4F13 Probabilistic Machine Learning - Gaussian Processes

Lawrence Tray
St John's College

November 5, 2020

Abstract

This report outlines the results of the first coursework for 4F13. Various properties of Gaussian Processes were investigated and how they can be used to model arbitrary data through careful choice of the covariance function. We focus on squared exponential (SE), periodic and composite combinations of these two covariance functions.

Contents

1 Questions

1.a Squared Exponential Covariance Function

We start with the squared exponential (SE) covariance function (equation 1). For 1-D inputs this is necessarily isotropic.

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

The hyperparameters are ν - scale factor and λ - length scale. We load in the training data from *'cw1a.mat'* and train a GP model, with zero mean and SE covariance. We assume a Gaussian likelihood function with noise variance σ^2 such that $P(y|\mu) = \mathcal{N}(y; \mu, \sigma^2)$, where μ is the associated mean. We train the model hyperparameters by minimising the negative log marginal likelihood - denoted $\mathcal{L}$. This is achieved through the commands in listing 1.

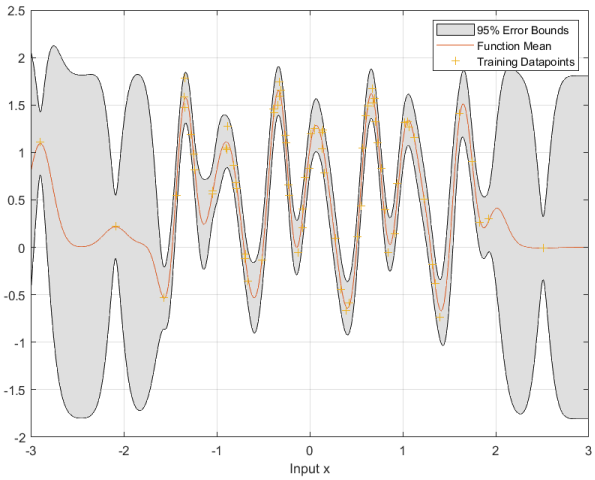
Listing 1: Hyperparameter optimisation

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

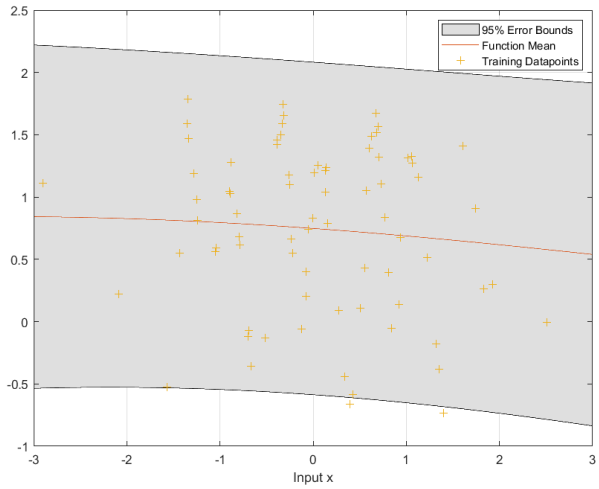
Parameter	A Initial	A Final	B Initial	B Final
$\log\lambda$	-1	-2.054	-0.45	2.085
λ	0.368	0.128	0.638	8.045
$\log\nu$	0	-0.109	0	-0.363
ν	1	0.897	1	0.696
$\log\sigma$	0	-2.139	0	-0.411
σ	1	0.118	1	0.663
$\mathcal{L}$	92.9	11.9	92.5	78.2

Table 1: Hyperparameter optimisation (log values included as reported by gpml toolbox)

Table 1 scenario A, shows one example of optimising the parameters[1] to minimise the negative log-likelihood. This yields a predictor as in figure 1a. The 95% error bound is computed as within two standard deviations of the mean.



(a) Basic optimised hyperparameters (case A)



(b) Alternative optimised hyperparameter (case B)

Figure 1: SE covariance GP trained on data from *'cw1a.mat'*

We see that the error bars are small in regions of high data density; we cannot make confident predictions in areas of sparse data (e.g. $|x| \geq 2$).

We have that the length scale of variation λ shrinks slightly to 0.128 - which agrees with the length scale of variation in the dataset. The model is able to match the data quite accurately. Indeed, the noise variance σ shrinks significantly from 1 to 0.118 meaning the error bars are quite narrow in regions of high data density; the error bars remain large outside these regions due to the modest baseline variance $\nu = 0.897$.

1.b Hyperparameter Initialisation

However, the optimisation only finds a local minimum of the negative log-likelihood $\mathcal{L}$. Therefore, a different initialisation of the hyperparameters can yield different results (case B of table 1). It was found that an initial value of $\log \lambda^{(0)} = -0.45$ was a critical point. Any setting of $\log \lambda^{(0)}$ above this would converge to the case-B optimum; anything below converges to the original case-A optimum. Varying $\nu^{(0)}$ only seemed to change the position of this critical point but would not converge to an altogether different solution.

This alternative optimum (B) converges to a very large value of the length scale $\lambda = 8.045$. This expects data to vary very slowly with respect to x and attributes all variation within the observed range to noise. Indeed, figure 1b shows that the mean varies little over the data range. Parameter setting B 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. Parameter setting B has a far larger final value of $\mathcal{L}$, ($\mathcal{L}_B = 78.2 >> 11.9 = \mathcal{L}_A$) so we can conclude that B is much more unlikely.

1.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 ρ, the period length. λ should no longer be thought of as a length scale but rather sets the strength of the correlation between neighbours and ν scales the covariance. We tune the hyperparameters in the same way as before (results in table 2).

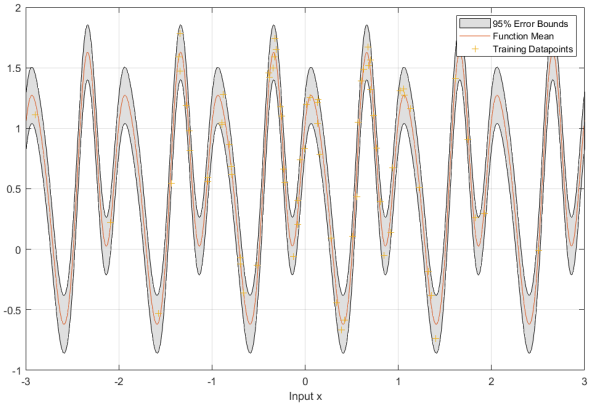
$$k_{PER}(x, x') = \nu^2 \exp\left\{-\frac{2\sin^2\left(\pi(x - x')/\rho\right)}{\lambda^2}\right\} \tag{2}$$

We plot the trained GP predictions on figure 2a which shows excellent agreement between model and data. The error bars are uniformly small - even in regions of sparse data. This initially appears a better fit than the SE model. However, we must be careful to avoid overfitting.

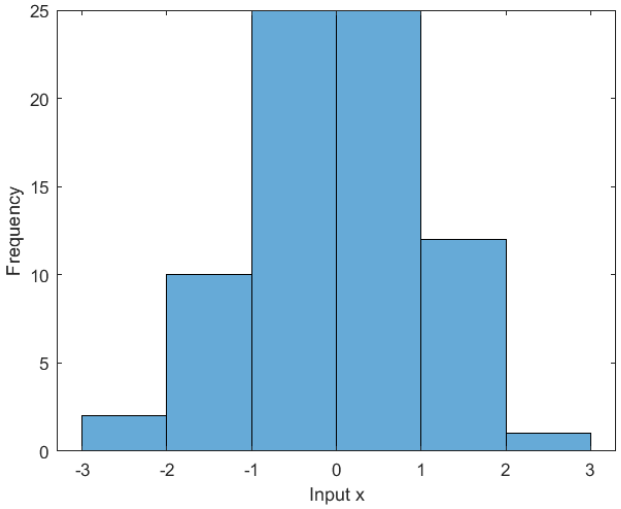
[1]The negative log marginal likelihood $\mathcal{L}$ is not a hyperparameter but rather a function evaluated on the dataset given the current setting of the hyperparameters. It is included in this table for reference.

Parameter	Initial	Final
$\log \lambda$	0	0.0437
λ	1	1.044
$\log \rho$	0	-0.0012
ρ	1	0.999
$\log \nu$	0	0.2122
ν	1	1.24
$\log \sigma$	0	-2.213
σ	1	0.109
$\mathcal{L}$	79.5	-35.3

Table 2: Periodic covariance function hyperparameter tuning



(a) Periodic covariance GP with tuned hyperparameters



(b) Histogram of training inputs

Figure 2: Periodic covariance fit on data from *'cw1a.mat'*

Indeed, the log-likelihood is far larger for the periodic covariance model compared to the SE model. Nevertheless, it does not appear that the training data was generated in a periodic manner. On figure 2b, we plot a histogram of the training inputs. This appears Gaussian and not uniform. Therefore, it is highly unlikely that the data was generated by applying a periodic function to uniformly generated inputs across the range [-3, 3]. We prefer the model in part a (SE covariance) as this remains uncertain in areas of sparse data.

1.d Sampling from a Gaussian Process

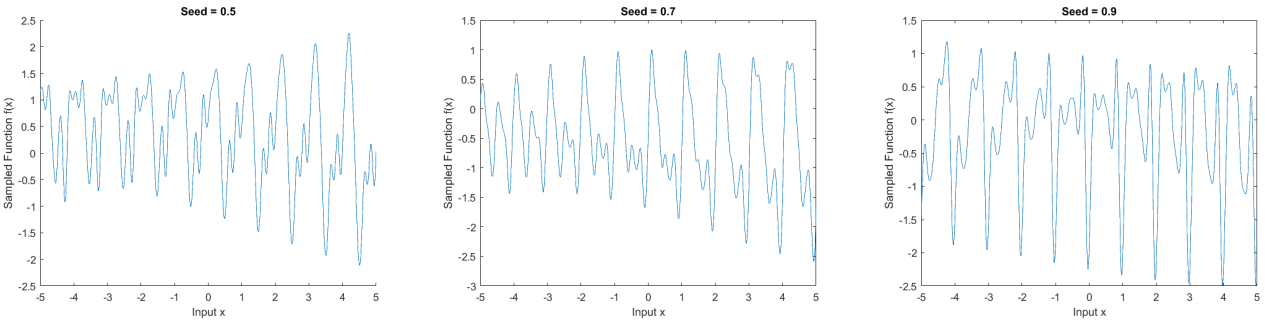
We can sample from an arbitrary GP at a finite array of input points $\mathbf{x}$, by evaluating the covariance matrix K for these points and exploiting the Cholesky Decomposition ($\text{chol}(K) = C$ s.t. $CC^T = K$). The code for this is in listing 2. We must add a small diagonal matrix to ensure that K is positive definite and thus accepts a Cholesky decomposition.

Listing 2: GP sampling

```
x = linspace(-5, 5, n)';
z = gpml_randn(seed, n, 1);
K = feval(covfunc{:}, cov, x);
K_pos_def = K + 1e-6 * eye(n);
y = chol(K_pos_def)'*z;
```

We use this code to sample from a GP with composite covariance function defined as the product of a periodic covariance function and a SE covariance function. The hyperparameters are set to $[\lambda, \rho, \nu]_{PER} = [0.607, 1, 1]$ for the periodic component and $[\lambda, \nu]_{SE} = [7.39, 1]$ for the SE component. We can run the code in 2 for different values of the seed to yield the set of plots in figure 3.

This set of characteristic functions share some key properties. The functions do appear in some way periodic with period 1. This agrees with the hyperparameter setting $\rho_{PER} = 1$. The SE component has a very long length scale $\lambda_{SE} = 7.39$ This



(a) Seed = 0.5 (b) Seed = 0.7 (c) Seed = 0.9

Figure 3: Random function drawn from GP with composite product covariance: periodic × SE

means that we only see distortion of the periodic structure over long length scales; adjacent periods are very similar but those that are further apart have different form.

1.e 2-Dimensional Input - Model Comparison

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

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

We wish to compare this simple model with a more complex, additive model. The covariance functions for the simple and additive model are given in equations 4 and 5. The vector $\theta = [\nu, \lambda_1, \lambda_2]$ denotes the hyperparameters of the SE-ARD covariance function.

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

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

The optimised parameters are presented in table 3. We can plot the mean of the predictive surface yielded by each model on figure 4. We see that both models fit the data quite well. Nevertheless, the additive model has a much lower $\mathcal{L}$ and so matches the training data more closely. This is to be expected as model a is a special case of model b where $\nu^{(b2)} = 0$. The additional degrees of freedom introduced by the more complex, additive model mean it can fit the data more closely and thus increases the marginal likelihood.

θ^a	**Final**
$\log \lambda_1$	0.413
λ_1	1.511
$\log \lambda_2$	0.252
λ_2	1.287
$\log \nu$	0.102
ν	1.107

(a) Simple model a

θ^{b1}	**Final**
$\log \lambda_1$	0.3644
λ_1	1.440
$\log \lambda_2$	6.3200
λ_2	555.6
$\log \nu$	0.0801
ν	1.083

(b) Component b1

θ^{b1}	**Final**
$\log \lambda_1$	6.000
λ_1	403.4
$\log \lambda_2$	-0.0072
λ_2	0.993
$\log \nu$	-0.3351
ν	0.715

(c) Component b2

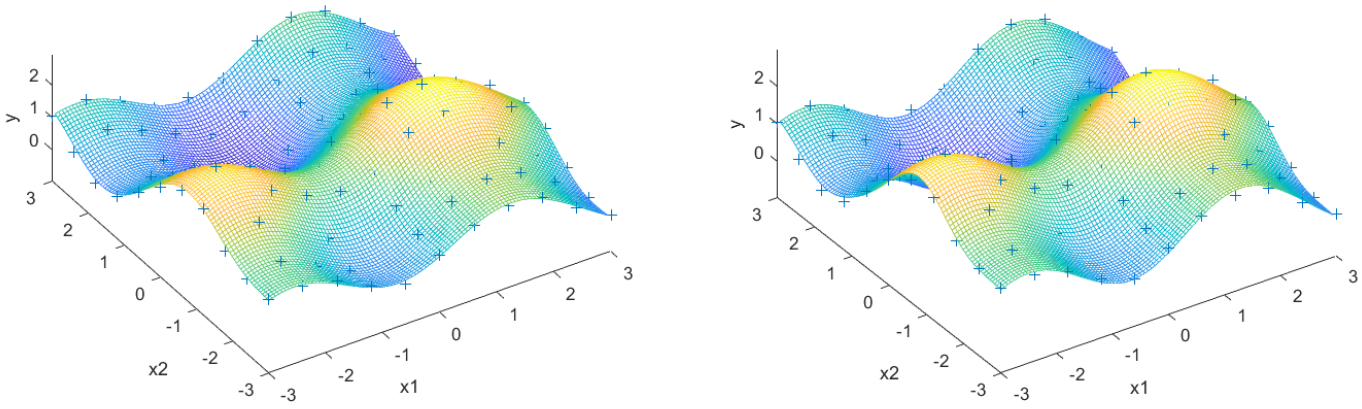
	Model a	**Model b**
$\mathcal{L}$	-19.2	-66.3

(d) Negative Log-likelihood

Table 3: Parameter optimisation for 2-D data

Nevertheless, figure 4 is hard to interpret so instead we plot the standard deviation surface of the trained GP's over the domain of interest (figure 5). For the simple model, the standard deviation surface (this is a measure of the uncertainty of each prediction) is approximately flat, only rising at the edges of the observed data. In particular the corners are highly uncertain as these have the fewest neighbouring data-points. The additive model has more ripples in its uncertainty surface due to its increased complexity. However, the magnitude of this uncertainty is uniformly lower.

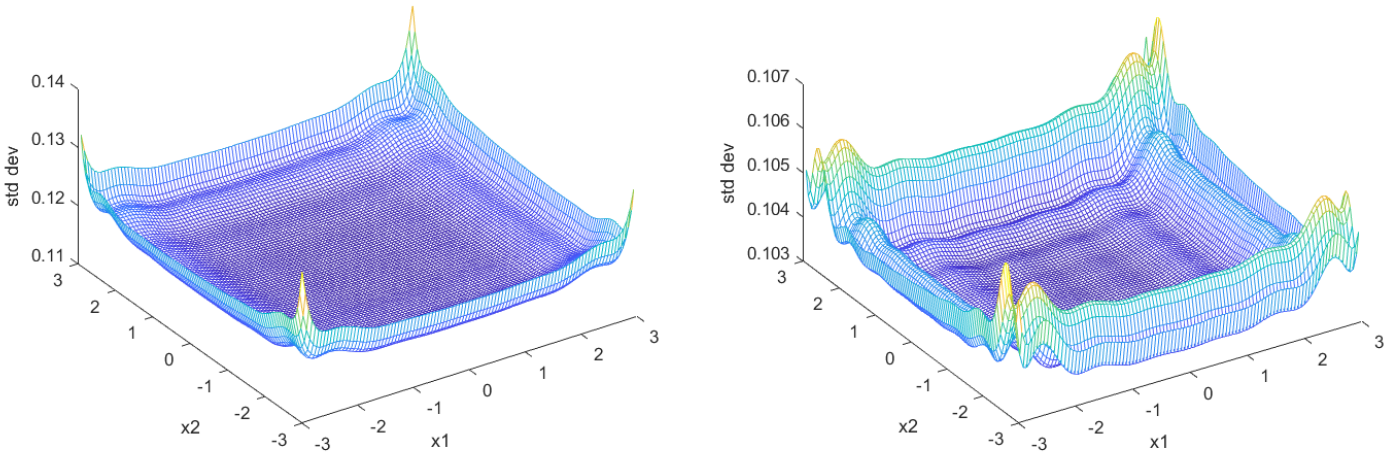
Words: 987



(a) Basic covariance function

(b) Additive covariance function

Figure 4: Comparison of covariance function fits on training data from *'cw1e.mat'*



(a) Basic covariance function

(b) Additive covariance function

Figure 5: Comparison of standard deviation for the two models on *'cw1e.mat'*