A Summary of Gaussian Processes

Coryn A.L. Bailer-Jones
Cavendish Laboratory
University of Cambridge
calj@mrao.cam.ac.uk

1997

Introduction

A general prediction problem can be posed as follows. We consider that the variable of interest, t, is related to the set of measurable variables, $\mathbf{x}$, via some function $\mathcal{F}$, such that

$$t = \mathcal{F}(\mathbf{x}) \quad . \tag{1}$$

Typically, the function $\mathcal{F}$ will be unknown. Cast probabilistically, the problem becomes one of evaluating $P(t_{N+1}|\mathcal{D}, \mathbf{x}_{N+1})$ given $\mathbf{x}_{N+1}$ and a set of *training* data, $\mathcal{D} = \{\{\mathbf{x}_N\}, \mathbf{t}_N\}$.

I shall use the following notation: t_{N+1} is the single data 'output' corresponding to the L inputs denoted by the vector $\mathbf{x}_{N+1}$ (i.e. the dimensionality of the input space is L). $\mathbf{t}_{N+1}$ is the vector of the $N+1$ values of t, for which the corresponding set of input vectors is the set of vectors $\{\mathbf{x}_{N+1}\}$, which can be considered as an $(N+1) \times L$ matrix.

The Gaussian Process Model

A graphical summary of how the Gaussian Process model performs predictions is given in Figure 1.

The Gaussian Process model is an attempt to solve this problem by assuming that the set of variables $\mathbf{t}_N$ has a joint Gaussian distribution,

$$P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{C}_N, \boldsymbol{\mu}) = \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{t}_N - \boldsymbol{\mu})^T \mathbf{C}_N^{-1} (\mathbf{t}_N - \boldsymbol{\mu})\right) \quad . \tag{2}$$

Note that the distribution is completely determined by $\boldsymbol{\mu}$ and $\mathbf{C}_N$. The covariance matrix, $\mathbf{C}_N$, has elements $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$. I will consider $\boldsymbol{\mu} = 0$. The use of $\boldsymbol{\mu} \neq 0$ will be discussed later. Thus

$$P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{C}_N) = \frac{1}{Z_N} \exp\left(-\frac{1}{2}\mathbf{t}_N^T \mathbf{C}_N^{-1} \mathbf{t}_N\right) \quad . \tag{3}$$

Let t_{N+1} be the value we wish to predict given its corresponding 'input' vector $\mathbf{x}_{N+1}$. The joint distribution of $\mathbf{t}_{N+1}$ is

$$P(\mathbf{t}_{N+1}|\{\mathbf{x}_N\}, x_{N+1}, \mathbf{C}_{N+1}) = \frac{1}{Z_{N+1}} \exp\left(-\frac{1}{2}\mathbf{t}_{N+1}^T \mathbf{C}_{N+1}^{-1} \mathbf{t}_{N+1}\right) \quad . \tag{4}$$

The predictive probability distribution for t_{N+1} is, therefore,

$$P(t_{N+1}|\mathbf{t}_N, \{\mathbf{x}_N\}, \mathbf{x}_{N+1}, \mathbf{C}_{N+1}) = \frac{P(\mathbf{t}_{N+1}|\{\mathbf{x}_N\}, \mathbf{x}_{N+1}, \mathbf{C}_{N+1})}{P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{C}_N)} \tag{5}$$

$$= \frac{Z_N}{Z_{N+1}} \exp\left[-\frac{1}{2}(\mathbf{t}_{N+1}^T \mathbf{C}_{N+1}^{-1} \mathbf{t}_{N+1} - \mathbf{t}_N^T \mathbf{C}_N^{-1} \mathbf{t}_N)\right] \quad . \tag{6}$$

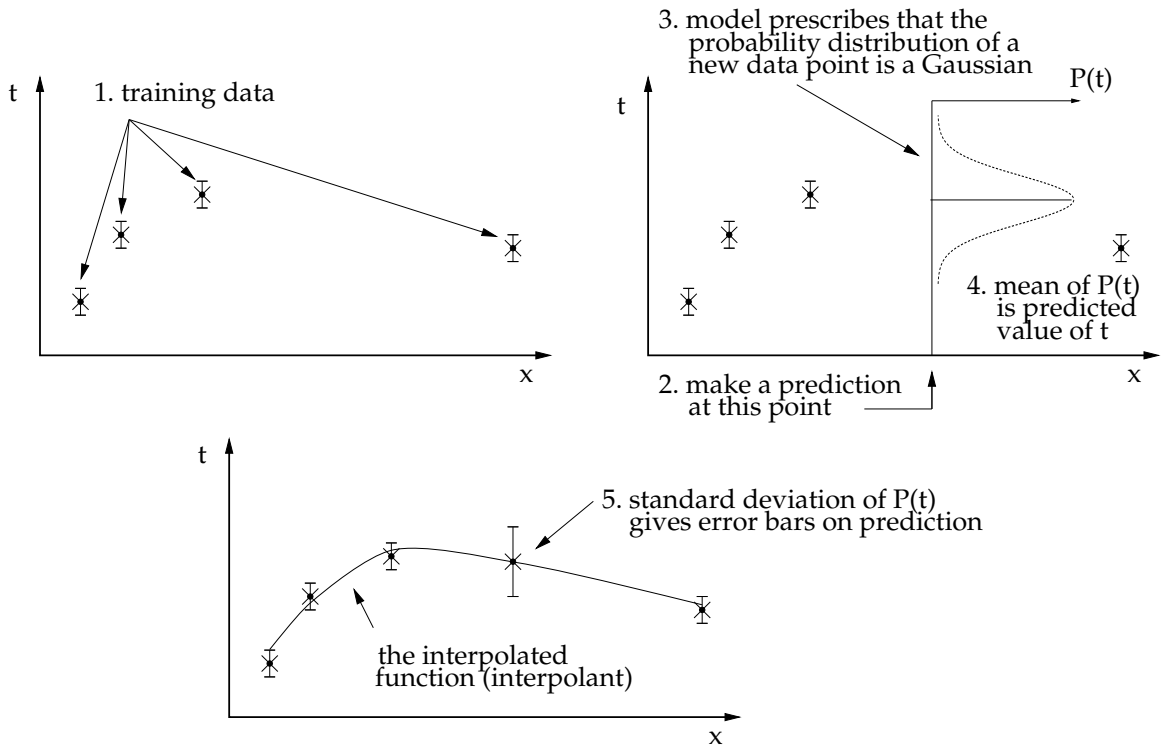

Figure 1: Summary of how predictions are made with a Gaussian Process model.

After some matrix manipulation it can be shown that

$$P(t_{N+1}|\mathbf{t}_N, \{\mathbf{x}_N\}, \mathbf{x}_{N+1}, \mathbf{C}_{N+1}) = \frac{1}{Z} \exp\left(-\frac{(t_{N+1} - \hat{t}_{N+1})^2}{2\sigma^2_{\hat{t}_{N+1}}}\right) \quad , \tag{7}$$

where

$$\hat{t}_{N+1} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \quad , \tag{8}$$

$$\sigma^2_{\hat{t}_{N+1}} = k - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \quad , \tag{9}$$

and

$$\mathbf{k} = [C(\mathbf{x}_1, \mathbf{x}_{N+1}), C(\mathbf{x}_2, \mathbf{x}_{N+1}), \ldots, C(\mathbf{x}_N, \mathbf{x}_{N+1})] \quad , \tag{10}$$

$$k = C(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) \tag{11}$$

As $\hat{t}_{N+1}$ is the maximum value of the probability distribution of interest[1] it is the predicted value for t_{N+1}. Thus $\hat{t}(x)$ is the interpolant of the data, i.e. the predicted function. Note that it does not depend on $\mathbf{C}_{N+1}$. Therefore we only have to invert a single covariance matrix once, namely $\mathbf{C}_N$, in order to make any number of predictions, t_{N+1}. A diagramatic representation of the relationship between these vectors and matrices is given in Figure 2.

[1]Note that even if $P(\mathbf{t}_N)$ and $P(\mathbf{t}_{N+1})$ are zero-mean, the conditional distribution $P(t_{N+1})$ is not necessarily zero-mean.

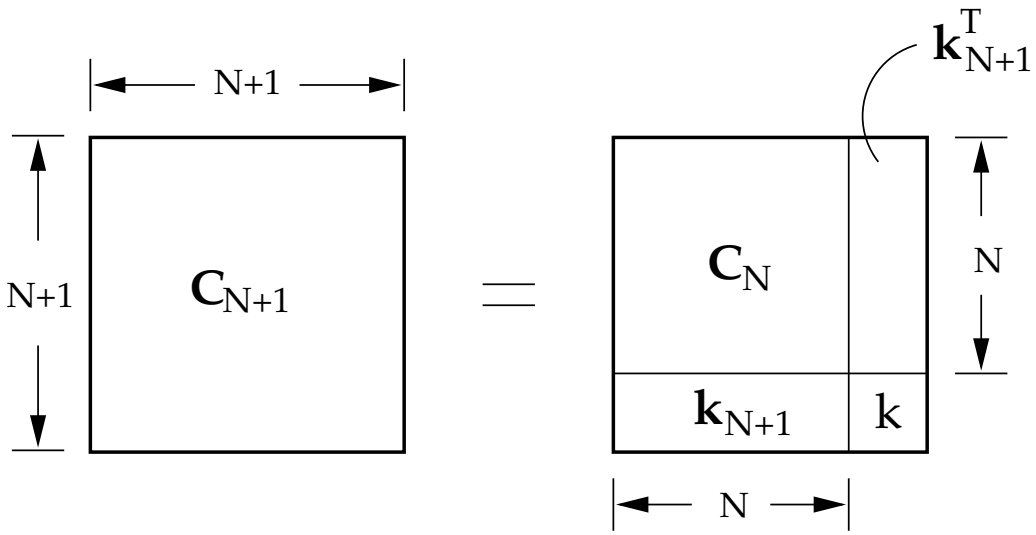

Figure 2: The relationship between the matrices $\mathbf{C}_N$ and $\mathbf{C}_{N+1}$.

Covariance Function

The elements of the covariance matrix, $\mathbf{C}_N$, are denoted C_{ij}. By definition, the covariance between t_i and t_j is defined as $C(t_i, t_j) = \mathcal{E}[(t_i - \mathcal{E}[t_i])(t_j - \mathcal{E}[t_j])]$, where $\mathcal{E}$ denotes expectation. But to evaluate expectation values we need to know the probability distribution over t, and it is exactly that which we are trying to find. Therefore we must parameterize the covariance function, C_{ij}, and infer these parameters from the training data. C_{ij} is a function of the training input data, $\{\mathbf{x}_N\}$, because these data determine the correlation between the training data outputs, $\mathbf{t}_N$. Thus instead explicity parameterizing the function, $\mathcal{F}$, and solving for its parameters by some form of regression, the Gaussian Process approach defines a parameterized probabilistic model for the correlation between different values of the function. These parameters are then found using standard optimisation techniques.

A suitable form of the covariance function is

$$C_{ij} = \theta_1 \exp\left[-\frac{1}{2}\sum_{l=1}^{l=L}\frac{(x_i^{(l)} - x_j^{(l)})^2}{r_l^2}\right] + \theta_2 + \theta_3\delta_{ij} + L_{ij} \quad , \tag{12}$$

where $x_i^{(l)}$ is the lth dimension of the ith input vector, $\mathbf{x}_i$. The four terms in this equation are now discussed.

1. The exponential term specifies that we wish to fit a smooth interpolant to the training data. The form of this term expresses our belief that inputs which are close to each other give rise to outputs which are close to each other: It achieves this by yielding a relatively large contribution to C_{ij} when $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar. Each input dimension is given a separate 'length scale', r_l, which dictates how rapidly the interpolant varies as a function of input x_l. If r_l is relatively large, the exponential term will be small, the contribution to C_{ij} will be large, and hence the function will not vary much as x_l is varied. r_l is a measure of the length over which x_l varies significantly. We can therefore also consider r_l^{-1} as a measure of the relevance of the lth input in determining the output. If the function hardly varies at all with one of the inputs (r_l large) we would say that this input has little relevance in determining the output: we could probably leave out this input and the model could compensate by an adjustment in the value of θ_2. Note

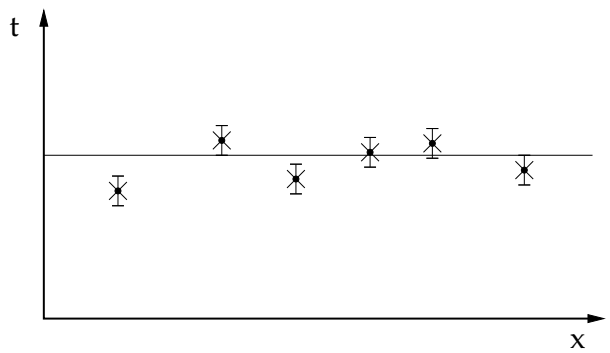

Figure 3: The constant term in C_{ij} (θ_2) contributes a constant to the interpolating function.

that *significance* is strictly defined as $\partial t/\partial x_l$, and is a function of $\mathbf{x}$. The parameter θ_1 gives the overall scale of variations in the t space.

2. The constant term, θ_2, provides for data, $\mathbf{t}_N$, with a non-zero mean. Consider a two-dimensional case. Let

$$\mathbf{C}_N = \begin{pmatrix} \theta_2 & \rho\theta_2 \\ \rho\theta_2 & \theta_2 \end{pmatrix} = \theta_2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad .$$

Therefore,

$$\mathbf{C}_N^{-1} = \frac{1}{\theta_2}\frac{1}{1-\rho^2}\begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \quad .$$

As $\rho \to 1$, $\mathbf{C}_N^{-1} \to \infty$, i.e. there is perfect correlation between t_i and t_j. So, if θ_2 is the only term in C_{ij}, then $C_{ij} = const$, which means $t_i = t_j$ or more specifically $t_{N+1} = const$. In other words, the interpolant would be a hyperplane of gradient zero (horizontal line in the one-dimensional case, see Figure 3). In general, then, the θ_2 term adds a *constant* offset to the interpolant, which is a similar role to the bias node in neural networks.

Another way of thinking about this is to consider the prediction equation,

$$\hat{t}_{N+1} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \quad . \tag{13}$$

Both $\mathbf{C}_N^{-1}$ and $\mathbf{t}_N$ depend only upon the training data, so are constant for any predictions and when we make predictions far from the data, $\mathbf{k}^T = \theta_2(1, 1, \ldots, 1)$, and hence $\hat{t}_{N+1} = \theta_2 \times const$.

3. This is the noise model for the outputs and therefore only occurs in C_{ij} when $i = j$. In this case the noise is assumed to be input independent Gaussian noise with variance θ_3.

4. Without the L_{ij} term in equation 12, $C_{ij} \to \theta_2$ for $|\mathbf{x}_i| \gg |\mathbf{x}_j|$, which would result in t_i and t_j assuming the same value. The L_{ij} allows us to model linear trends in the data. The equation of a plane in 'real' (rather than covariance) space can be written

$$t_i = \sum_{l=1}^{L} a_l x_i^{(l)} \quad .$$

The covariance of any two such functions, t_i and t_j is

$$\begin{aligned}
\mathrm{Cov}[t_i, t_j] &= \mathcal{E}[(t_i - \mathcal{E}(t_i))(t_j - \mathcal{E}(t_j))] && (14)\\
&= \mathcal{E}[t_i t_j] - \overline{t_i t_j} && (15)\\
&= \mathcal{E}\left[\sum_l a_l^2 x_i^{(l)} x_j^{(l)} + \sum_{m \neq n} a_m a_n x_i^{(m)} x_j^{(n)}\right] - \left(\sum_l \overline{a_l} x_i^{(l)}\right)\left(\sum_l \overline{a_l} x_j^{(l)}\right) && (16)
\end{aligned}$$

where $\mathcal{E}$ is the expectation operator. In order to evaluate these expectations we need to know what the prior probability distributions over t_i and t_j are, which corresponds to needing to know the priors for the parameters a_l. If the a_l are independent then we have

$$\mathrm{Cov}[t_i, t_j] = \mathcal{E}\left[\sum_l a_l^2 x_i^{(l)} x_j^{(l)}\right] - \left(\sum_l \overline{a_l} x_i^{(l)}\right)\left(\sum_l \overline{a_l} x_j^{(l)}\right) \quad .$$

If we put Gaussian priors on the a_l with zero mean and variances σ_l, then we get

$$\mathrm{Cov}[t_i, t_j] = \sum_l \sigma_l^2 x_i^{(l)} x_j^{(l)} \quad . \tag{17}$$

Alternatively, we may want to put a delta function prior on the a_l, i.e. $P(a_l) = \delta(a_l - \alpha_l)$, in which case we get

$$\mathrm{Cov}[t_i, t_j] = \rceil\left[\sum_l \alpha_l^2 x_i^{(l)} x_j^{(l)}\right] - \left(\sum_l \alpha_l x_i^{(l)}\right)\left(\sum_l \alpha_l x_j^{(l)}\right) \quad . \tag{18}$$

Thus the expression in either equation 17 or equation 18 can be used as the L_{ij} term in equation 12. When optimizing the model we can then put any priors on the α_l (or alternatively on the σ_l) that we want.

We can write the linear term in equation 17 as $\mathbf{z}.\mathbf{b}$ where $z^{(l)} = x_i^{(l)} x_j^{(l)}$ and $\mathbf{a} = (\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2)$. $C = \mathbf{z}.\mathbf{b}$ is just the equation of a hyperplane with normal $\mathbf{b}$. (This hyperplane is in $C = C(z^{(1)}, z^{(2)}, \ldots, z^{(L)})$ space.) Inclusion of this term means that we can model linear trends in the function $t = \mathcal{F}(\mathbf{x})$. If $\mathbf{a}$ were negative, then $x_i \gg x_j$ implies that C_{ij} would be large and negative, meaning that t_i and t_j would be very different. If we did not have this term then $t \to const$ for values of t which deviate greatly from the range in the training data, as was discussed in point number 1 in this list.

There are other forms of the covariance function which could be used, such as a more complex (input dependent) noise model. The only restriction is that the covariance matrix be positive definite. From comparison with neural network methods, the parameters of a Gaussian Process are often referred to as *hyperparameters* rather than parameters, and this nomenclature will now be adopted. The reason for this distinction is that the hyperparameters of a Gaussian Process do not parameterize the function in the way that the neural network weights do.

I have considered $\boldsymbol{\mu} = 0$. It looks as though equation 2 would then give the most probable value for $\mathbf{t}_N$ as $\mathbf{t}_N = \mathbf{0}$. However, this assumes that $\mathbf{C}_N$ is not singular. If we had just the θ_2 (constant) term in C_{ij}, then $\mathbf{C}_N$ would be singular and we find that $\mathbf{t}_N = a(1, 1, \ldots, 1)$, where a is a constant. Thus a zero-mean Gaussian Process is completely general provided we have a consant term in the covariance function. If we use a Gaussian Process with a non-zero mean, $\boldsymbol{\mu} = \theta_0(1, 1, \ldots, 1)$, then θ_0 is

another hyperparameter which must be inferred from the data. We would expect its value to be near to the mean of the training data. When I have used a non-zero mean hyperparameter, the particular implementation of a Gaussian Process I used set θ_0 to some sensible value and set θ_2 to zero. In theory there is no reason to use a model with both θ_0 and θ_2, athough there may be numerical reasons.

Hyperparameter Determination

The various hyperparameters of the covariance function are of course not known in advance, and they must be determined using the training data. From the Bayesian point of view we would like to integrate over all possible hyperparameters. These integrations can be achieved in principle using Monte Carlo methods. Alternatively we can take the maximum likelihood approach.

Let $\mathbf{\Theta}$ be the set of all hyperparameters, $\mathbf{\Theta} = \{\theta_2, \theta_1, \theta_3, a_1, \ldots, a_L, r_1, \ldots, r_L\}$. The likelihood of the parameters, $L(\mathbf{\Theta})$, is

$$L(\mathbf{\Theta}) \equiv P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{C}_N) \tag{19}$$

$$\equiv P(\mathbf{t}_N|\{\mathbf{x}_N\}, \Theta, C_f) \quad , \tag{20}$$

where C_f specifies the form of the covariance function. The maximum likelihood approach is to maximize $L(\mathbf{\Theta})$ to yield the optimum hyperparameters. An improvement over this is to incorporate a prior, $P(\Theta)$, on the hyperparameters. By Bayes' theorem, the posterior probability of the hyperparameters given the training data is

$$P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f) = \frac{P(\mathbf{t}_N|\{\mathbf{x}_N\}, \Theta, C_f)P(\Theta|\{\mathbf{x}_N\}, C_f)}{P(\mathbf{t}_N|\{\mathbf{x}_N\}, C_f)} \quad . \tag{21}$$

Maximisation of $P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f)$ is known as the *maximum a posteriori* (MAP) approach, which is a Bayesian version of maximum likelihood estimation. This will be a good approximation to the 'full' Bayesian approach (i.e. integrating over all hyperparameters) if the probability mass of the probability distribution for the hyperparameters is strongly concentrated around the maximum likelihood solution:

$$P(\mathbf{t}_N|\{\mathbf{x}_N\}, C_f) = \int P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{\Theta}, C_f)P(\mathbf{\Theta}|\{\mathbf{x}_N\}, C_f)d\mathbf{\Theta} \tag{22}$$

$$\simeq P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{\Theta}_{MAP}, C_f)\Delta \tag{23}$$

where $\mathbf{\Theta}_{MAP}$ is the most probable value of the hyperparameters evaluated by the MAP method (i.e. it is the value which maximises $P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{\Theta}, C_f)$). Δ is a 'volume' term which takes into account the finite width of the $P(\mathbf{t}_N|\{\mathbf{x}_N\}, \mathbf{\Theta}, C_f)$ distribution.

In maximizing equation 21, we can consider the denominator as a constant because it is independent of $\mathbf{\Theta}$. The term $P(\Theta|\{\mathbf{x}_N\}, C_f)$ incorporates our prior knowledge of the hyperparameters. But the prior, $P(\Theta)$, is independent of either $\{\mathbf{x}_N\}$ or C_f, so

$$P(\Theta|\{\mathbf{x}_N\}, C_f) = P(\Theta) \quad . \tag{24}$$

(Note that the prior appears in equation 23.) In practice it is easier to maximize the logarithm of $(P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f)$:

$$\ln P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f) = \ln P(\mathbf{t}_N|\{\mathbf{x}_N\}, \Theta, C_f) + \ln P(\Theta|\{\mathbf{x}_N\}, C_f) \tag{25}$$

$$- \ln P(\mathbf{t}_N|\{\mathbf{x}_N\}, C_f) \quad . \tag{26}$$

Substituting equation 24 into this and collecting all terms independent of Θ into the term c, the optimum hyperparameters are given the maximum of

$$\begin{aligned}
\ln P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f) &= \ln P(\mathbf{t}_N|\{\mathbf{x}_N\}, \Theta, C_f) + \ln P(\Theta) + c && (27)\\
&= \ln\left[\frac{1}{Z_N}\exp\left(-\frac{1}{2}\mathbf{t}_N^T\mathbf{C}_N^{-1}\mathbf{t}_N\right)\right] + \ln P(\Theta) + c && (28)\\
&= -\frac{1}{2}\mathbf{t}_N^T\mathbf{C}_N^{-1}\mathbf{t}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{C}_N| + \ln P(\Theta) + c && (29)
\end{aligned}$$

where $Z_N = (2\pi)^{N/2}\sqrt{|\mathbf{C}_N|}$. For conciseness, let $\mathcal{L} = \ln P(\Theta|\mathbf{t}_N, \{\mathbf{x}_N\}, C_f)$. The derivative of this with respect to one of the hyperparameters, θ, is

$$\frac{\partial\mathcal{L}}{\partial\theta} = -\frac{1}{2}Tr\left(\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta}\right) + \frac{1}{2}\mathbf{t}_N^T\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta}\mathbf{C}_N^{-1}\mathbf{t}_N + \frac{\partial\ln P(\theta)}{\partial\theta} \tag{30}$$

where the prior on θ, $P(\theta)$ is assumed to be independent of the other priors. The maximum of this function can be found by standard optimisation procedures (such as gradient descent or conjugate gradients). Note that $\mathbf{C}_N^{-1}$ must be evaluated at each step of the optimisation algorithm. Direct methods for this include LU decomposition (so-called 'direct' and 'indirect') and Gauss-Jordan elimination, both of which are $O(N^3)$ methods. If N is large we can use Skilling's approximate inversion methods which are $O(N^2)$.