4F13 Probabilistic Machine Learning - Gaussian Processes

Lawrence Tray
St John's College

November 3, 2020

Abstract

1 Introduction

2 Questions

2.a Squared Exponential Covariance Function

We start with a simple squared exponential (SE) covariance function. As we start by working in one dimension this is necessarily isotropic. The covariance function is given by:

$$k_{SE}(x, x') = \nu^2 \exp\left\{-\frac{(x - x')^2}{2l^2}\right\} \tag{1}$$

The hyperparameters are ν and l which control the baseline variance level and length scale of variation respectively. We load in the training data from *'cw1a.mat'* and train a GP model, with zero mean and covariance function given by equation 1. We train the model by minimising the negative log marginal likelihood (denoted $\mathcal{L}$). This is achieved through the commands in listing 1.

Listing 1: Hyperparameter optimisation

```
cov = [-1, 0]; lik = 0;
hyp = struct('mean', [], 'cov', cov, 'lik', lik);
hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
```

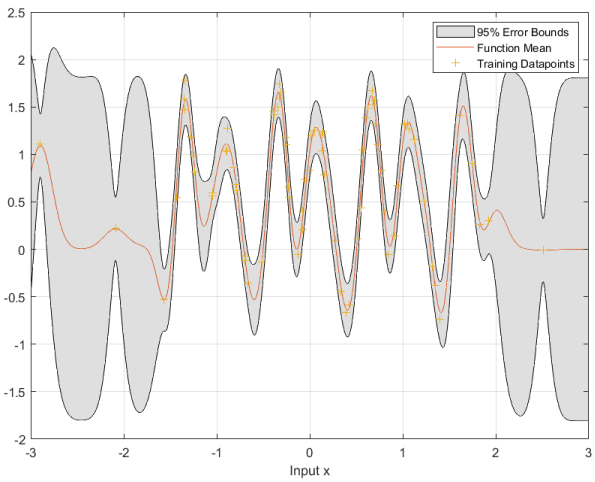
Parameter	**A Initial**	**A Final**	**B Initial**	**B Final**
$\log l$	-1	-2.054	-0.45	2.085
l	0.368	0.128	0.638	8.045
$\log \nu$	0	-0.109	0	-0.363
ν	1	0.897	1	0.696
$\mathcal{L}$	0	-2.139	0	-0.411

Table 1: Parameter optimisation (log values included for reference as reported by gpml toolbox)

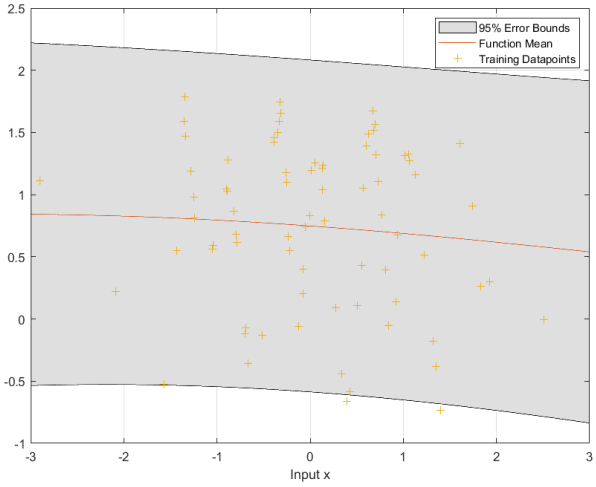
Table 1, scenario A shows one example of optimising the parameters[1] to minimise $\mathcal{L}$. This yields a predictor as in figure 1a. The 95% error bound is computed by $[\mu(x) - 2\sigma(x), \mu(x) + 2\sigma(x)]$. In other words, as $y \sim \mathcal{N}(\mu(x), \sigma(x)^2)$, there is a 95% chance of y falling within two standard deviations of the mean (all evaluated at a specific x).

We see that the error bars are always centred on the mean and that they have small standard deviations for regions in which we have many datapoints observed. This makes intuitive sense as we cannot make confident predictions in areas where the training data is sparse (such as for $|x| \geq 3$). The hyperparameters do not change enormously form the optimisation. We have that the length scale of variation l shrinks slightly to 0.128 - which agrees with the length scale of variation in the dataset. The scale factor ν also shrinks slightly to 0.897 as the model is able to match the data quite accurately.

[1]The negative log-likelihood $\mathcal{L}$ is not strictly a hyperparameter as we do not directly control its value but it is included in this table for ease of reference



(a) Basic hyperparameter initialisation



(b) Alternative hyperparameter initialisation

Figure 1: Squared Exponential covariance Gaussian Process trained on data

2.b Hyperparameter Initialisation

However, the optimisation only finds a local minimum of the negative log-likelihood $\mathcal{L}$. Therefore, a different intialisation of the hyperparameters can yield different results. This is illustrated in scenario B of table 1. It was found that an initial value of $\log l = -0.45$ was a critical point. Any setting of $\log l$ above this would converge to the case-B optimum; anything below converges to the original case-A optimum. Varying ν only seemed to change the position of this critical point but would not converge to an altogether different solution.

This alternative optimum converges to a very large value of the length scale $l = 8.045$. This model expects data to vary very slowly with respect to x and indeed attributes all variation within the observed range to noise. Indeed, figure 1b shows that the mean varies little over the data range. Indeed, the error bars remain nearly constant and very large. This model does not seem to fit the data well just by comparing figures 1a and 1b. This intuition can be quantified through the marginal log-likelihood. Model B has a far smaller marginal likelihood $|\mathcal{L}_B| = 0.411, |\mathcal{L}_A| = 2.139$ and so we can conclude that this model is more improbable.

2.c Periodic Covariance Function

We can instead use a periodic covariance function to model the data (as in equation 2. We introduce a new parameter p which is the period of the repetition in covariance. l should no longer be thought of as a length scale but rather sets the strength of the correlation between neighbours and v sets the baseline variance.

$$k_{PER}(x, x') = \nu^2 \exp\left\{-\frac{2\sin^2\left(\pi(x - x')/p\right)}{l^2}\right\} \tag{2}$$

We tune the hyperparameters in much the same way as before; the results are summarised in table 2. All hyperparameters stay close to their initial values. Indeed, the period p stays very close to 1.

Parameter	Initial	Final
$\log l$	0	0.0437
l	1	1.044
$\log p$	0	-0.0012
p	1	0.999
$\log \nu$	0	0.2122
ν	1	1.24
$\mathcal{L}$	0	-2.213

Table 2: Periodic covariance function hyperparameter tuning

We can plot the predictions on figure 2 which shows excellent agreement between model and data. Indeed, the error bars are

uniformly small. On first inspection this appears to be a more natural fit than the squared exponential covariance function. However, we must be careful to avoid overfitting.

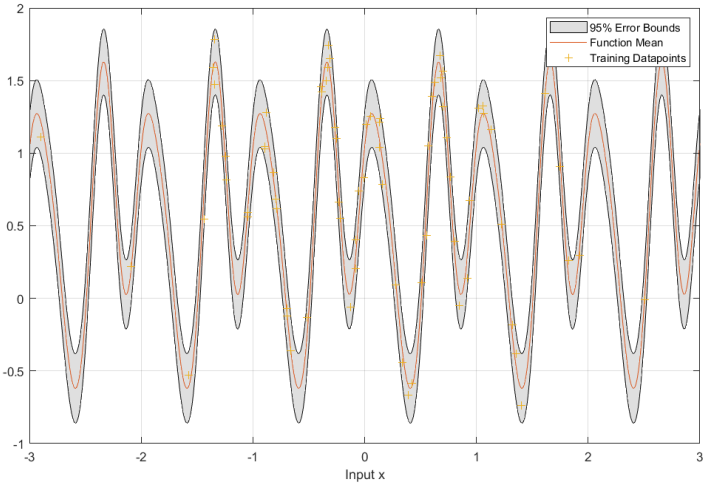


Figure 2: Periodic covariance GP on same training data

Indeed, the marginal likelihood is slightly greater for the periodic covariance model. Nevertheless, it does not appear that the training data was generated in a periodic manner. On figure 3, we plot a histogram of the training inputs. This appears Gaussian with zero mean and modest variance $O(1)$. Therefore, it is highly unlikely that the data was generated in a periodic manner across the range [-3, 3]. We prefer the model in part a (SE covariance) as this remains uncertain in areas of sparse data.

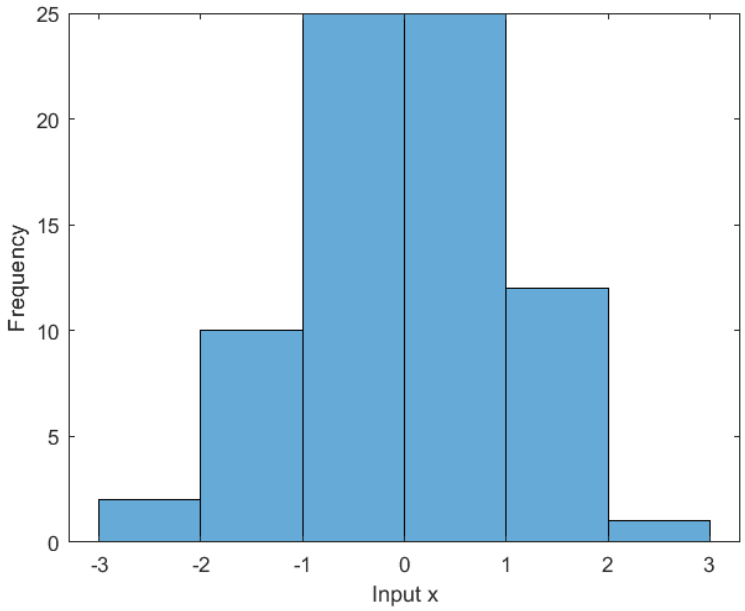


Figure 3: Histogram of training inputs

2.d Sampling from a Gaussian Process

2.e 2-Dimensional Input - Model Comparison

We now extend our investigation to deal with 2-dimensional input data. For this we need a new covariance function. For the simple case we choose a Squared Exponential - Automatic Relevance Determination (SE-ARD) covariance function. This is in general anisotropic and has formula given by equation 3. The index i iterates through every dimension in our input (total D dimensions). In this example, the data is 2-dimensional ($D = 2$).

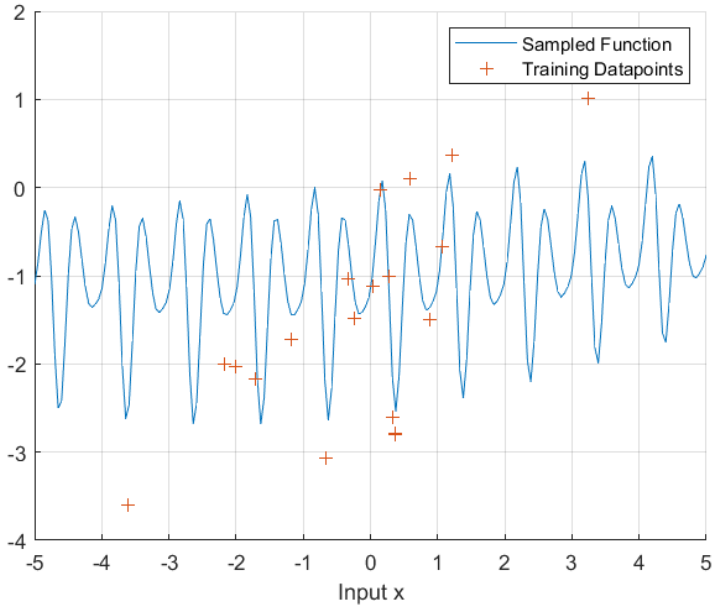


Figure 4: Trained GP with initial hyperparameter settings

$$k_{ARD}(\mathbf{x}, \mathbf{x}') = \nu^2 \exp\left\{-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i - x_i')^2}{l_i^2}\right\} \tag{3}$$

We wish to compare this simple model with a more complex model. The covariance functions for the simple and additive model are given in equations 4 and 5 respectively.The vector $\theta = [\nu, l_1, l_2]$ denotes the hyperparameters of the SE-ARD covariance function.

$$k^a(\mathbf{x}, \mathbf{x}') = k_{ARD}(\mathbf{x}, \mathbf{x}'; \theta^a) \tag{4}$$

$$k^b(\mathbf{x}, \mathbf{x}') = k_{ARD}(\mathbf{x}, \mathbf{x}'; \theta^{b1}) + k_{ARD}(\mathbf{x}, \mathbf{x}'; \theta^{b2}) \tag{5}$$

θ^a	**Initial**	**Final**
$\log l_1$	0	0.413
l_1	1	1.511
$\log l_2$	0	0.252
l_2	1	1.287
$\log \nu$	0	0.102
ν	1	1.107

(a) Simple model a

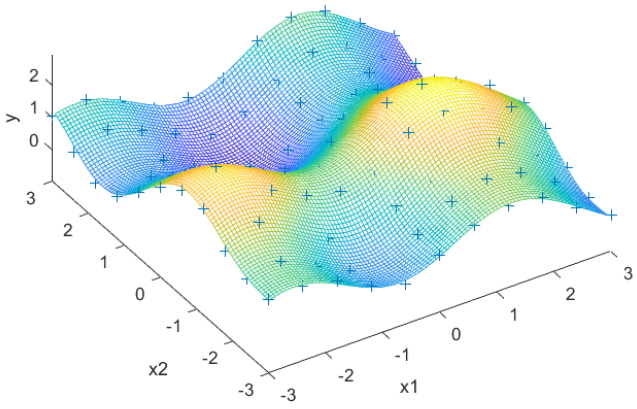
θ^{b1}	**Initial**	**Final**
$\log l_1$	0	0.3644
l_1	1	1.440
$\log l_2$	0	6.3200
l_2	1	555.6
$\log \nu$	0	0.0801
ν	1	1.083

(b) Component b1

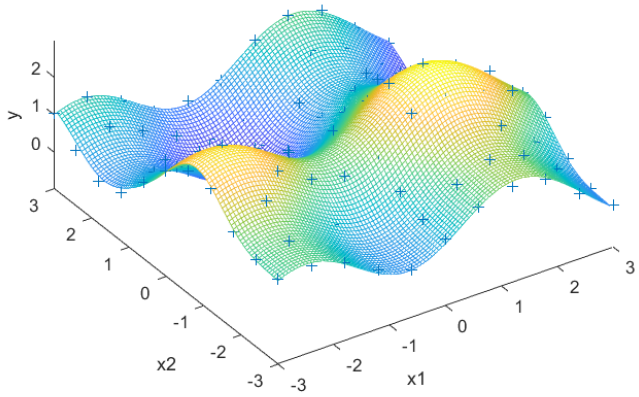
θ^{b1}	**Initial**	**Final**
$\log l_1$	0	6.000
l_1	1	403.4
$\log l_2$	0	-0.0072
l_2	1	0.993
$\log \nu$	0	-0.3351
ν	1	0.715

(c) Component b2

Table 3: Parameter optimisation for 2-D data

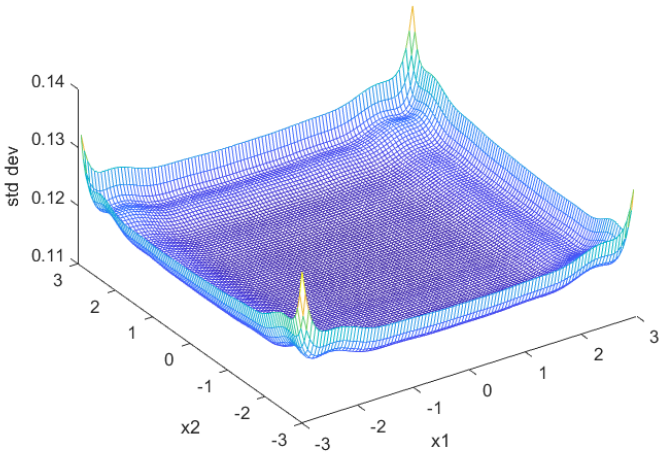


(a) Basic covariance function: single covSEard model

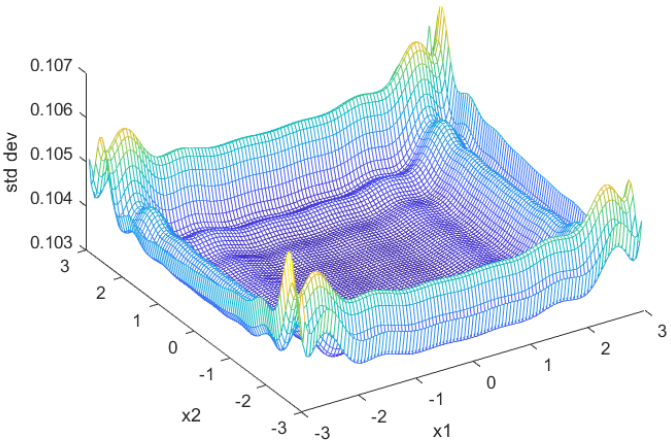


(b) Additive covariance function: two covSEard model

Figure 5: Comparison of two covariance function fits on training data



(a) Basic covariance function: single covSEard model



(b) Additive covariance function: two covSEard model

Figure 6: Comparison of standard deviation for