CUED - Engineering Tripos Part IIB 2019-2020 **Module Coursework**

Module	4F13	Title of report	Gaussian Processes, Coursework 1
Date submitted: 8/11/19		Assessment for this module is ☒ 100% / ☐ 25% coursework of which this assignment forms _____ %	

UNDERGRADUATE STUDENTS ONLY		POST GRADUATE STUDENTS ONLY			
Candidate number:	5584F	Name:		College:	

Feedback to the student ☐ **See also comments in the text**		Very good	**Good**	Needs improvmt
C O N T E N T	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
P R E S E N T A T I O N	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Overall assessment (circle grade)		A*	**A**	B	C	D
Guideline standard		>75%	**65-75%**	55-65%	40-55%	<40%
Penalty for lateness:		*20% of marks per week or part week that the work is late.*				

Marker: Date:

4F13 - Coursework One - Gaussian Processes

Candidate Number: 5584F

Word count: 998

November 8, 2019

I. A

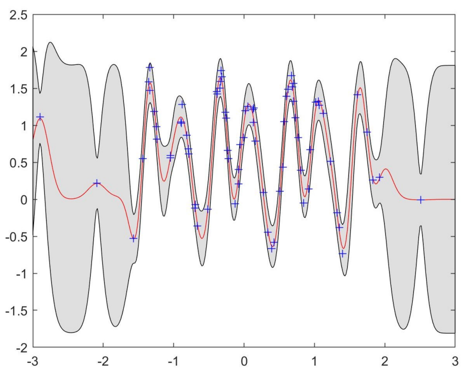


Figure 1: *GP with squared exponential covariance trained on data a with log hyper parameters initialised at [-1, 0, 0]*

Figure 1 shows the result of training a GP with squared exponential covariance function with log hyper parameters initialised at [-1, 0, 0]. After minimisation of negative log marginal likelihood (-log[p(y|x,$\mathcal{M}$)]) the hyper parameters ($\boldsymbol{\theta}$) were [0.1282 ,0.897, 0.1178] which corresponds to characteristic length scale, signal σ and noise σ respectively. The optimised log[p(y|x,$\mathcal{M}$)] was -11.9. The hyper parameters are observed in figure 1 as the length scale representing feature size in the x direction, signal σ the feature amplitude, and noise σ the data variation about the mean. The 95% error bars are very thin where data is observed corresponding to a high likelihood of the function existing here, at the extremes data is more sparse so the uncertainty in function shape is greater.

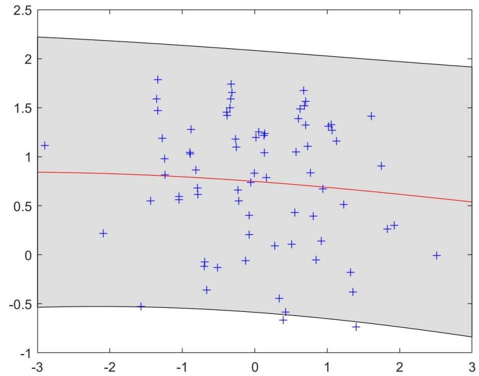
All essential code listing can be seen in the appendix.

II. B

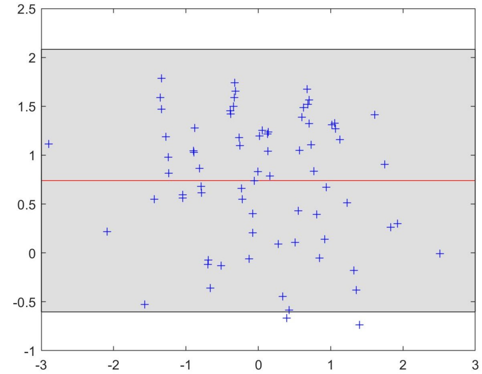
Additional local minima in negative log marginal likelihood can be located from different initialisations. Figure 2 shows four of these local minima, and their respective log[p(y|x,$\mathcal{M}$). Figure 2a shows a fit with length scale 8 which is more then the observed range. This causes large uncertainties and is only able to only capture the downward trend. Figure 2b extends this length scale even further to the point where the model believes a constant but very noisy function is correct. Figure 2c shows a model of very small length scale, the mean passes through the data very well but its very noisy between data points. Figure 2d shows a similar model to the initial one, but with smaller signal σ and therefore smaller confidence intervals.
By looking at the log marginal likelihood you can assess how good a fit the model is, the best two being figures 1 and 2d. Both these appear to fit the data equally well without overfitting, but due to the still significantly higher log[p(y|x,$\mathcal{M}$)] of figure 1 this model is likely to be the best fit.

III. C

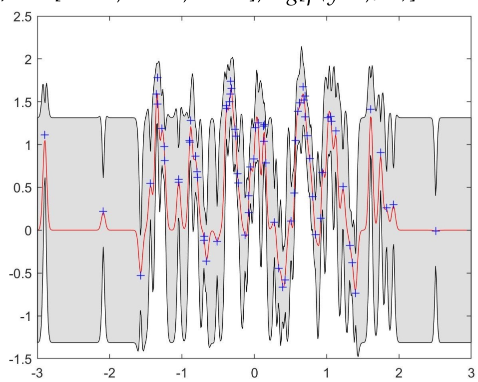
The same data was modelled using a periodic covariance function, this requires an additional hyper parameter (θ[2]) for the period. Figures 3 and 4 show these models. By using a peri-



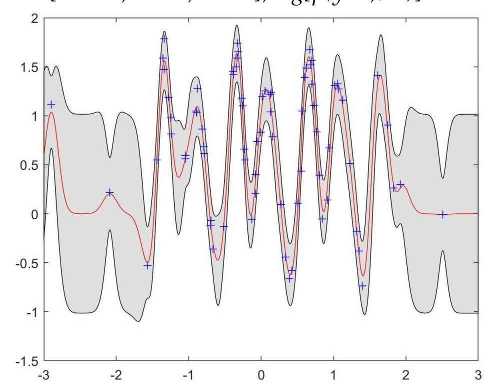
(a) *θ = [8.042, 0.696, 0.663], log[p(y | x,$\mathcal{M}$)] = -78.22*



(b) *θ = [3.3e10, 0.744, 0.667], log[p(y | x,$\mathcal{M}$)] = -78.34*



(c) *θ = [0.0263, 0.641, 0.142], log[p(y | x,$\mathcal{M}$)] = -57.56*



(d) *θ = [0.097, 0.49, 0.132], log[p(y | x,$\mathcal{M}$)] = -23.46*

Figure 2: *The effect of different hyper parameters*

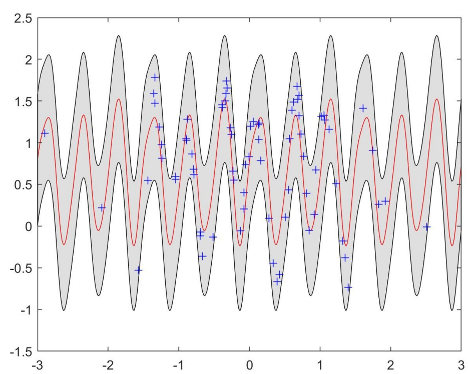


Figure 3: *Periodic covariance with θ = [0.49, 1.5, 1.17, 0.356], log[p(y | x,$\mathcal{M}$)] = -48.1*

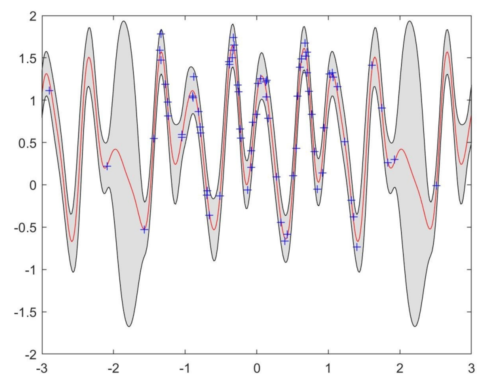
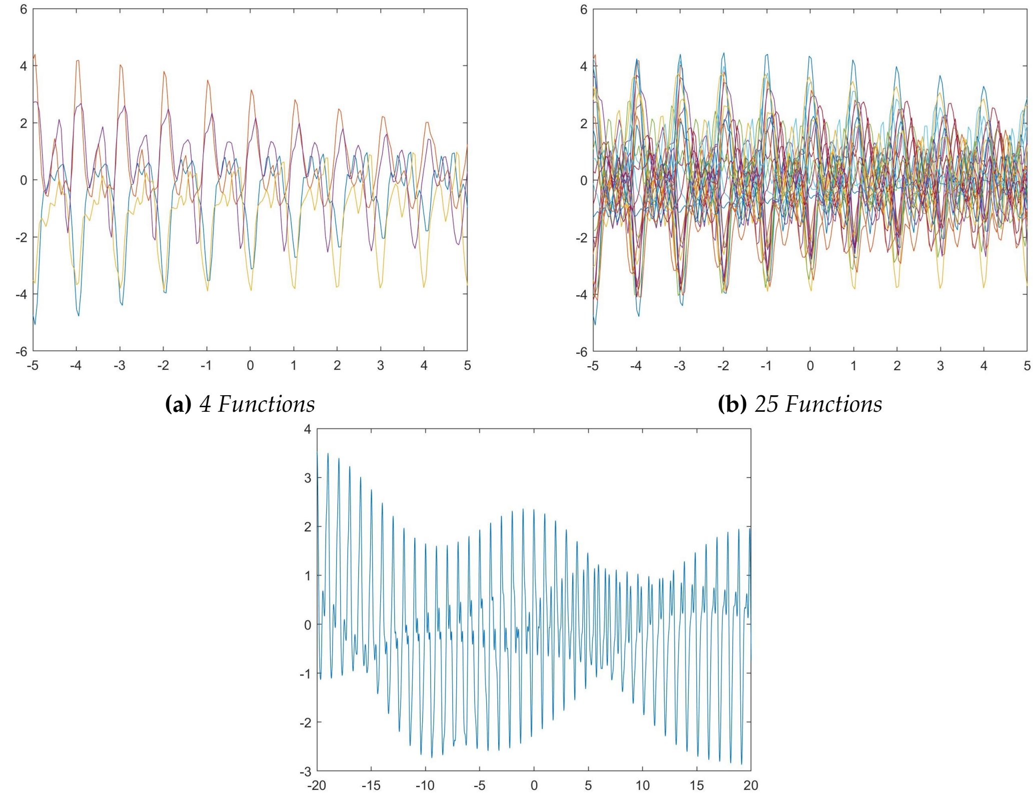


Figure 4: *Periodic covariance with θ = [0.205, 4.0, 0.93, 0.116], log[p(y | x,$\mathcal{M}$)] = -6.7*

odic covariance function the model gains prior information. This means that in regions of low data around x=±2.5 the model still shows small error bars compared to using a squared exponential covariance as before. Increasing signal σ from figure 3 to 4 reduces the error bars, models the peaks better and increases log[p(y | x,$\mathcal{M}$)] to above that of the squared exponential models. This higher likelihood and small predictive error bars at extremes makes



(a) *4 Functions*

(b) *25 Functions*

(c) *Wider range*

Figure 5: *Various random functions*

me believe the data was generated periodically. However, shown in figure 2a there is a general negative trend in the data which isn't periodic, meaning the generation could be a combination of periodic and linear terms.

IV. D

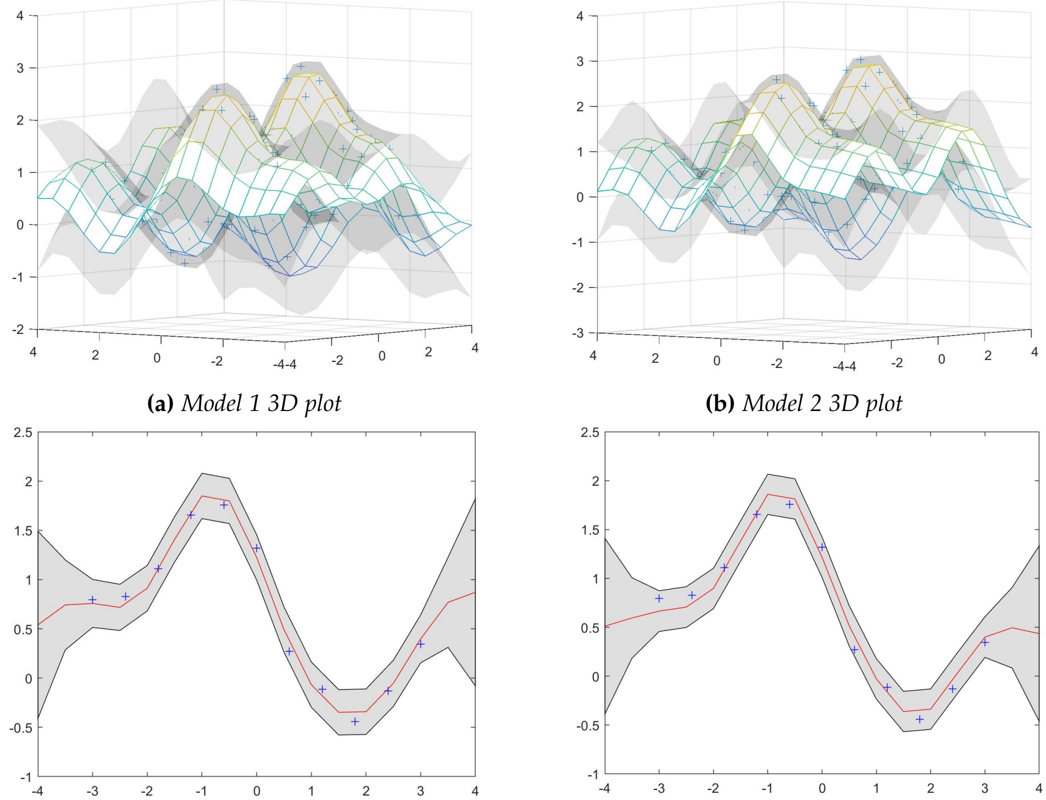
Random functions can be generated using desired covariance functions, figure 5 shows some sample functions using the covariance function in equation 1, and hyper parameters $[L_p, p, S_{\sigma p}, L_s, S_{\sigma s}]$ = [0.606, 1, 1, 7.39, 1]. When generating a small diagonal matrix must be added to the covariance matrix, this is to ensure the matrix is positive semi-definite. Rounding errors can cause small negative eigenvalues which this addition corrects. Positive semi-definite covariance matrices ensure the probability distribution assigns positive mass to all values and therefore is valid.

$$k(x,z) = S_{\sigma p}^2 S_{\sigma s}^2 exp\left(\frac{-2Sin^2 \frac{\pi(x-z)}{p}}{L_p^2} - \frac{(x-z)^2}{2L_s^2}\right) \tag{1}$$

The parameters can be observed in the functions in figure 5. The period of 1 is the distance between the large peaks. The characteristic length scales can be seen in figure 5c, 7.39 is the squared exponential scale and determines envelope frequency, and 0.606 is the small variation between the main periods. The signal σ of both parts is 1 and is the size of these variations.

V. E

Two dimensional data was loaded and modelled using two models of varying complexity.



(a) *Model 1 3D plot* **(b)** *Model 2 3D plot*

(c) *Model 1 slice through x(1) = 0* **(d)** *Model 2 slice through x(1) = 0*

Figure 6: *Visualisations of two models on the 2D data*

The covariance functions are shown in equations 2 and 3. The resulting log[p(y | $\mathbf{x}$,$\mathcal{M}$)], and sum of uncertainty is listed in table 1. Visually both models fit the data well, however the additional term is shown to increase the likelihood significantly meaning the extra complexity is worthwhile. The additional complexity also doesn't run into the problems of over fitting as marginally likelihood accounts for this. Additional models with more terms in the covariance function were tested, but these showed no further improvements.

$$k_1(x,z) = S^2_{\sigma s} exp(-\frac{(x-z)^T \begin{bmatrix} L_1^2 0 \\ 0 L_2^2 \end{bmatrix}^{-1} (x-z)}{2}) \tag{2}$$

$$k_2(x,z) = k_1(x,z) + k_1(x,z) \tag{3}$$

Model	log[p(y | $\mathbf{x}$,$\mathcal{M}$)]	$\Sigma_1^D \sigma_y^2$
One	19.219	20.18
Two	66.408	14.9
Three	66.404	14.89
Four	66.403	14.9
Five	66.40	14.87
Six	66.40	14.94

Table 1: *Model Performance*

VI. Appendix - Code listings

i. a,b,c

```
meanfunc = [];
covfunc = @covSEiso;
likfunc = @likGauss;
hyp = struct('mean', [], 'cov', [-1 0],
    'lik', 0);
```

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

ii. d

```
 x= linspace(-5,5,200)';
 K = feval(covfunc{:}, hyp.cov, x);
  for i = 1:25
    x2 = gpml_randn(2, 200, 1);
    y = chol(K+1e-6*eye(200))'*x2 ;
    plot(x, y)
    hold on
  end
```

iii. e

```
%3D plotting
figure(2);
[mu2 s2] = gp(hyp2min, @infGaussLik,
    ↪ meanfunc, covfunc2, likfunc, x,
    ↪ y, xs);
mesh(reshape(xs(:,1),17,17),reshape(xs
    ↪ (:,2),17,17),reshape(mu2,17,17))
    ↪ ;
hold on;
scatter3(x(:,1),x(:,2), y, '+');
hold on;
surf(reshape(xs(:,1),17,17),reshape(xs
    ↪ (:,2),17,17),reshape(mu2+2*sqrt(
    ↪ s2),17,17),'FaceAlpha','0.1','
    ↪ EdgeColor' , 'none','FaceColor',
    ↪ 'k');
hold on;
surf(reshape(xs(:,1),17,17),reshape(xs
    ↪ (:,2),17,17),reshape(mu2-2*sqrt(
    ↪ s2),17,17),'FaceAlpha','0.1','
    ↪ EdgeColor' ,'none','FaceColor','
    ↪ k');
```