

# 4F13 Coursework 1 - Gaussian Processes

October 2024

## 1 Task A

```
meanfunc = []; hyp_init.mean = [];
covfunc = @covSEiso; hyp_init.cov = [-1 0];
likfunc = @likGauss; hyp_init.lik = 0;
hyp_opt = minimize(hyp_init, @gp, -100, @infGaussLik,
                  meanfunc, covfunc, likfunc, x, y);
[mu, s2] = gp(hyp_opt, @infGaussLik, meanfunc, covfunc, likfunc, x, y, xs);
```

Listing 1: Code to train hyper-parameters and generate the predictive distribution of a GP with squared exponential covariance

Our model has a GP with squared exponential (SE) covariance function and zero mean along with a gaussian noise. The SE covariance function has two hyper-parameters,  $\lambda$  which controls the characteristic length scale of the process and  $\sigma_f^2$  which controls the variance in the output value of the GP. The gaussian noise has one hyper-parameter,  $\sigma_n^2$ , the variance of our measurement noise.

$$y = f(x) + \eta : \eta \sim \mathcal{N}(0, \sigma_n^2)$$

$$f = \mathcal{N}(0, k(x, x'))$$

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')'(x - x')}{2\lambda^2}\right)$$

	$\lambda$	$\sigma_f$	$\sigma_n$	$\ln(Z_{ \mathbf{y}})$
Initial	0.3679	1	1	$-9.2910 \times 10^1$
Optimised	0.1282	0.8970	0.1178	$-1.1899 \times 10^1$

Table 1: Initial and optimum hyper-parameter values when maximising log marginal likelihood of data

When we train our model hyper parameters to maximise log marginal likelihood we achieve the optimum values shown in Table 1. The covariance length scale,  $\lambda$  is similar to the length scale of the undulations in our data. The  $\sigma_f$  is much larger than  $\sigma_n$  this makes sense as most of the variation in our data can be explained by our GP instead of the measurement noise.

We plot the mean and 95% error bars of the predictive distribution (Equ.1) in Fig 1. We can see that the predictive error bars are smaller where we have data and return to the prior variance ( $k(x, x) = \sigma_f^2$ ) plus  $\sigma_n^2$  when there is no nearby data. This is clear from the form of the predictive distribution covariance, the first term is the prior covariance, the second is the noise variance and the third reduces the variance based on the distance between the sample point and our data.

$$y_* \sim \mathcal{N}(m_{|\mathbf{y}}(x_*), k_{|\mathbf{y}}(x_*, x_*))$$

$$m_{|\mathbf{y}}(x) = m(x) + k(x, \mathbf{x})[k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1}(\mathbf{y} - m(\mathbf{x})) \quad (1)$$

$$k_{|\mathbf{y}}(x, x') = k(x, x') + \sigma_n^2 - k(x, \mathbf{x})[k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1}k(\mathbf{x}, x')$$

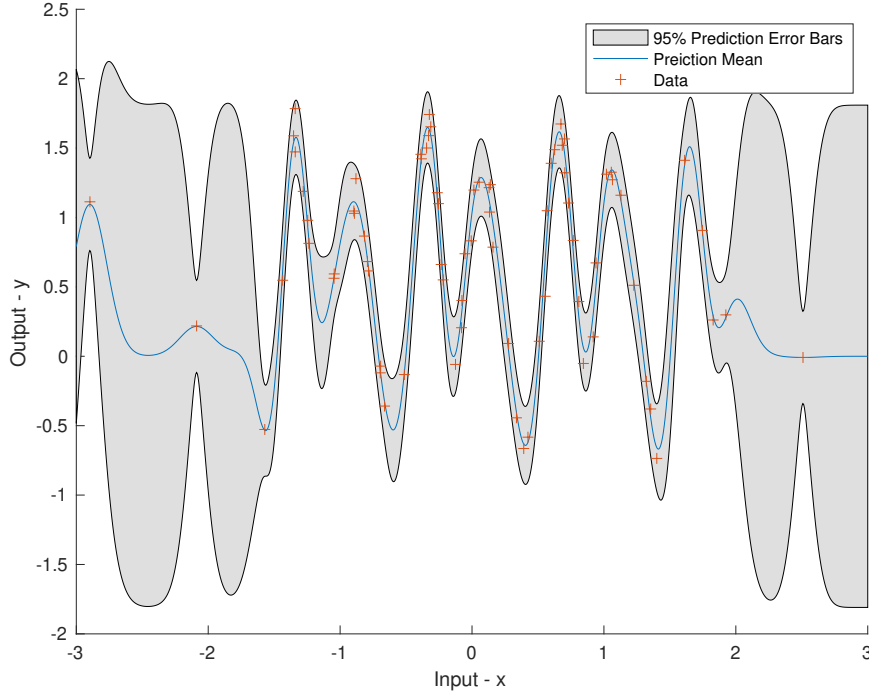


Figure 1: Mean and 95% prediction interval of GP with squared exponential covariance function, optimised hyper-parameters used given in Table 1

## 2 Task B

```

sf = 0;
[ell, sn] = meshgrid(-4:0.1:4, -4:0.1:1);
nlZ = zeros(size(ell));
for i = 1:numel(ell)
    hyp = struct('mean', [], 'cov', [ell(i) sf], 'lik', sn(i));
    nlZ(i) = gp(hyp, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
end
minima = islocalmin2(nlZ);

```

Listing 2: Code to sweep  $\lambda$  and  $\sigma_n$  to identify local minima of  $\log(Z_{|\mathbf{y}})$

	$\lambda$	$\sigma_f$	$\sigma_n$	$\log(Z_{ \mathbf{y}})$
Optimum 1	0.1282	0.8970	0.1178	$-1.1899 \times 10^1$
Optimum 2	8.0492	0.6957	0.6631	$-7.8220 \times 10^1$

Table 2: Hyper-parameter values at 2 local maxima of log marginal likelihood

## 3 Task C

```

covfunc = @covPeriodic; hyp_init.cov = [-1 0 0];

```

Listing 3: Code to use periodic SE covariance. Training and prediction code same as Listing 1

## 4 Task D

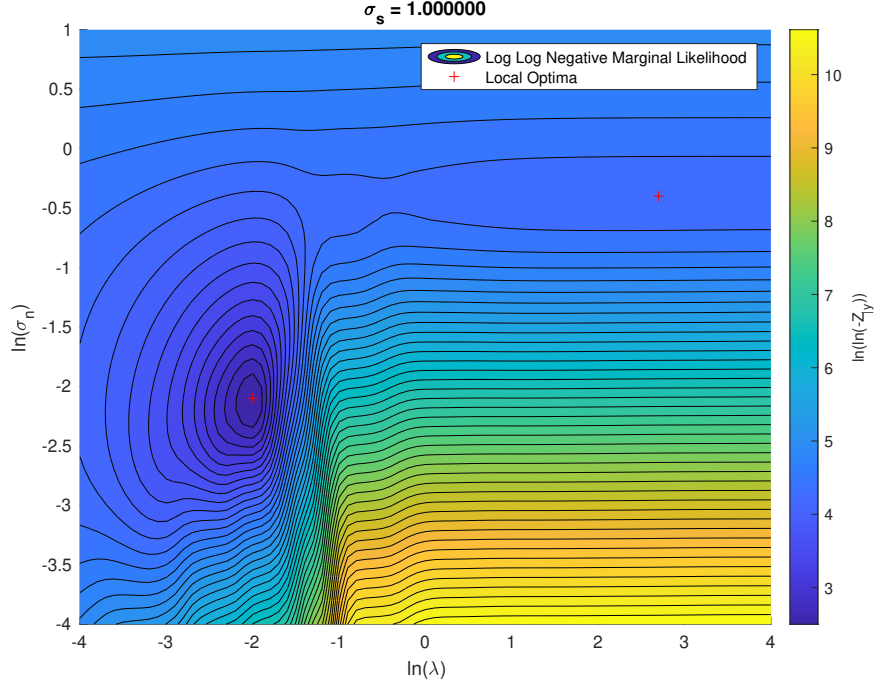


Figure 2: Contour of  $\ln(\ln(Z_{|\mathbf{y}}))$  with  $\lambda$  and  $\sigma_n$  for fixed  $\sigma_f = 1$ . Shows the two local maxima of marginal likelihood given in Table 2

$\lambda$	$p$	$\sigma_f$	$\sigma_n$	$\log(Z_{ \mathbf{y}})$
0.7048	0.9991	0.6941	0.0851	$2.9300 \times 10^1$

Table 3: Hyper-parameter values for periodic SE covariance function

```

N_points = 200; N_samples = 3;
x = linspace(-5,5,N_points)';
cov_func = {@covProd, {@covPeriodic, @covSEiso}}; hyp.cov = [-0.5 0 0 2 0];
K = feval(cov_func{:}, hyp.cov, x);
y = chol(K + 1e-6*eye(N_points))' * gpml_randn(2, N_points, N_samples);

```

Listing 4: Code to sample a GP with covariance defined by the product of a periodic SE kernel and a SE kernel

$p^1$	$\lambda^1$	$\sigma_f^1$	$\lambda^2$	$\sigma_f^2$
1	0.6065	1	7.3891	1

Table 4: Hyper-parameter values for periodic SE covariance function

## 5 Task E

```

[xs1, xs2] = meshgrid(linspace(-10, 10, 101), linspace(-10, 10, 101));
xs = [reshape(xs1, [], 1), reshape(xs2, [], 1)];
mean_func = []; hyp_init.mean = [];
cov_func = {@covSum, {@covSEard, @covSEard}}; hyp_init.cov = 0.1*randn(6,1);
lik_func = @likGauss; hyp_init.lik = 0;
hyp_opt = minimize(hyp_init, @gp, -100, @infGaussLik,
                  mean_func, cov_func, lik_func, x, y);
[mu, s2] = gp(hyp_opt, @infGaussLik, mean_func,
              cov_func, lik_func, x, y, xs);

```

Listing 5: Code to train and calculate prediction intervals of a 2D GP with SE ARD kernel

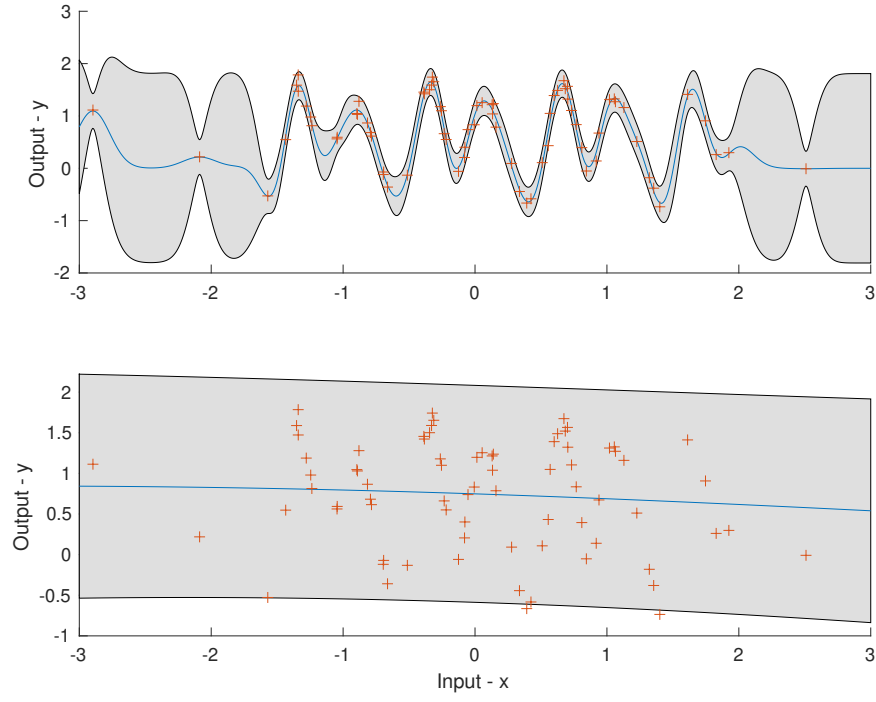


Figure 3: Prediction intervals of GP with hyper-parameters from Table 2. Top: Optimum 1, Bottom: Optimum 2

	$\sigma_f^1$	$\lambda_1^1$	$\lambda_2^1$	$\sigma_f^1$	$\lambda_1^1$	$\lambda_2^1$	$\sigma_n$	$\ln(Z_{ \mathbf{y} })$
$covSEard$	1.107	1.511	1.285	-	-	-	0.1026	$1.9218 \times 10^1$
$covSEard + covSEard$	0.7116	1104	0.9864	1.108	1.446	1281	0.0979	$6.6394 \times 10^1$

Table 5: Hyper-parameter values for periodic SE covariance function

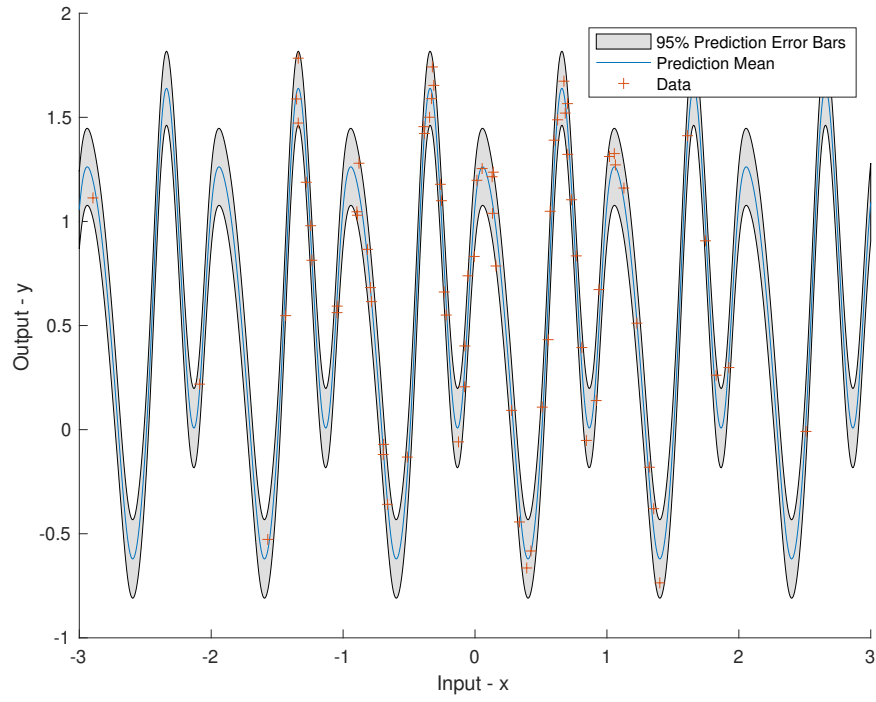


Figure 4: Prediction intervals for periodic SE covariance function with hyper parameters given in Table 3

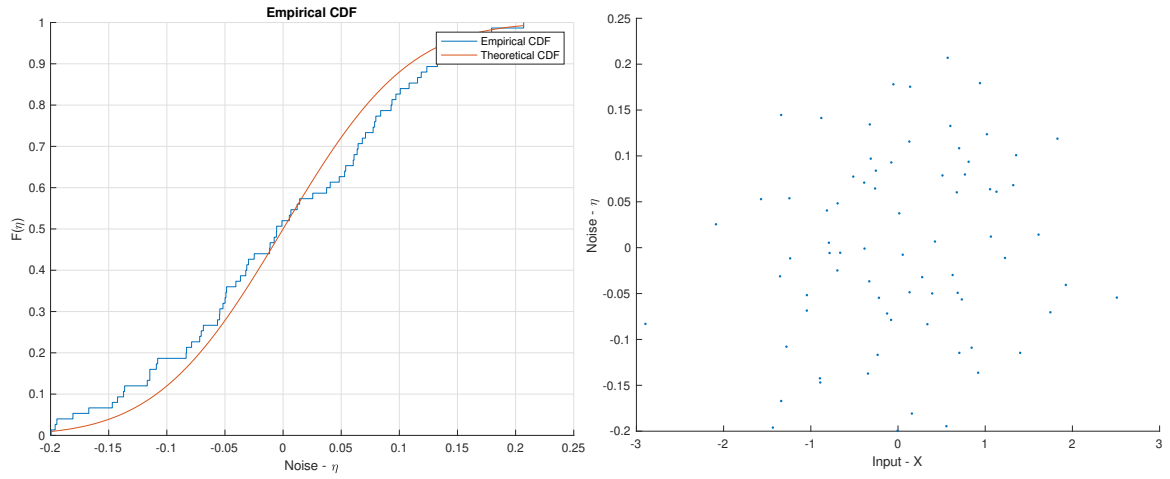


Figure 5: Caption for figure 2

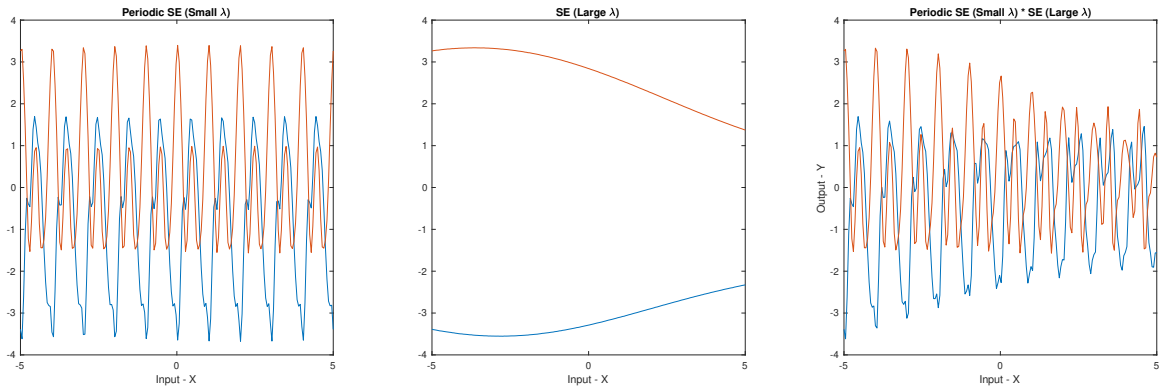


Figure 6: Samples from 3 covariance kernels, hyper-parameters for each given in Table

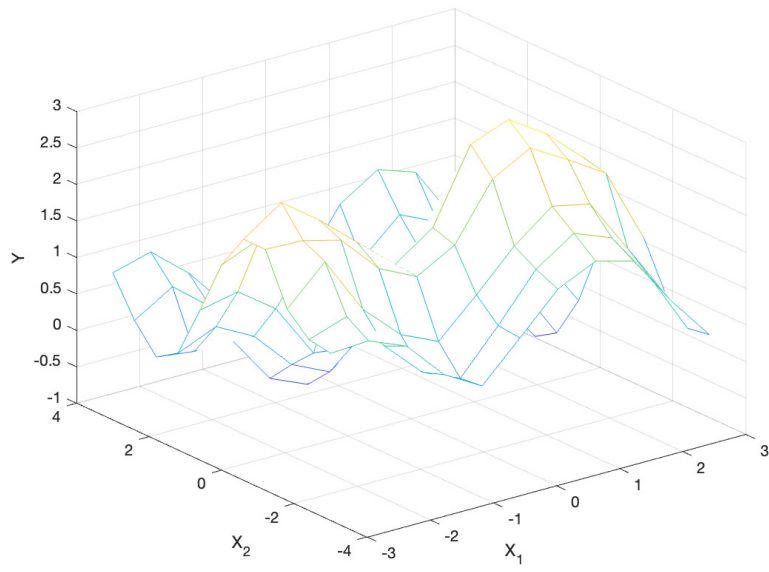


Figure 7: Visualisation of data in 'cw1e.mat'

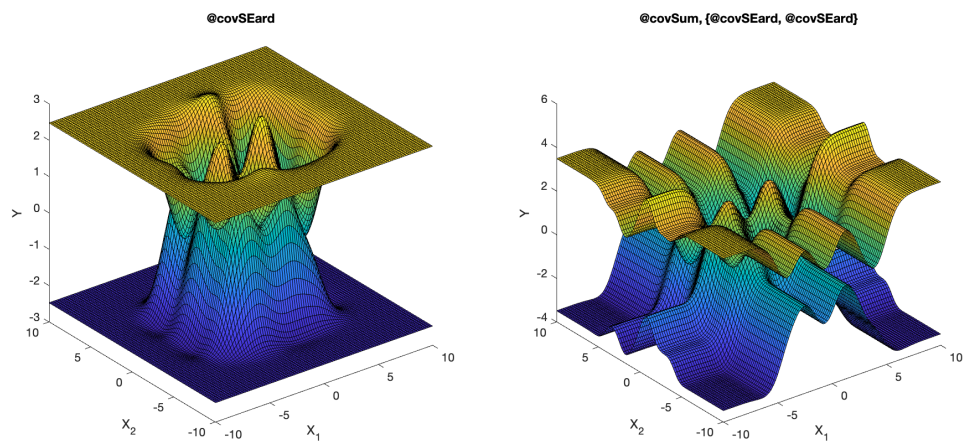


Figure 8