Coursework 1 - Gaussian Processes

Candidate number: H801L
word count: 998

5th November 2020

1 Part A

Figure 1 shows a Gaussian posterior predictive distribution evaluated at 1000 x-points. There seems to be a periodic trend in the data with majority of samples lying in the interval $(-1.5, 1.5)$. In this interval the uncertainty is relatively small with average standard deviation equal to: 0.15. This is due to a high number of data points in this interval and low σ_{noise}. Outside of the (-1.5,1.5) interval the there are less data points and the uncertainty is high, due to large σ_{signal}. It defines our prior belief about function variability (uncertainty before seeing data).

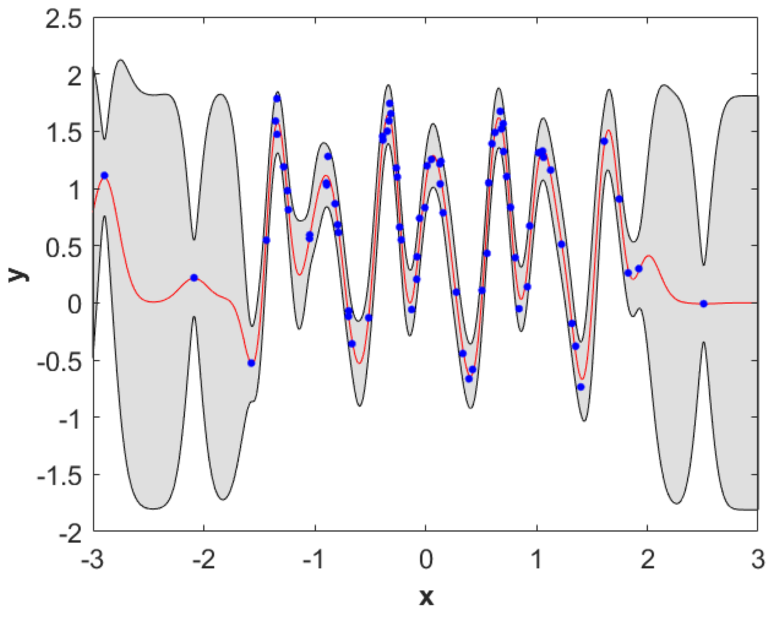
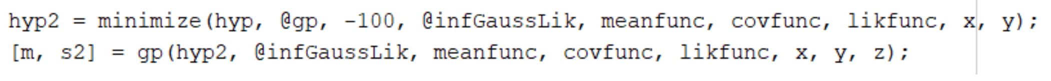


Figure 1: Gaussian process posterior predictive distribution, with Squared Exponential covariance function with lengthscale $l = 0.13$, signal standard deviation $\sigma_{signal} = 0.90$ and the noise standard deviation is $\sigma_{noise} = 0.12$

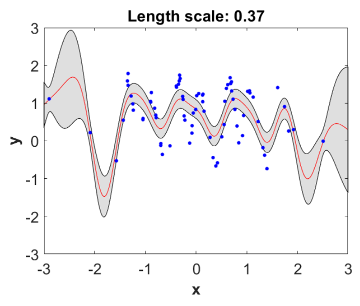


```
hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
[m, s2] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, z);
```

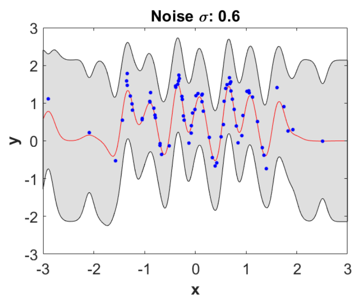
Figure 2: minimize(...) function was used to optimise hyperparameters. gp(...) computes means and variances of underlying function.

Noise standard deviation defines our uncertainty about the actual values $f(x)$ of sampled training data. The greater it is the more relaxed $\boldsymbol{\mu}$ becomes as more data variations can be explained by noise (see Figure 3.b).

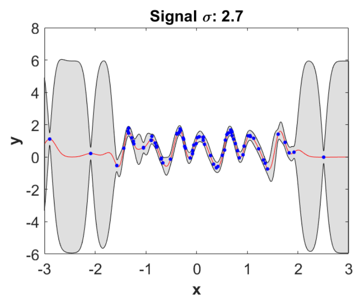
Optimised **lengthscale** assumes value $l = 0.13$ in order to capture local trend in the data - the smaller it is, the less influential far away points become. In Figure 3.a we show that $l = 0.37$ is too large to capture the local curvature.



(a) Increased lengthscale



(b) Increased σ_{noise}

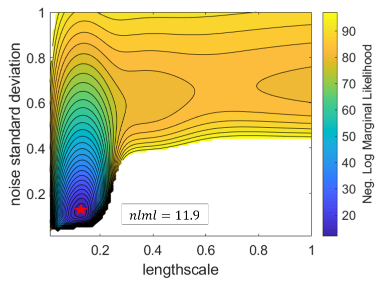


(c) Increased σ_{signal}

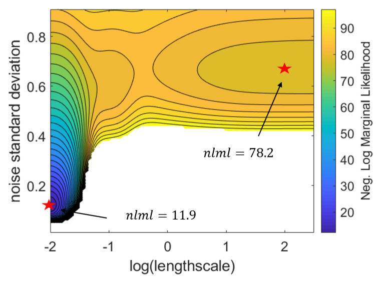
Figure 3: Effect of increasing hyperparameters on the modell fit, while keeping other hyperparameters as identified in Figure 1.

2 Part B

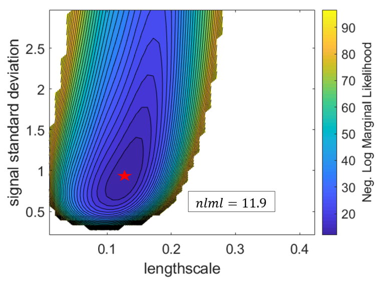
When optimising hyperparameters using negative log marginal likelihood using gradient descent algorithms [3], we may often arrive at different local optima, depending on the initial choice of hyperparameters. Figure 4 shows a plot of negative log marginal likelihood as a function of two hyperparameters, while third being held constant. It allows us to identify two local minima for the problem presented in Part A. First one has $nlml = 11.9$ and another $nlml = 78.2$. Fits of both models are presented in Figure 5.



(a)



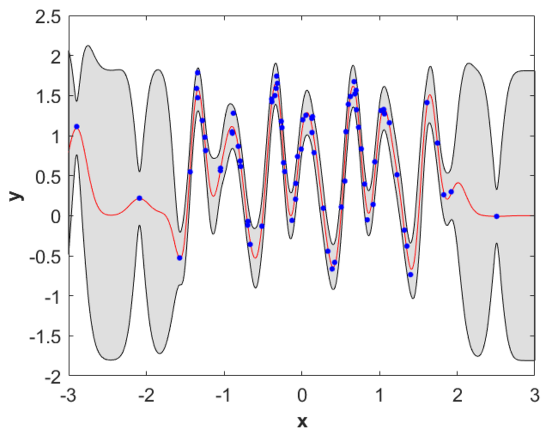
(b)



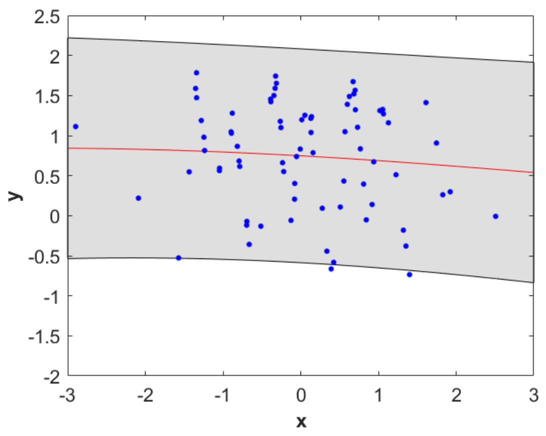
(c)

Figure 4: Surface plots of Negative Log Marginal likelihood as a function of two hyperparameters. For each two hyperparameters being varied, the third was kept constant at it's optimal value stated in Figure 1.

First model (Figure 5.a) assumes low lengthscale $l = 0.13$ such that the SE covariance can follow the local data trend. As a result, the mean function $\boldsymbol{\mu}$ passes very close to each data point and thus keeping the noise low ($\sigma_{noise} = 0.12$). The second model (Figure 5.b) assumes high lengthscale $l = 7.91$ (nearly flat) and attempts to classify all the data variation as being caused by a noise ($\sigma_{noise} = 0.66$).



(a) $nlml = 11.9$



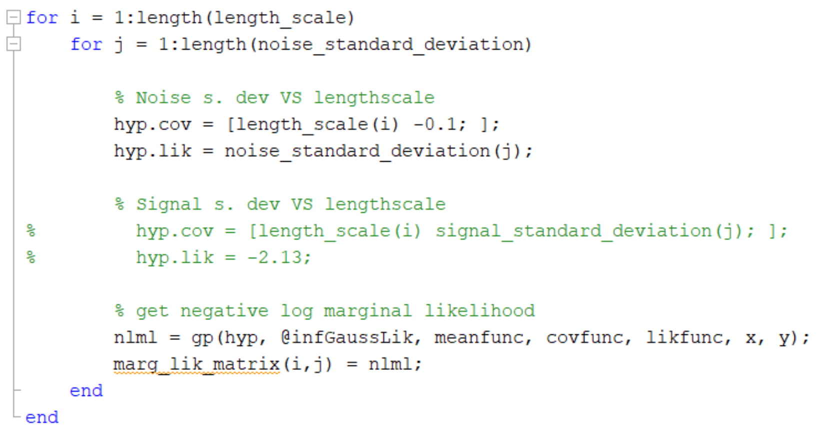
(b) $nlml = 78.2$

Figure 5: Two local optima of Negative log marginal likelihood for a GP with SE kernel.

While in a fully Bayesian approach we could weight predictions from both models, here this is unnecessary because model in 5.b has much higher Negative Log Marginal Likelihood (see Table 1). While both models fit the data well (all points lie within or very close to the 95% error bars), the large σ_{noise} of model in Figure 5.b makes it a more complex model, as the space of datasets it can represent is larger. Using Occam's razor principle we prefer GP in Figure 5.a as it has similar data fit but lower complexity.

	Log Marginal Likelihood	Data Fit	Neg. Model Complexity
Figure 5.a	-11.9	-37.5	28.2
Figure 5.b	-78.2	-37.5	94.5

Table 1: Comparisons of two local optima. (Note use of non-negative marginal likelihood and neg. model complexity (the higher it is the less complex the model).



```
for i = 1:length(length_scale)
    for j = 1:length(noise_standard_deviation)

        % Noise s. dev VS lengthscale
        hyp.cov = [length_scale(i) -0.1; ];
        hyp.lik = noise_standard_deviation(j);

        % Signal s. dev VS lengthscale
%         hyp.cov = [length_scale(i) signal_standard_deviation(j); ];
%         hyp.lik = -2.13;

        % get negative log marginal likelihood
        nlml = gp(hyp, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
        marg_lik_matrix(i,j) = nlml;
    end
end
```

Figure 6: Loop used to compute neg. log marginal likelihood over the grid of hyperparameters

3 Part C

Figure 7 shows a posterior predictive distribution of a GP with periodic covariance function (@covPeriodic), with optimal hyperparameters found to be: length scale $l = 1.10$, period $p = 1.00$ and $\sigma_{signal} = 3.82$.

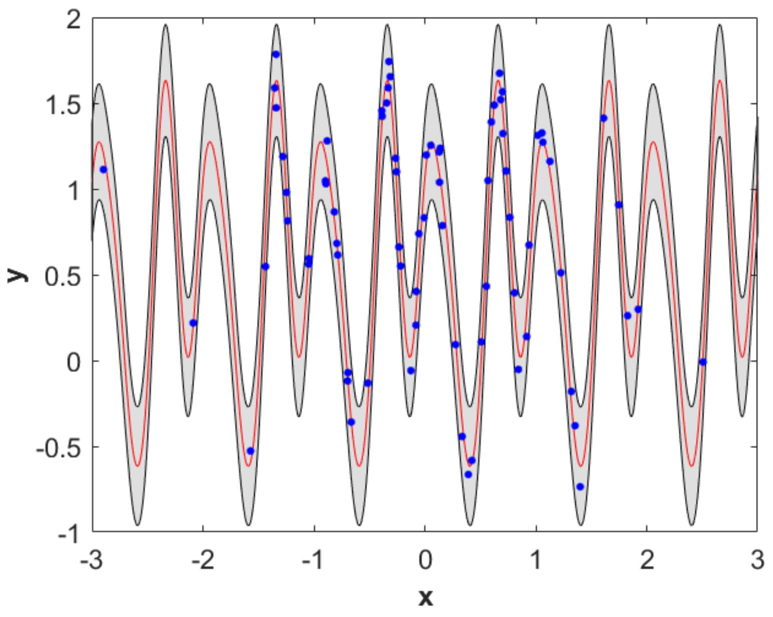


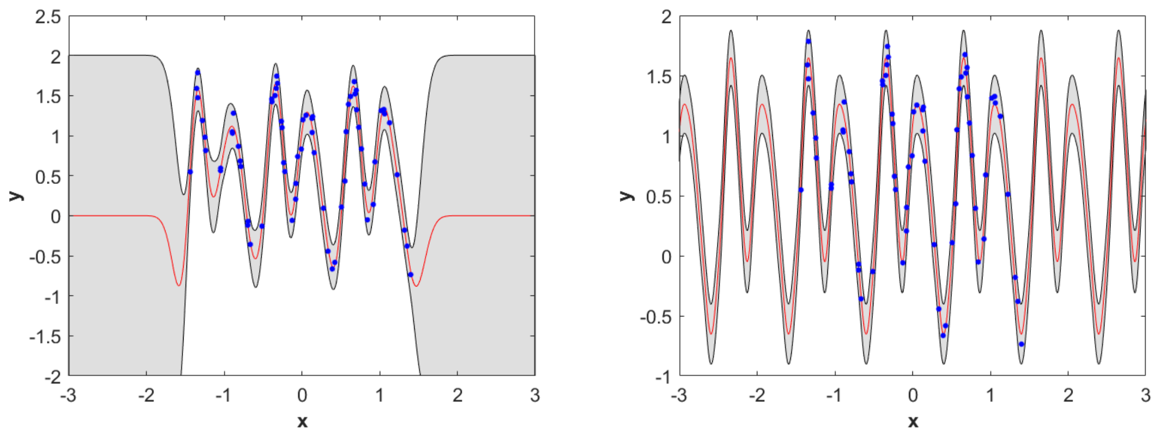
Figure 7: Gaussian process with a periodic covariance function (@covPeriodic), defined by length scale $l = 1.10$, period $p = 1.00$ and the signal standard deviation $\sigma_{signal} = 3.82$.

To analyse the difference between the two let's fit GPs to the interval $(-1.5, 1.5)$ where the majority of data lies (Figure 8). In the GP with SE covariance, due to short lengthscale $l = 0.13$, each point is correlated with points in its closest proximity. Such SE covariance has little ability to generalise away from the data, thus the 95% error-bars grow quickly away from the data-cluster. On the other hand, Periodic covariance translates the data points into a manifold where all points lie close to each other (a circular shape in a polar coordinate system). Periodic kernel is able to generalise well because it assumes continuity of the periodic trend.

The data is more likely to have been generated by a periodic process. GP with Periodic covariance has not only higher Log Marginal Likelihood, $lml = 22.3$, , but also a better data fit and higher Negative Model Complexity (so less complex), (see Table 2). The only argument against such conclusion, is higher $\sigma_{noise} = 0.15$, although it seems not so significant.

	Log Marginal Likelihood	Data Fit	Neg. Model Complexity
SE-ISO	-11.9	-37.5	100.4
Periodic	22.3	-16.4	135

Table 2: Model comparison between GP with SE-ISO and Periodic kernels (Note: non-negative log marginal likelihood).



(a) (b)

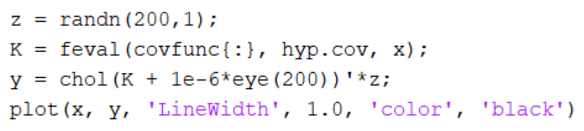
Figure 8: GP posterior function trained using data from $(-1.5, 1.5)$ interval only, where the data was more densely sampled.

4 Part D

Sampling from a prior distribution can be described in five steps (see Figure 9 for code):

1. Compute $K(\boldsymbol{X_*}, \boldsymbol{X_*})$ for n test points $\boldsymbol{X_*}$
2. Add small diagonal matrix ϵI to $K(\boldsymbol{X_*}, \boldsymbol{X_*})$
3. Compute Cholesky decomposition $K = LL^T$
4. Generate n random samples $\boldsymbol{u} \sim N(0, 1)$
5. Compute $\boldsymbol{y} = \boldsymbol{m} + \boldsymbol{Lu}$ (here $\boldsymbol{m} = 0$)

Small diagonal matrix ϵI is added for numerical reasons. It provides stability when computing Cholesky decomposition, by preventing eigenvalues of covariance matrix from decaying rapidly. By setting ϵ to a small value such as $1 * 10^{-6}$ the effect on samples is negligible [3].



```
z = randn(200,1);
K = feval(covfunc{:}, hyp.cov, x);
y = chol(K + 1e-6*eye(200))'*z;
plot(x, y, 'LineWidth', 1.0, 'color', 'black')
```

Figure 9: Sampling from a prior distribution.

The product of SE kernel and Periodic kernel is another kernel, sometimes referred to as **Locally periodic covariance** [4] (See Equation 1). It models functions with local periodicity which can evolve smoothly over time.

$$k_{LP}(x, x') = k_P(x, x') \times k_{SE}(x, x') = \sigma_P^2 \sigma_{SE}^2 \left(\frac{-2sin^2(\pi|x - x'|/p)}{l_P^2} \right) \times exp\left(\frac{-(x - x')^2}{2l_{SE}^2} \right) \tag{1}$$

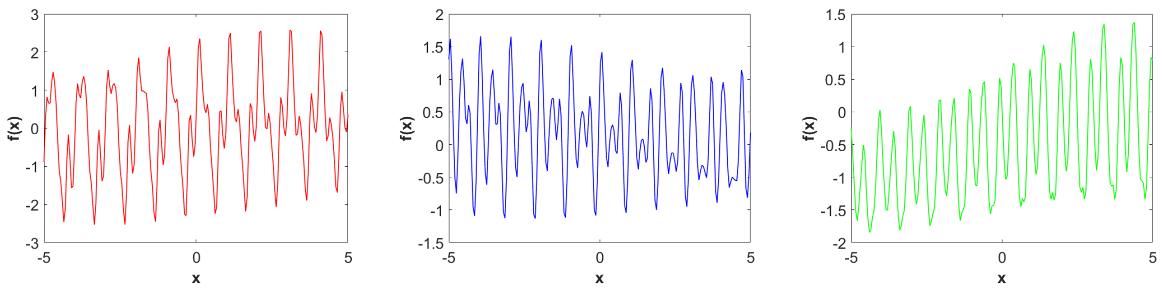


Figure 10: Three randomly sampled functions from a Locally periodic covariance function, woth following hyperparameters: $[l_P = e^{-0.5},\ p = 1,\ \sigma_P = 1,\ l_{SE} = e^2,\ \sigma_{SE} = 1]$

In a GP with LP kernel, periodic length scale l_P controls smoothness of a function within each period. GP in Figure 11.a has lower l_P and thus predictive distribution is smoother than in Figure 11.b. The SE length scale l_{SE} controls how smoothly function changes from period to period. For large values of l_{SE}, SE kernel tends to 1 and thus the variation from period to period diminishes, as shown in Figure 11.c.

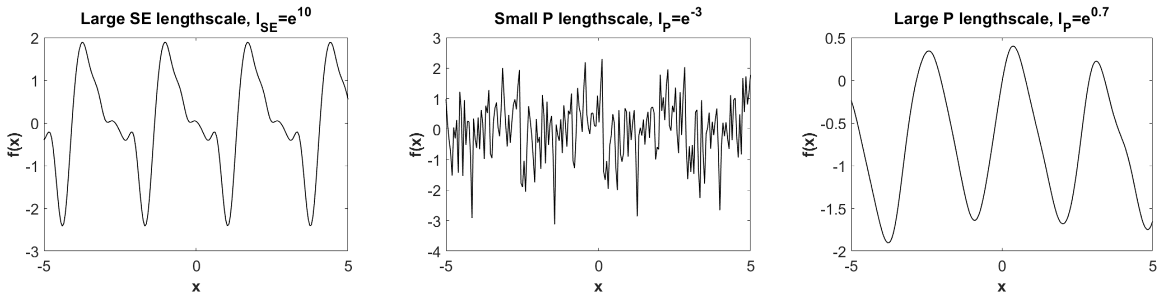


Figure 11: Functions sampled from a prior with period $p = e^1$, where the periodic and SE length scales were varied as defined by figure titles.

5 Part E

The SE-ARD kernel (Equation 2) is a product of two Squared exponential kernels over different dimensions. It chooses a separate lengthscale for each dimension to model their variability independently.

$$k_{SE-ARD}(x, x') = \sigma_s^2 exp\left(\sum_{d=1}^{D} \frac{-(x_d - x'_d)^2}{2l_d^2}\right) \tag{2}$$

SUM-SE-ARD (Equation 3) is a sum of two SE-ARD kernels, thus it has two lenghscales per dimension. It allows to capture slow and fast variations in the function. One kernel (with short lengthscale) can be responsible for high frequency changes while, another (with longer lengthscale) for representing function's global trend [4].

$$k_{SUM-SE-ARD}(x,x') = \sigma_{1,s}^2 exp\left(\sum_{d=1}^{D} \frac{-(x_{1,d}-x'_{1,d})^2}{2l_{1,d}^2}\right) + \sigma_{2,s}^2 exp\left(\sum_{d=1}^{D} \frac{-(x_{2,d}-x'_{2,d})^2}{2l_{2,d}^2}\right) \quad (3)$$

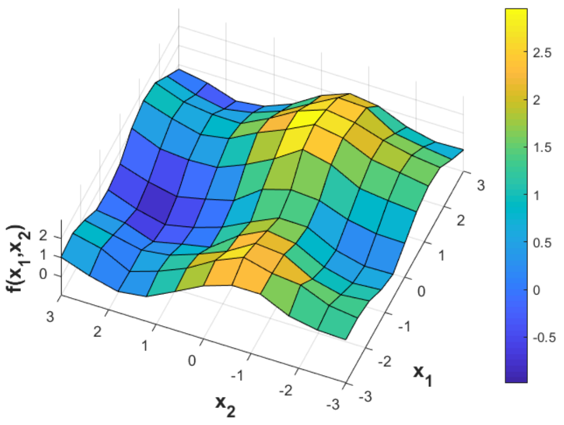
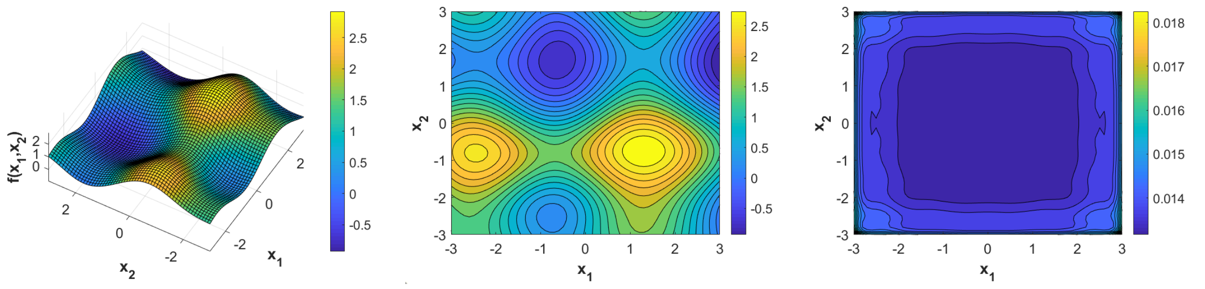


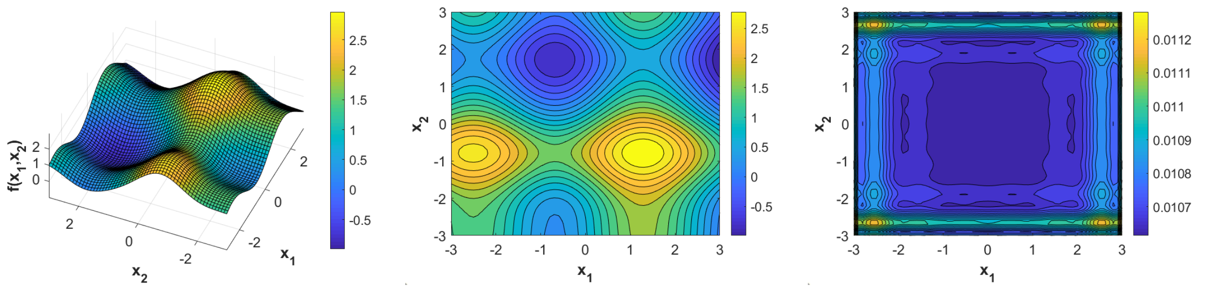
Figure 12: 2D data, evenly sampled on a grid, which was used in Part E of this coursework

Figures 13 and 14 show fits of GPs with SE-ARD and SUM-SE-ARD kernels respectively, fitted to data presented in Figure 15.



(a) Posterior pred. mean (b) Posterior pred. mean (c) Posterior pred. uncertainty

Figure 13: Posterior predictive mean and uncertainty for GP with SE-ARD kernel.



(a) Posterior pred. mean (b) Posterior pred. mean (c) Posterior pred. uncertainty

Figure 14: Posterior predictive mean and uncertainty for GP with SUM-SE-ARD kernel.

Using Equations 4 and 5 we decompose Log Marginal Likelihood of two GPS and present results in Table 3.

$$Log\ Marginal\ likelihood = (-data\ fit) - (model\ complexity) + (constant) \tag{4}$$

$$\log p(y|x, M_i) = \left(\frac{1}{2}y^T[K + \sigma_n^2 I]^{-1}y\right) + \left(\frac{1}{2}log|K + \sigma_n^2 I|\right) + \left(\frac{n}{2}log(2\pi)\right) \tag{5}$$

	Log Marginal Likelihood	Data Fit	Neg. Model Complexity
SE-ARD	19.2	-60.5	191
SUM-SE-ARD	66.3	-60.6	238

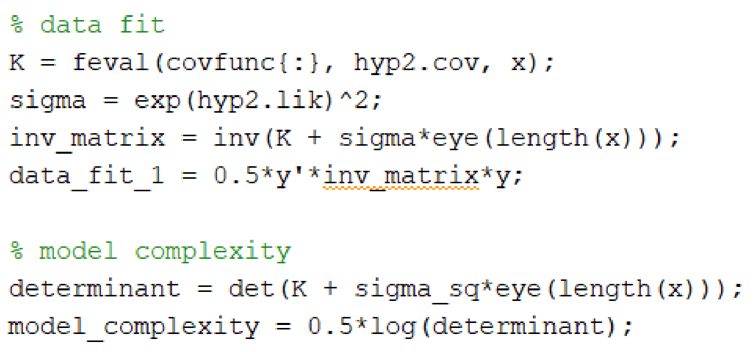
Table 3: Model fit comparison between SE-ARD and SUM-SE-ARD. (Note: non-negative Log Marginal Likelihood).

The data fit term is almost the same for both models, which means that both GPs assume similar posterior predictive distribution of the data about the mean. ($y^T[K + \sigma_n^2 I]^-1y$ - Sum of Euclidean distances of targets from zero mean is the same).

	σ_{noise}	$l_{x1}^{(1)}$	$l_{x2}^{(1)}$	$l_{x1}^{(2)}$	$l_{x2}^{(2)}$	$\sigma_{signal}^{(1)}$	$\sigma_{signal}^{(2)}$
SE-ARD	0.10	1.51	1.29	-	-	1.11	-
SUM-SE-ARD	0.10	1.44	556	447	0.98	1.11	0.7

Table 4: Optimised hyperparameters of two GP models.

Model complexity $|K + \sigma_n^2 I|$, takes low values for a matrices with a few dominant eigenvalues, because then the data is distributed along a few principal axis and the search space is restricted. GPs with low signal variance and large lengthscales (given the same noise variance, i.e. here $\sigma_{noise} = 0.1$) can generate fewer potential functions and thus have lower model complexity. SUM-SE-ARD kernel has two large lengthscales which drive the corresponding exponential terms to 1. Additionally, the ratio $\frac{(\sigma_{signal}^{(1)})^2}{(\sigma_{signal}^{(2)})^2}$ equals to 2.5 which reduces the function space of second SE-ARD kernel in SUM-SE-ARD. This makes GP with SUM-SE-ARD a less complex one and, by Occam's razor principle, it is preferable.



```
% data fit
K = feval(covfunc{:}, hyp2.cov, x);
sigma = exp(hyp2.lik)^2;
inv_matrix = inv(K + sigma*eye(length(x)));
data_fit_1 = 0.5*y'*inv_matrix*y;

% model complexity
determinant = det(K + sigma_sq*eye(length(x)));
model_complexity = 0.5*log(determinant);
```

Figure 15: Code for calculating Data fit and Model complexity.

References

[1] Llorente, Fernando, et al. "Marginal likelihood computation for model selection and hypothesis testing: an extensive review." arXiv preprint arXiv:2005.08334 (2020).

[2] The PyMC Development Team (n.d.). Bayes Factors and Marginal Likelihood — PyMC3 3.9.3 documentation. [online] docs.pymc.io. Available at: https://docs.pymc.io/notebooks/Bayes_factor.html [Accessed 31 Oct. 2020].

[3] Carl Edward Rasmussen and Williams, C.K.I. (2008). Gaussian processes for machine learning. Cambridge, Mass. Mit Press.

[4] Kristjanson Duvenaud, D. (2014). Automatic Model Construction with Gaussian Processes. [PhD Thesis] p.Chapter 2. Available at: https://www.cs.toronto.edu/ duvenaud/thesis.pdf [Accessed 2 Nov. 2020].