 1.21 | 1.15 | 1.20 | 0.89 | 1.37 | 1.01 | 1.48 | 1.84 |
| 25 | 0.37 | 0.58 | 0.97 | 0.92 | 1.01 | 0.75 | 1.24 | 0.88 | 1.20 | 1.45 |
| 30 | 0.27 | 0.43 | 0.69 | 0.84 | 0.87 | 0.58 | 1.00 | 0.71 | 1.03 | 1.27 |

Table 3: Normalised Mahalanobis distance M_n for different number of inputs and design points.

Mahalanobis distance is within a standard deviation from the theoretical mean. The normalised Mahalanobis distances for the data drawn in the previous section using $\delta = 1$ are shown in table 3.

Examining the results for REML, we see that for two to three inputs, 15 design points per input seem sufficient for obtaining an average Mahalanobis distance that is closer to the theoretical mean than 1 standard deviation. For 5-10 inputs 25-30 points per input are needed for getting the same result. Also the increasing trend indicates that for more than 10 inputs, more points per input will be needed for keeping the average Mahalanobis distance close to its theoretical mean. The respective results for ML, show that the distances are larger in average compared to those of REML. This may be an indication that the predictions made using ML might be slightly more overconfident compared to REML¹.

2.2 Conclusion

As a conclusion we can say that the estimation of hyperparameters depends both on the dimensionality of the input but also on the inherent correlation length of the data. We saw that for $\delta = 1$, which represents a fairly smooth function, ten to 15 points per input suffice for 2-3 inputs, while for 5-10 inputs 25 to 30 points per input are needed. This number goes up dramatically as the output data vary faster. For example, we found that for 5 dimensions and $\delta = 0.3$, 50 points per input are needed (i.e. $n = 250$), and this number increases steeply as more dimensions are added.

3 Comparison between ML and REML

In this section we highlight the main differences between the ML and REML estimation methods. We also try to provide some insight by individually examining their terms.

Table 4 shows the average estimates of δ from the data used to make table 1. Table 5 shows the respective confidence intervals. The two tables show that ML slightly underestimates the correlation lengths and REML slightly overestimates them. However, the difference from the real value, which is $\delta = 1$, is so small that it would probably have no effect on the prediction.

The difference between the ML and REML estimates of δ seems to be consistent, as it is illustrated

¹We should note here that the primary use of the Mahalanobis distance is not to compare alternative models, but rather to indicate potential conflicts between the emulator and the simulator. However, because we are averaging the distances over many realisations, we believe that table 3 gives an indication of the comparative performance of ML and REML.

| n/p | REML | | | | | ML | | | | |
|-------|---------|---------|---------|---------|----------|---------|---------|---------|---------|----------|
| | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ |
| 10 | 1.03 | 1.05 | 1.09 | 1.07 | 1.07 | 0.93 | 0.92 | 0.89 | 0.99 | 1.04 |
| 15 | 1.02 | 1.02 | 1.03 | 1.03 | 1.03 | 0.99 | 0.96 | 0.94 | 0.98 | 1.00 |
| 20 | 1.02 | 1.01 | 1.02 | 1.02 | 1.02 | 1.00 | 0.97 | 0.96 | 0.98 | 1.00 |
| 25 | 1.02 | 1.01 | 1.01 | 1.01 | 1.01 | 1.00 | 0.99 | 0.97 | 0.98 | 1.00 |
| 30 | 1.02 | 1.01 | 1.00 | 1.01 | 1.01 | 1.01 | 0.99 | 0.98 | 0.98 | 0.99 |

Table 4: Average estimates of δ as a function of the design points and the dimensionality of the input.

| n/p | REML | | | | | ML | | | | |
|-------|---------|---------|---------|---------|----------|---------|---------|---------|---------|----------|
| | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ | $p = 2$ | $p = 3$ | $p = 5$ | $p = 8$ | $p = 10$ |
| 10 | 0.011 | 0.013 | 0.018 | 0.023 | 0.026 | 0.010 | 0.011 | 0.014 | 0.028 | 0.047 |
| 15 | 0.008 | 0.007 | 0.010 | 0.013 | 0.016 | 0.007 | 0.007 | 0.009 | 0.014 | 0.019 |
| 20 | 0.006 | 0.005 | 0.007 | 0.010 | 0.011 | 0.006 | 0.005 | 0.007 | 0.010 | 0.012 |
| 25 | 0.005 | 0.005 | 0.006 | 0.008 | 0.009 | 0.005 | 0.005 | 0.006 | 0.008 | 0.009 |
| 30 | 0.005 | 0.004 | 0.005 | 0.007 | 0.008 | 0.005 | 0.004 | 0.005 | 0.007 | 0.008 |

Table 5: Confidence intervals (95%) for table 4.

in figure 2. This figure shows the ML estimates plotted against those of REML, for $p = 3$. The REML estimates seem to always be higher than the ML, as it is indicated from the boundary that seems to exist along the line $\delta_{\text{ML}} = \delta_{\text{RL}}$.

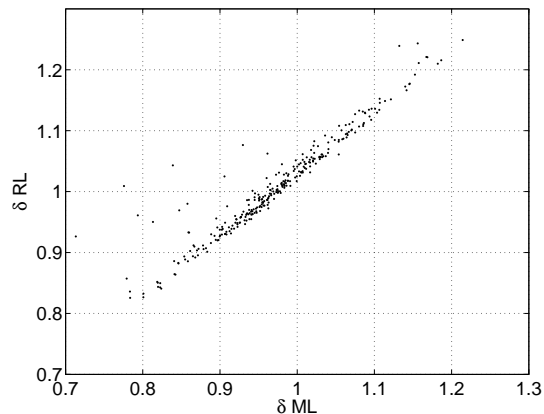


Figure 2: Scatter plot of $\hat{\delta}_{\text{ML}}$ against $\hat{\delta}_{\text{RL}}$, showing that $\hat{\delta}_{\text{RL}} > \hat{\delta}_{\text{ML}}$.

Section 2 highlighted two more differences between ML and REML: the first was that ML needed slightly more design points so as to explain the data using all the available inputs; the second was that the average Mahalanobis distance showed that ML might be slightly more overconfident than REML in its predictions. Finally, REML has the theoretical property of producing unbiased estimates of σ^2 , which was confirmed during our simulations. The following analysis will attempt to explain the above differences between REML and ML.

The expressions that need to be maximised for estimating δ using the ML and REML methods (6,

10) can be written as

$$L_{\text{ML}} \propto -\frac{1}{2} \ln |A| - \frac{n}{2} \ln S \quad (24)$$

$$L_{\text{RL}} \propto -\frac{1}{2} \ln |A| - \frac{n-q}{2} \ln S - \frac{1}{2} \ln |H^T A^{-1} H| \quad (25)$$

with $S \equiv (y - H\hat{\beta})^T A^{-1} (y - H\hat{\beta})$ being the generalised residual sum of squares. For convenience we denote the terms involved in the above expressions as

$$\begin{aligned} L_{\text{ML}}^1 &\equiv -\frac{1}{2} \ln |A| & L_{\text{RL}}^1 &\equiv -\frac{1}{2} \ln |A| \\ L_{\text{ML}}^2 &\equiv -\frac{n}{2} \ln S & L_{\text{RL}}^2 &\equiv -\frac{n-q}{2} \ln S \\ & & L_{\text{RL}}^3 &\equiv -\frac{1}{2} \ln |H^T A^{-1} H| \end{aligned}$$

The terms L_{ML}^1 and L_{RL}^1 are identical and L_{ML}^2 and L_{RL}^2 differ only in their multiplying factors. To illustrate the function of the above terms within the expressions of the respective log likelihoods we use 5 data points drawn from a Gaussian Process in $[0, 1]$ with $\delta = 1$ and plot the above terms as a function of $\ln \delta$. The term L_{ML}^1 (fig. 3 left panel) is known as the model complexity penalty ([3], §5.4). This term favours larger correlation lengths because the model gets more rigid and therefore less complex. The right panel of figure 3 shows the data fit terms L^2 , which are the only ones that depend on the data. The L^2 terms support smaller correlation lengths because the model then gets more flexible and fits the data better. The compromise between these two terms gives the ML solution.

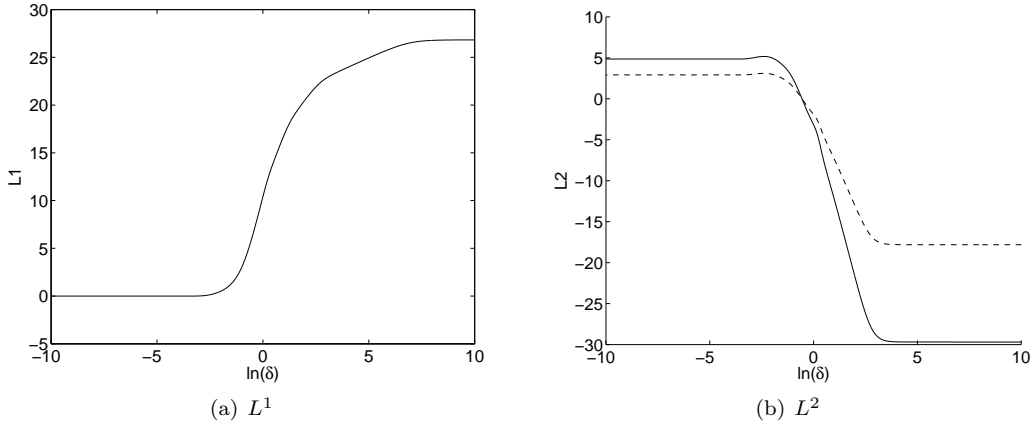


Figure 3: The left panel shows the model complexity penalty L^1 . The right panel shows the data fit terms L_{ML}^2 (cont. line), L_{RL}^2 (dashed line).

Figure 3 also provides a clue as to why the REML estimates of δ tend to be higher than those of ML. L_{RL}^2 is bounded by L_{ML}^2 and as a result has a less steep slope. This is likely to be the reason why the correlation lengths estimated with REML are generally higher than those estimated with ML.

The left panel of figure 4 shows the L_{RL}^3 term. The right panel of the same figure shows L_{ML} (continuous line), L_{RL} (dashed line) and the difference $L_{\text{RL}} - L_{\text{RL}}^3$ (dashed red line). This figure indicates that the L_{RL}^3 term helps equalising the two ‘wings’ (plateaus towards $\pm\infty$) of L_{RL} . Perhaps

most importantly it helps eradicating the positive slope that appears in L_{ML} for $\ln \delta \in [3, 8]$. We can envisage a situation in which if the maximisation algorithm landed in the above interval, it would diverge towards $+\infty$ for ML, while it would converge to the maximum of L_{RL} . This could be a reason why REML explains the data using all the inputs more often than ML does.

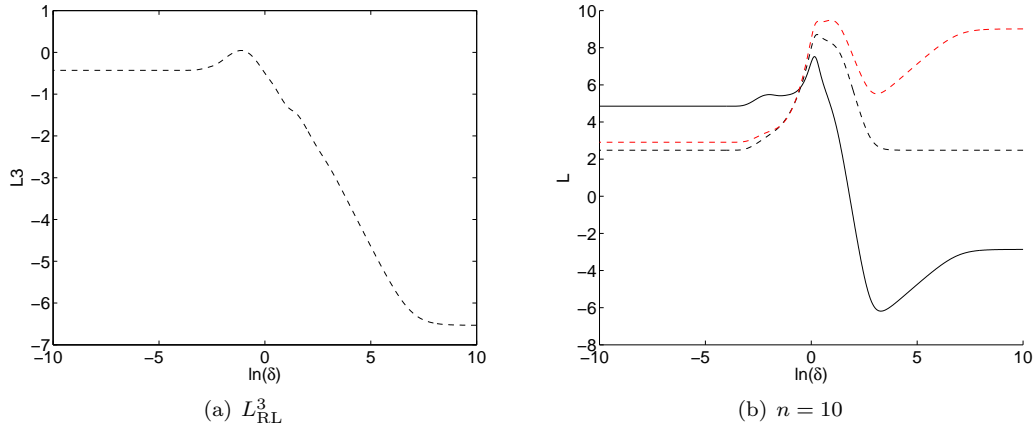


Figure 4: L^3_{RL} is shown in the left panel. The right panel shows the L_{ML} and L_{RL} terms (continuous and dashed black lines) and the difference $L_{RL} - L^3_{RL}$ (red dashed line).

Finally, the lower Mahalanobis distances for REML, which indicate that is less overconfident than ML, might be due to the fact that REML accounts for the uncertainty in β and as a result, its predictions have greater variance. For the same reason we would expect the Mahalanobis distance for the Toolkit method to be even lower.

4 Accounting for the uncertainty in δ

In section 1 we saw that the REML and Toolkit methods for parameter estimation are derived from Maximum Likelihood by accounting for the uncertainty in β and σ^2 respectively. An obvious extension would be to account for our uncertainty about the value of δ . However, this is not as straightforward as it is with the two other parameters, because the integral $\int p(y|\delta)p(\delta) d\delta$ can not be calculated analytically. We therefore have to resort to numerical or approximation methods. Two such methods are considered here: integration via MCMC and the normal approximation of the posterior distribution.

Before using MCMC integration, there is one technicality that we need to address. The likelihood functions 3, 8 and 13 do not decay to zero as τ tends to $\pm\infty$; as a result, the posterior calculated using uniform priors is improper. This result is also proved in [4]. MCMC integration can fail to converge when improper posteriors are used. This problem can be alleviated using priors that make the posterior distribution proper, but at the same time will not alter drastically the shape of the likelihood, at least in the parameter range we are interested in. Two such priors are examined in the next section.

4.1 Correlation length priors

4.1.1 Reference prior

The reference priors were proposed by Berger et al. [4] for 1 dimension, and were extended to higher dimensions by Paulo [5]. They can be considered as Jeffrey’s priors on the marginalised likelihood $p(y|\sigma^2, \delta)$ and their derivation is based on the maximisation of the Kullback Leibler divergence between the prior and the posterior distribution. The reference prior is given by

$$p(\tau) \propto |\mathcal{I}(\tau)|^{1/2} \quad (26)$$

where

$$\mathcal{I} = \begin{bmatrix} n-q & \text{tr}W_1 & \text{tr}W_1 & \cdots & \text{tr}W_p \\ & \text{tr}W_1^2 & \text{tr}W_1W_2 & \cdots & \text{tr}W_1W_p \\ & & \ddots & \cdots & \vdots \\ & & & & \text{tr}W_p^2 \end{bmatrix}$$

and

$$W_k = \frac{\partial A}{\partial \tau_k} A^{-1} (I - H(H'A^{-1}H)^{-1}H'A^{-1})$$

Figure 5 shows the logarithms of the likelihood, the reference prior and the resulting posterior distribution, for 10 points drawn from a 1 dimensional Gaussian process in $[0, 1]$ with $\tau = 0$. The reference prior is relatively flat in the region of interest, while it drops off as τ takes very large or very small values, hence making the posterior distribution proper.

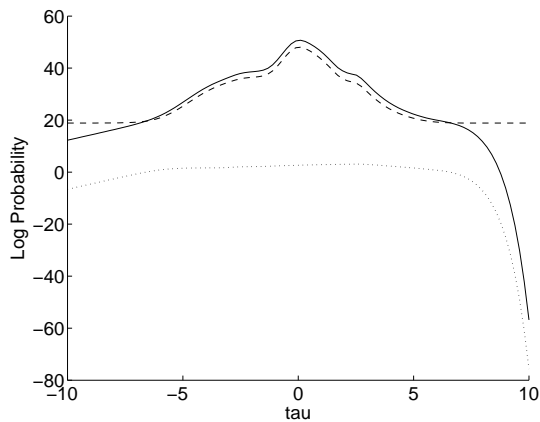


Figure 5: Log of the likelihood (dashed) reference prior (dotted) and posterior (continuous) distributions corresponding to 10 points drawn from a 1 dimensional GP in $[0, 1]$ with $\tau = 0$.

To check the effect of the reference prior on the estimates of δ we compared the REML estimates ($\hat{\delta}_{\text{RL}}$) with the MAP estimates of the posterior formulated by the REML likelihood and the reference prior ($\hat{\delta}_{\text{RF}}$). One hundred realisations of data drawn from a GP in $[0, 1]$ were made for three different cases: a) $p = 1, n = 10, \delta = 1$, b) $p = 1, n = 10, \delta = 0.3$ and c) $p = 3, n = 30, \delta = 1$. Figure 6 shows the histograms of the differences ($\delta_{\text{RL}} - \delta_{\text{RF}}$). The majority of the errors is less than 0.02, which is expected to have a rather negligible effect in prediction.

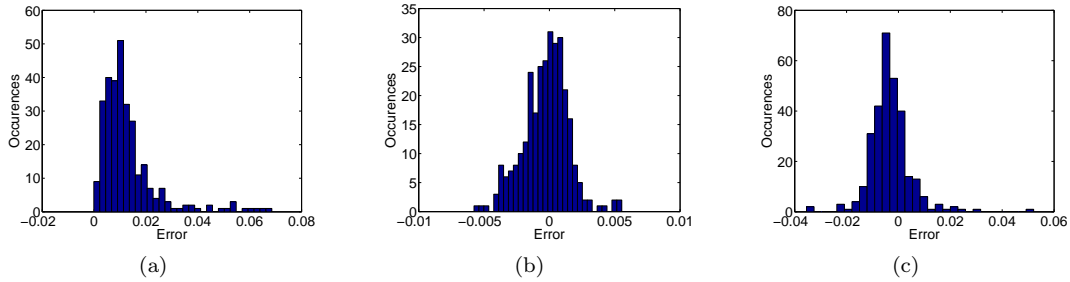


Figure 6: Histograms of the difference between the REML estimate $\hat{\delta}_{\text{RL}}$ and the estimate obtained with the reference prior $\hat{\delta}_{\text{RF}}$ for three cases a) $p = 1$, $n = 10$, $\delta = 1$, b) $p = 1$, $n = 10$, $\delta = 0.3$ and c) $p = 3$, $n = 30$, $\delta = 1$.

4.1.2 Exponential Inverse Gamma prior

Apart from the reference prior we introduce here another prior that is derived from an exponential transformation $g(x) = e^{-x}$ of the inverse Gamma distribution. That is, if a random variable x follows an Exponential Inverse Gamma (EIG) distribution, then e^{-x} will follow the inverse Gamma. The formula for the EIG distribution² is

$$p(\tau) = e^{-\alpha\tau} \exp(-\gamma e^{-\tau}) \quad (27)$$

The logarithm of the EIG distribution is shown in figure 7. The hyperparameter γ controls the slope for $\tau \rightarrow -\infty$ and α controls the slope for $\tau \rightarrow \infty$. The mode of this distribution is found at $\tau = -\ln(\alpha/\gamma)$.

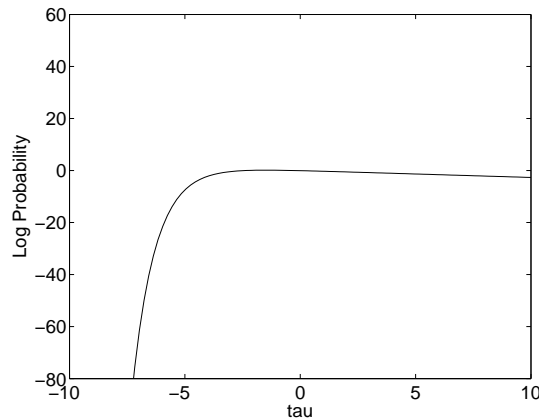


Figure 7: Logarithm of the EIG prior for $\alpha = 0.2667$ and $\gamma = 0.0595$.

This is clearly an informative prior, whose shape does not depend on the data as the shape of the reference prior does. As we are mainly interested in making the posterior distribution proper, we only wish to provide vague information on the value of δ . We start from the assumption that correlation lengths outside the range $\delta \in [0.05, 2]$ are very unlikely to occur. In the space of τ this translates to $\tau \in [-1.5, 6]$. We place the mode of our prior at $\tau = -1.5$. This implies the condition $\tau_0 = -\ln(\alpha/\gamma) = -1.5$. At $\tau_1 = 6$ we want the logarithm of our prior to have dropped by approximately 2, which represents a rough 95% interval. As the rate of this drop is determined

²This distribution is closely related to the Extreme Value Distribution

by the $e^{-\alpha\tau}$ term, we require that $e^{-\alpha(\tau_0-\tau_1)} = e^2$. The above two conditions yield $\alpha = 0.2667$ and $\gamma = 0.0595$. These values of the hyperparameters were used for the prior shown in figure 7. Figure 8 shows the likelihood also shown in figure 5, the EIG prior and the resulting posterior distribution.

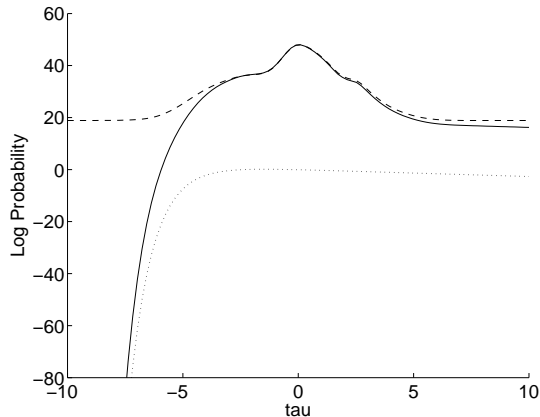


Figure 8: Log of the likelihood (dashed) EIG prior (dotted) and posterior (continuous) distributions corresponding to 10 points drawn from a 1 dimensional GP in $[0, 1]$ with $\tau = 0$.

To evaluate the effect of this prior on the estimates of δ we performed the same experiment as with the reference prior, and the results are shown in figure 9. The errors are comparable to those of the reference prior, although there is a small negative bias, which implies that that δ 's are slightly overestimated when the EIG prior is used. Nevertheless, the errors are rather small so we should not expect them to have a notable impact on prediction. On the other hand, the EIG priors are a lot cheaper computationally, which can be an important advantage in MCMC simulations.

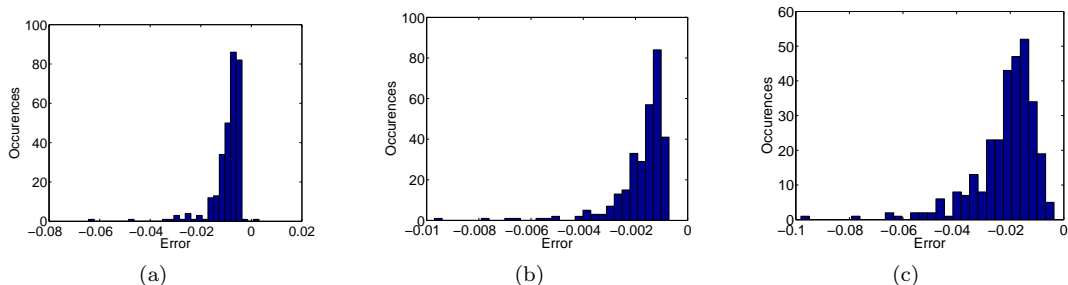


Figure 9: Histograms of the difference between the REML estimate $\hat{\delta}_{\text{RL}}$ and the estimate obtained with the EIG prior $\hat{\delta}_{\text{EG}}$ for three cases a) $p = 1$, $n = 10$, $\delta = 1$, b) $p = 1$, $n = 10$, $\delta = 0.3$ and c) $p = 3$, $n = 30$, $\delta = 1$.

Another advantage of using either of the above priors is that they might make the optimisation procedure for finding the ML estimates of δ more robust. The reason is that because the priors are essentially applying weight in the more interesting ranges of τ , the optimisation algorithm is less likely to converge to unrealistically large or small values for the correlation lengths.

4.2 Methods for accounting for the uncertainty in δ

As we mentioned at the beginning of this section, accounting for the uncertainty in δ is not possible via an analytic integration of $p(\eta(\mathbf{x})|\delta, y)$ w.r.t. δ . We now present two methods of sidestepping this problem.

4.2.1 MCMC integration

The first method involves the construction of an MCMC algorithm that can sample from the posterior $p(\tau|y)$. The resulting samples of τ can then be used for prediction. An outline of how this can be achieved is given below.

We first estimate the value of τ that maximises $p(\tau|y)$ and call it $\hat{\tau}$. We then find the Hessian matrix at $\hat{\tau}$, using the expressions for the derivatives from equations (20 - 22). If we call this matrix $H_{\hat{\tau}}$ we can then define a matrix $V_{\hat{\tau}} = -H_{\hat{\tau}}^{-1}$. We then setup the following MCMC (Metropolis) algorithm

1. Set $\tau^{(1)}$ equal to $\hat{\tau}$
2. Add to $\tau^{(i)}$ a normal variate drawn from $\mathcal{N}(0, V_{\hat{\tau}})$, call the result τ^*
3. Calculate the ratio $\alpha = \frac{p(y|\tau^*)}{p(y|\tau^{(i)})}$
4. Set $\tau^{(i+1)} = \tau^*$ with probability α and $\tau^{(i+1)} = \tau^{(i)}$ with probability $1 - \alpha$
5. Repeat steps 2-4 until a sufficient number of samples has been drawn.

Once a sufficient number of samples M has been drawn, we can do inference about the output of the emulator. For example, an estimate of the mean of $\eta(x)$ can be

$$\hat{\eta}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M \int \eta(\mathbf{x}) p(\eta(\mathbf{x})|\tau^{(i)}, y) d\eta$$

The main drawback of the MCMC based scheme, is that it is computationally demanding. It is reported however to give good coverage probability [2, 5].

4.2.2 Normal approximation of the posterior

An alternative method of obtaining samples of δ involves the approximation of the posterior $p(\tau|y)$ using a normal distribution, as described in [2]. As with MCMC, the mode $\hat{\tau}$ of the posterior $p(\tau|y)$ is first found. Then the inverse Hessian matrix $V_{\hat{\tau}} = -H_{\hat{\tau}}^{-1}$ is calculated. Finally, the posterior $p(\tau|y)$ is approximated with the normal distribution $\mathcal{N}(\hat{\tau}, V_{\hat{\tau}})$, which is used for drawing samples of τ .

A benefit of this approach is that the samples are i.i.d., and is also significantly faster than the MCMC based method. A potential problem however, can be that the samples are not drawn from the actual posterior, but from an approximation that may or may not be accurate. For example, if the optimisation procedure used for finding the mode $\hat{\tau}$ converges to a local minimum, this mode will then be approximated but not the main one. Nevertheless, Nagy et al. [2] report coverage probabilities similar to those obtained with MCMC.

5 Conclusion

In this report we saw how three parameter estimation methods, ML REML and the Toolkit approach, are connected, via marginalisation of their likelihood functions. We also showed that the Toolkit approach is closely related to REML and that for $n - p \gg$ they become identical. We also derived expressions for the derivatives for the REML/Toolkit cost functions, which speed up significantly the optimisation process and are helpful when accounting for our uncertainty in δ .

We then investigated the effect of the number of design points to the estimation of the correlation lengths. We saw that the accuracy of estimation strongly depends on the dimensionality of the input as well as on the inherent correlation length of the data. A main finding was that for moderately smooth data 15 points per dimension were needed for 2-3 dimensional inputs, while for 5-10 dimensions the estimates were satisfactory for 25-30 design points per input. It is also likely that more points per input will be needed for dimensions higher than 10.

A comparison between the ML and REML methods, showed that the latter needed somewhat fewer design points for estimating the correlation lengths, while ML was slightly more overconfident in its prediction. This might as well be due to the fact that ML does not account for the uncertainty in β . Finally, we found that ML slightly underestimated and REML slightly overestimated the correlation lengths, but not by an amount that should have a notable effect on the predictions.

In the final part of the report we considered two methods that account for the uncertainty in δ , one based on MCMC integration and one on the approximation of the posterior with a normal distribution. We focused on the application of priors that result in proper posterior distributions, as required by MCMC. We implemented the reference prior, and proposed a much simpler one, based on a transformation of the inverse gamma distribution. None of the priors seemed to alter the shape of the likelihood, in the parameter range of interest, but both priors resulted in proper posterior distributions. The evaluation of the normal approximation and the MCMC methods using the proposed priors consists work in progress.

References

- [1] D. Harville, "Bayesian inference for variance components using only error contrasts," *Biometrika*, vol. 61, pp. 383–385, 1974.
- [2] B. Nagy, J. Loepky, and W. Welch, "Correlation parameterization in random function models to improve normal approximation of the likelihood or posterior," Dept. of Statistics, The University of British Columbia, URL <http://stat.ubc.ca/Research/TechReports/techreports/229.pdf>, Tech. Rep. 229, 2007.
- [3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.
- [4] J. O. Berger, V. D. Oliveira, and B. Sansó, "Objective Bayesian analysis of spatially correlated data," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1361–1374, 2001.
- [5] R. Paulo, "Default priors for Gaussian processes," *Annals of Statistics*, vol. 33, pp. 556–582, 2005.