4F13 Probabalistic Machine Learning
Coursework 1: Gaussian Processes

Candidate Number: 5746G

October 31, 2020

Question a)

Listings 1: Training a Gaussian process with a squared exponential covariance function, `covSEiso`.

```
meanfunc = [];                    % empty: don't use a mean function
covfunc = @covSEiso;              % Squared Exponental covariance function
likfunc = @likGauss;              % Gaussian likelihood

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

hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x,
y);
nlml2 = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
[mu s2] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, xs);
```

Given a one-dimensional, non-parametric Gaussian process model with Gaussian likelihood and Gaussian process prior with zero mean and *squared exponential* covariance function k, the *prior variance* on the noisy targets y is:

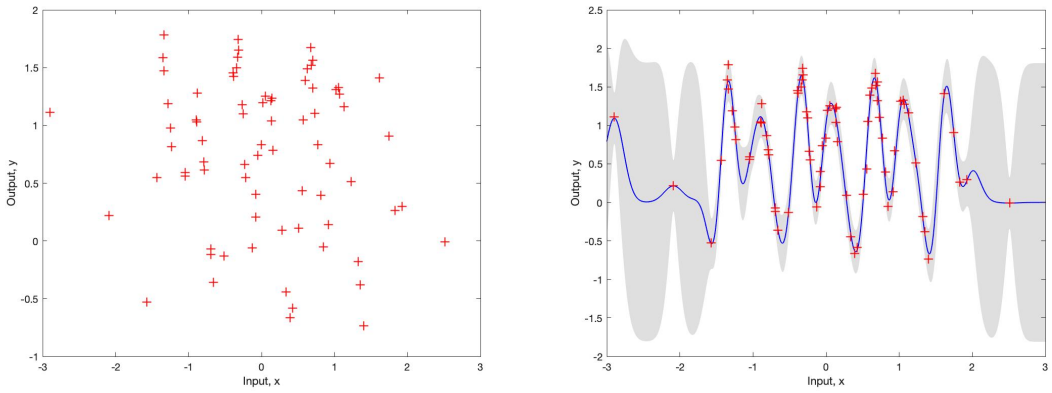
$$k_y\left(x_i, x_j\right) = k\left(x_i, x_j\right) + \delta_{ij}\sigma_n^2 \tag{1}$$

$$= \sigma_f^2 \exp\left(-\frac{\left(x_i - x_j\right)^2}{2\ell^2}\right) + \delta_{ij}\sigma_n^2 \tag{2}$$

where δ_{ij} is a Kronecker delta, which is one iff $i = j$ and zero otherwise. The model is trained to find the length-scale ℓ, the signal variance σ_f^2 and the noise variance σ_n^2 (collectively referred to as the *hyperparemetrs*) which minimise the negative log marginal likelihood $-\log p\left(\mathbf{y} \mid X\right)$ for the dataset shown in Fig. 1(a).

Table 1: Hyperparameter and marginal likelihood values prior to and post minimising the negative log marginal likelihood.

	ℓ	σ_f	σ_n	$p\left(\mathbf{y} \mid X\right)$
Initial value	0.368	1.00	1.00	4.46×10^{-41}
Optimised value	0.128	0.897	0.118	6.78×10^{-6}



(a) **(b)**

Figure 1: Panel (a) plots the data set `cw1a.mat`, which consists of 75 observations of scalar input and output pairs. Panel (b) plots the pointwise mean of the Gaussian process prediction in blue. The shaded area represents the pointwise mean plus and minus two times the standard deviation (corresponding to the 95% confidence region).

Under the model, the variance of the predictive distribution of a test target y_* takes the form:

$$\mathbb{V}[y_*] = k(x_*, x_*) + \sigma_n^2 - k(x_*, X)\left(K(X,X) + \sigma_n^2 I\right)^{-1} k(X, x_*) \tag{3}$$

As a property of the Gaussian distribution, the variance depends solely on the inputs, and not the observed targets. The first two terms comprise the prior variance $k_y(x_*, x_*)$, from which is subtracted a positive term which represents information the observations yield about the function. The prior variance is independent of x_*, giving an upper-bound on the variance:

$$\mathbb{V}[y_*] < k(x_*, x_*) + \sigma_n^2 \tag{4}$$

$$= \sigma_f^2 + \sigma_n^2. \tag{5}$$

Since $k(x_*, X) > 0$, this upper bound cannot be attained. The maximum width of the 95% predictive error bars is calculated using this limit and values for the optimised hyperparameters given in Table 1:

$$4\sqrt{\mathbb{V}[x_*]} < 4\sqrt{\sigma_f^2 + \sigma_n^2} = 3.62 \tag{6}$$

This is in good agreement with the plot of the Guassian process prediction shown in Fig. 1(b); the error bars reflect the prior standard deviation of the process for input values that are distant from any training points. As the test point moves closer to the data, the last term in the expression for the variance increases, resulting in a decrease in variance. The plot shows a corresponding contraction in the size of the error bars.

The small optimised length-scale ℓ allows potential functions to vary substantially, giving rise to rapid variations which are seen as 'wiggly' behaviour.

Question b)

Listings 2: Training a Gaussian process with a squared exponential covariance function, `covSEiso`. As Listings 1, with line 5 modified as shown.

```
5  hyp = struct('mean', [], 'cov', [1 0], 'lik', 0);
```

Table 2: Hyperparameter and marginal likelihood values prior to and post minimising the negative log marginal likelihood.

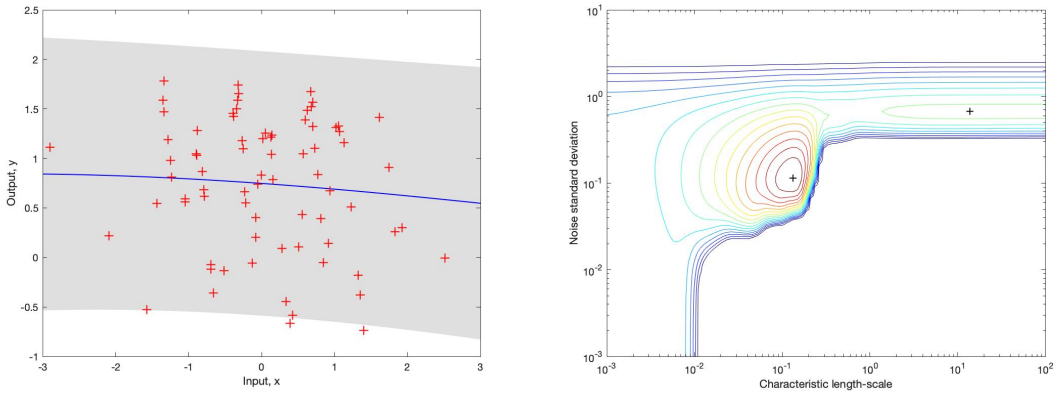
	ℓ	σ_f	σ_n	$p\left(\mathbf{y} \mid X\right)$
Initial value	2.72	1.00	1.00	3.03×10^{-39}
Optimised value	8.35	0.699	0.663	1.07×10^{-34}

Listings 3: Contour plot of log marginal likelihood. The signal variance is set to $\sigma_f^2 = 1$.

```
meanfunc = [];
covfunc = @covSEiso;
likfunc = @likGauss;

hyp = struct('mean', [], 'cov', [log(l(i)) 0], 'lik', log(sigma_f(j)));

[nlZ, dnlZ] = gp(hyp, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
lZ(j,i) = - nlZ;
```



(a) **(b)**

Figure 2: Panel (a) plots the Gaussian process posterior for the optimal hyperparameter configuration in Table 2. Panel (b) is a contour plot showing the log marginal likelihood as a function of characteristic length-scale and the noise level, with the signal variance set to $\sigma_f^2 = 1$. The contour lines are spaced 10 units apart in log probability density, and the two local optima marked by '+'.

The Gaussian predictive distribution shown in Fig. 2(a) has a long length-scale ($\ell = 8.35$). Potential functions cannot vary fast enough to explain large variations in outputs for nearby inputs. Hence, the model explains all the observations as noise, no longer requiring the signal covariance. This second model is disfavoured by the marginal likelihood: it is a factor of $\frac{1.07\times10^{-34}}{6.78\times10^{-6}} = 1.58 \times 10^{-29}$ times smaller than the marginal likelihood for the optimised model from a).

Fig. 2(b) shows the variation in log marginal likelihood as a function of the characteristic length-scale and the noise standard deviation for the squared exponential covariance function given in eq. (2). The signal variance σ_f^2 was set to 1. The lower-left optimum corresponds to the global optimum, where the free parameters maximise the log marginal likelihood given the fixed signal variance. The second local optimum occurs for a longer length-scale and at a noise level of $\sigma_n^2 = 4.71$. In this region, the marginal likelihood becomes almost independent of the length-scale, another instance of the model explaining everything as noise.

On the other hand, for small noise and short length-scales in the order of 0.01, the marginal likelihood becomes almost independent of the noise variance. In this case, the functions vary rapidly enough to explain all the data points. Despite interpolating all the data exactly, there is less support for these models; the marginal likelihoods are lower than at the global optimum.

Question c)

Listings 4: Training a Gaussian process with a periodic covariance function, `covPeriodic`. As Listings 1, with lines 2 and 5 modified as shown.

```
2  covfunc = @covPeriodic;            % periodic covariance function

5  hyp = struct('mean', [], 'cov', [0 0 0], 'lik', 0);
```

For a Gaussian process with *periodic* covariance function, the error bars are constrained to be periodic. Consequently, both the mean function and error bars in Fig. 3 are similar over all observations. Error bars which remain thin across the entire domain imply that the correlation with the training data is high for any given test input x_*. This is in contrast to the error bars produced by the squared exponential covariance function shown in Fig. 1(b), which inflate where data is more sparse.

This model is favoured strongly by the marginal likelihood: it is a factor of $\frac{2.04\times10^{15}}{6.78\times10-6} = 3.00 \times 10^{20}$ times greater than that for the model from a), conclusive evidence that the data generating mechanism is strictly periodic.

Table 3: Hyperparameter and marginal likelihood values prior to and post minimising the negative log marginal likelihood for a periodic covariance function.

	ℓ	p	σ_f	σ_n	$p\left(\mathbf{y} \mid X\right)$
Initial value	1.00	1.00	1.00	1.00	2.84×10^{-35}
Optimised value	1.04	0.999	1.24	0.109	2.04×10^{15}

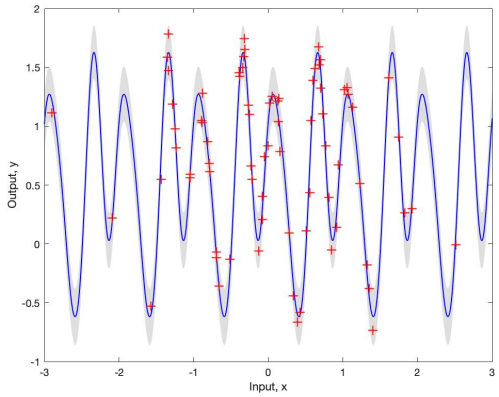


Figure 3: Gaussian process posterior for a periodic covariance function under the optimal hyper-parameter configuration in Table 3.

Question d)

Listings 5: Generation of noise free function from Gaussian processes with a product of periodic and squared exponential covariance function.

```
1  x = linspace(-5,5,200)';          % points to evaluate functions
2  covfunc = {@covProd, {@covPeriodic, @covSEiso}};    % product of periodic
   and SE covariance function
3  hyp.cov = [-0.5 0 0 2 0];
4  [K, dK] = feval(covfunc{:}, hyp.cov, x);
5  y = chol(K + 1e-6*eye(200))' * gpml_randn(seed, 200, 1); % initialising with
   the same seed generates the same pseudo-random numbers every time
```

The Cholesky decomposition in line 5 of Listings 5 factorises the covariance matrix into a product of a lower triangular matrix and its transpose. This decomposition can only be performed on strictly positive definite matrices. Hence, a small diagonal matrix is added to the covariance matrix to ensure it satisfies this constraint.

The covariance function for these processes are constructed as the product of periodic and squared exponential covariance functions:

$$\begin{aligned} k(x_i, x_j) &= k_p(x_i, x_j) \times k_{SE}(x_i, x_j) \\ &= \sigma_{f1}^2 \exp\left(-\frac{2}{\ell_1^2}\sin^2\left(\frac{\pi(x_i - x_j)}{p}\right)\right) \times \sigma_{f2}^2 \exp\left(-\frac{(x_i - x_j)^2}{2\ell_2^2}\right). \end{aligned} \tag{7}$$

The sampled periodic functions in Fig. 4 are defined for a period $p = 1$, resulting in 10 complete cycles over the input range $x \in [-5, 5]$. Multiplying the periodic and squared exponential covariance functions has no effect on the argument to the exponentiated sinusoid term. Thus, there is underlying oscillatory behaviour in the product covariance function $k(x_i, x_j)$. Since the amplitude of these oscillations is scaled by the size of the squared exponential covariance function, there is a degree of amplification/attenuation, which varies with x_i and x_j. The rate of increase/decrease in amplitude is controlled by the characteristic length-scale of the squared exponential; the large value of $\ell_2 = 7.39$ yields gradual changes in the variation of amplitude.

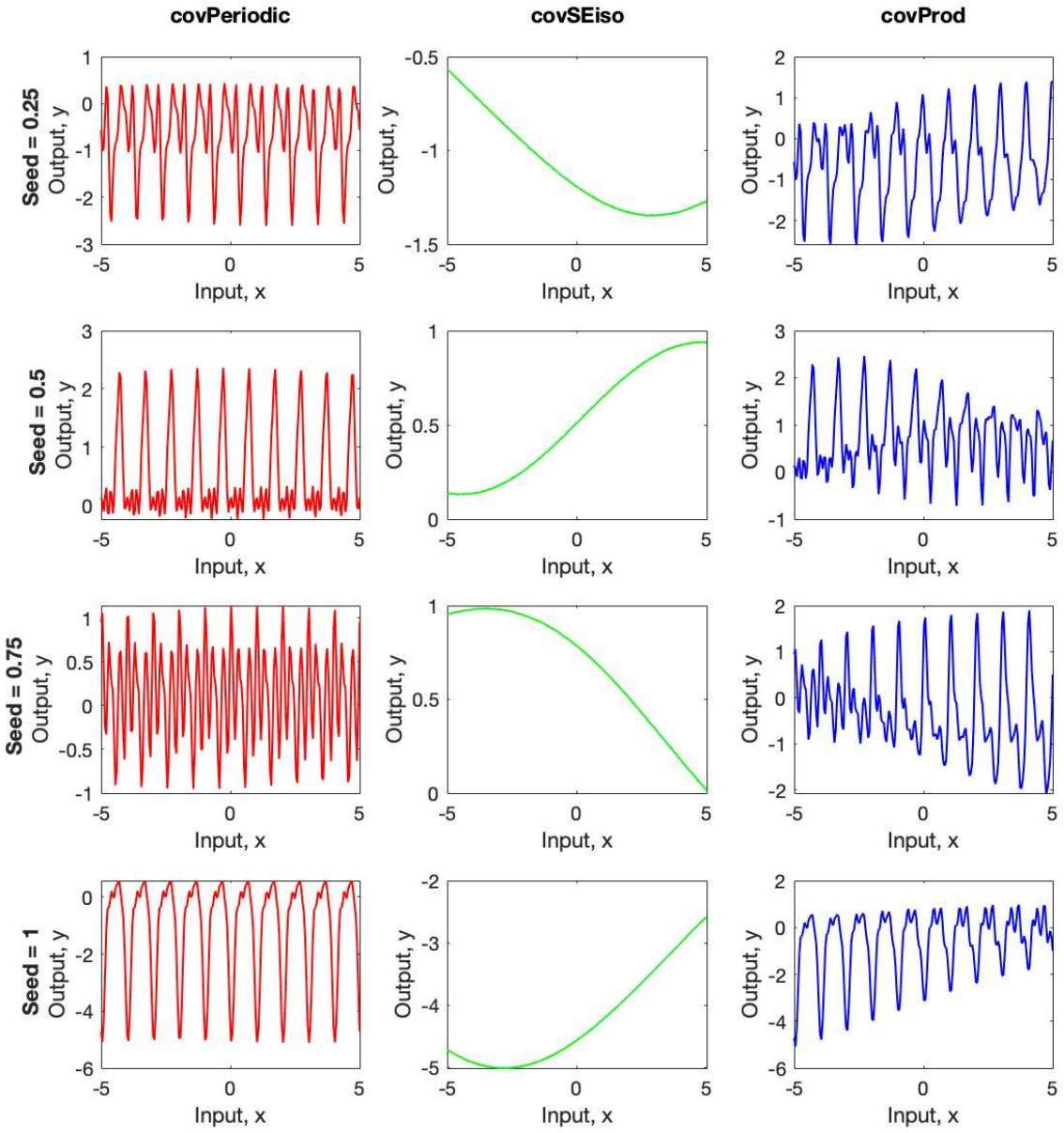


Figure 4: Random noise free functions sampled from Gaussian processes with different forms of covariance function.

Question e)

Listings 6: Training a Gaussian process with a single ARD covariance function, `covSEard`.

```
1  meanfunc = [];
2  covfunc = @covSEiso;              % Squared Exponential covariance function
   with Automatic Relevance Detemination (ARD) distance measure
3  likfunc = @likGauss;
4
5  hyp.cov = 0.1 * randn(3,1);
6  hyp = struct('mean', [], 'cov', hyp.cov, 'lik', 0);
7  nlml = gp(hyp, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
8
9  hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x,
   y);
10 nlml2 = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
11 [mu s2] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, Xs);
```

Listings 7: Training a Gaussian process with a the sum of two ARD covariance functions. As Listings 6, with lines 2 and 5 modified as shown.

```
2  covfunc = {@covSum; {@covSEard, @covSEard}};     % Sum of two SE ARD
   covariance functions

5  hyp.cov = 0.1 * randn(6,1); functions
```

A single *automatic relevance determination* (ARD) covariance function in two dimensions takes the form:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_{f1}^2 \exp\left(-\sum_{d=1}^{2} \frac{\left(x_d - x_d'\right)^2}{2\ell_{1,d}^2}\right). \tag{8}$$

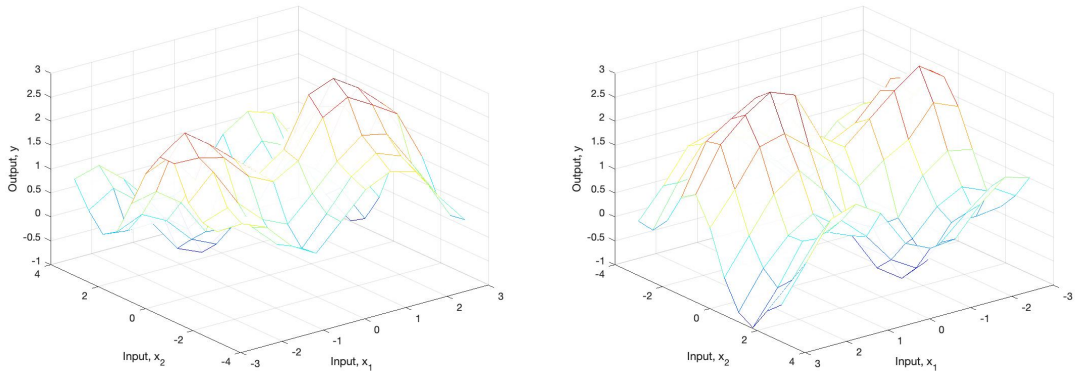
The predictive distribution for a Gaussian process model with such prior is trained on the data set `cw1e.mat`, shown in Fig. 5, for which the predictive distribution is plot in Fig. 6. The characteristic length-scale is similar along the two input axes; the function varies comparably with x_1 and x_2. It is observed that mean prediction for an input far from from the training points in either dimension is close to zero; if either of the terms $\left(x_d - x_d'\right)$ for $d = 1, 2$ is large, the exponential of the sum of the terms will be small. The result is that a single ARD covariance function poorly leverages correlations in two dimensions.

A second Gaussian process is trained on the same data set, but with the sum of two ARD covariance functions:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_{f1}^2 \exp\left(-\sum_{d=1}^{2} \frac{\left(x_d - x_d'\right)^2}{2\ell_{1,d}^2}\right) + \sigma_{f2}^2 \exp\left(-\sum_{d=1}^{2} \frac{\left(x_d - x_d'\right)^2}{2\ell_{2,d}^2}\right). \tag{9}$$

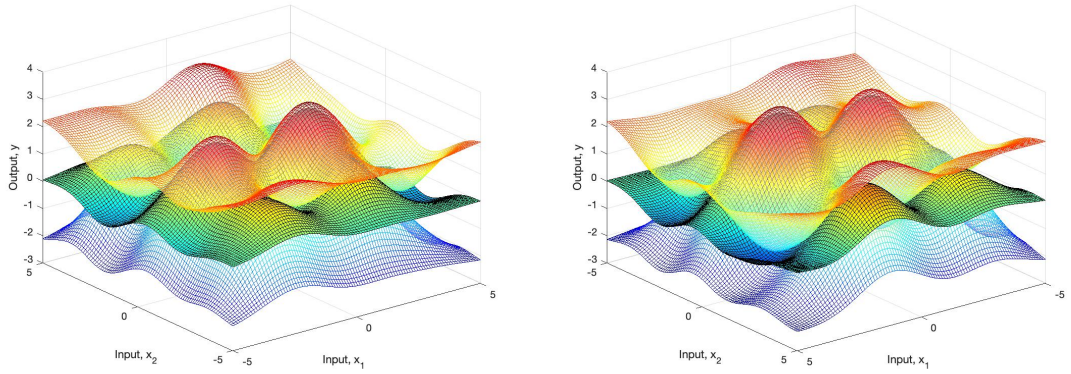
Table 4: Hyperparameter and marginal likelihood values prior to and post minimising the negative log marginal likelihood for a single ARD prior covariance function.

	$\ell_{1,1}$	$\ell_{1,2}$	$\sigma_{f,1}$	σ_n	$p\left(\mathbf{y} \mid X\right)$
Initial value	0.751	0.920	0.907	1.00	7.98×10^{-64}
Optimised value	1.51	1.29	1.11	0.103	2.22×10^{8}



(a) **(b)**

Figure 5: Panels (a) and (b) show visuals of the data set `cw1e.mat`, which consists of 121 observations of two dimensional input and scalar output pairs.

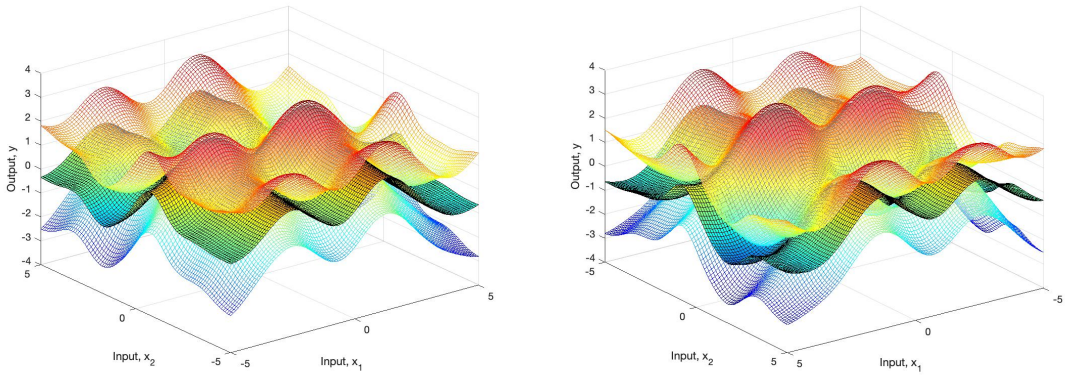


(a) **(b)**

Figure 6: Pointwise mean (surface) and 95% confidence region (mesh) for a Gaussian process prediction with a single ARD covariance function, trained on the data set `cw1e.mat`.

Table 5: Hyperparameter and marginal likelihood values prior to and post minimising the negative log marginal likelihood for the sum of two ARD covariance functions.

	$\ell_{1,1}$	$\ell_{1,2}$	$\sigma_{f,1}$	$\ell_{1,1}$	$\ell_{1,2}$	$\sigma_{f,1}$	σ_n	$p\left(\mathbf{y} \mid X\right)$
Initial value	0.997	0.923	1.18	0.987	0.931	1.14	1.00	4.83×10^{-68}
Optimised value	1.44	1.92×10^{3}	1.11	1.28×10^{3}	0.985	0.710	0.0978	6.87×10^{28}



(a) **(b)**

Figure 7: Pointwise mean (surface) and 95% confidence region (mesh) for a Gaussian process prediction with a sum of two ARD covariance functions, trained on the data set `cw1e.mat`.

Fig. 7 plots the predictive distribution for this more complex choice of prior covariance. By breaking the symmetry in the initialised hyperparametrs, one ARD function is trained to model the covariance distribution along each axis. This fact is highlighted in Table 4; the characteristic length-scale is different along the two input axes for each function. The first ARD function varies rapidly as a function of x_2, but less rapidly as a function of x_1. The opposite is true of the second function. Better able to explain the covariance of the observations, this model achieves improved fit to the training data compared to use of a single ARD function.

The model is favoured strongly by the marginal likelihood; it is a factor of $\frac{6.87\times10^{28}}{2.22\times10^{8}} = 3.09 \times 10^{20}$ times larger than that of the simpler single ARD model, revealing that the increase in model complexity is offset by the improvement in data-fit.

Word Count (excluding Listings): 994

References

Rasmussen, C. E. & Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, The MIT Press.