4F13 Probabilistic Machine Learning
Coursework #1: Gaussian Process

Candidate Number: 5504C
Word count: 972

November 1, 2023

1 Exercise a)

After unconstrained optimization with initial hyper-parameter values of $[-1, 0, 0]$ (characteristic length scale, signal noise, data noise), we obtained optimized hyper-parameters of $[0.128, 0.897, 0.118]$ and $log(p(y|x, M_i))$ as -11.9. The variance for predictive distribution of GP is:

$$\mathrm{k}\left(x', x'\right) + \sigma^2_{\text{noise}} \; - k\left(x', x\right)^\top \left[\mathrm{K} + \sigma^2_{\text{noise}}\,\mathrm{I}\right]^{-1} \mathrm{k}\left(x', x\right)$$

The error bars represent a 95% confidence interval, derived from the predictive variance for each point x' using the given formula, which adjusts for new data points. The predictive error bars shown in Figure.1 are narrower in areas where there are many observed data points, indicating higher confidence. Conversely, due to the short length scale $l = 0.128$, uncertainty increases for observations further apart, indicating that the function correlates over short distances, leading to swift oscillations. The signal variance of 0.897 indicates a notable variation in the function, aligning with the observed oscillating amplitude. Meanwhile, a data noise of 0.118 adds a moderate noise level to the observations, affecting prediction confidence but reducing the likelihood of overfitting.

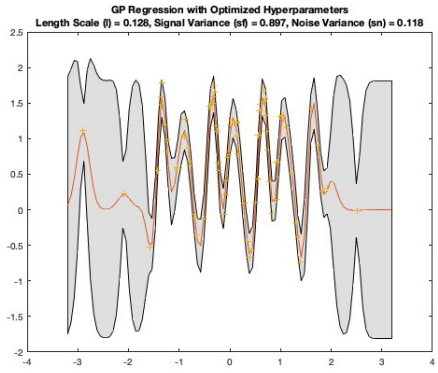


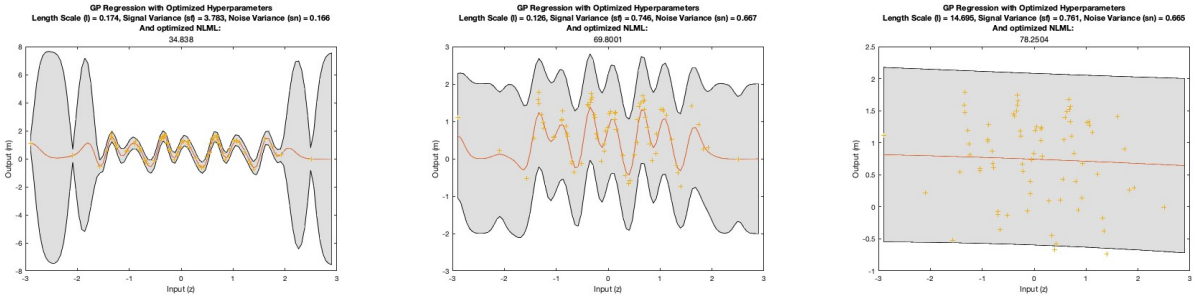
Figure 1: GP process with optimized hyper-parameters

2 Exercise b)

Different optimization convergence with different hyper-parameters initialization is shown in Figure.2a, Figure.2b and Figure.2c. The large **length scale** in the GP model of 2c suggests minimal data variation due to high correlations between distant data points, resulting in an almost flat prediction, unlike 2a

and 2b which better capture data oscillations. 2a with higher **signal variance** more effectively represents the data's oscillatory nature than the more subdued 2b. **Noise variance**, indicating observation noise, is higher in 2b and 2c compared to 2a, as shown by 2b's wider confidence bands around the data points.

The larger the log marginal likelihood, the better the fit. This indicates that 2a is the most appropriate model among the three, which balances complexity and fit well. It captures the oscillatory nature of the data without being overly reactive to individual data points, suggesting a good trade-off between bias and variance. Based on the above observations, it is confident that 2a provides the best fit among the presented options.



(a) $\theta_{opt} = [0.174, 3.783, 0.166]$ and log marginal likelihood = -34.838

(b) $\theta_{opt} = [0.126, 0.746, 0.667]$ and log marginal likelihood = -69.800

(c) $\theta_{opt} = [14.695, 0.761, 0.665]$ and log maginal likelihood = -78.250

Figure 2: Effects of changing the range of hyper-parameters initialization

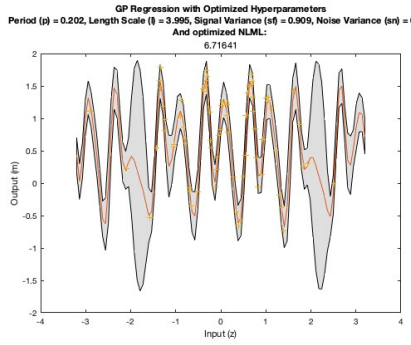
3 Exercise c)

With the periodic covariance function introduced, the hyper-parameters we care about now becomes four: period, characteristic length scale, signal variance, and noise variance. Two plots of different initial hyper-parameters settings are shown in Figure.4.

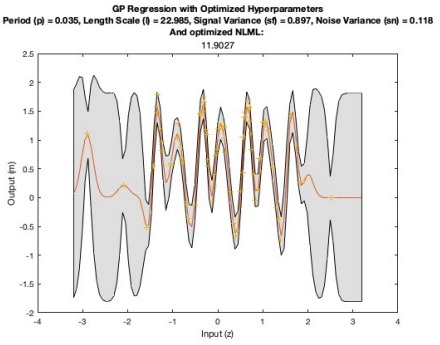
The first plot with $\theta_{opt} = [0.202, 3.995, 0.909, 0.115]$ compared with the fit in part (a) with RBF kernel has similar behaviours in regions with multiple observations. However, the former shows a periodic pattern in regions with no observations, which is characteristic of the periodic kernel that assumes the function has a repetitive behavior. Also, it has a higher log marginal likelihood, implying a potential of better performance.

An interesting thing to note that when $\theta_{opt} = [0.035, 22.985, 0.897, 0.118]$, despite the differences in the nature of the covariance function, the plot looks quite similar to fit from part (a) with RBF kernel. While the periodic kernel inherently models periodicity, a large length scale ($l = 22.985$) can cause it to produce smoother patterns that resemble the behavior of an RBF kernel, especially in areas without data points. This effect is evident in the regions of uncertainty in the plot, where the periodic effects are less discernible, and the regression appears to be smoother and less oscillatory.

The observed data displays periodic tendencies but also showcases deviations, implying it doesn't follow a perfect periodic pattern. The fit of the GP with the periodic kernel suggests the data has a periodic element, but error bars in data-sparse regions show the model's uncertainty about a strict periodic nature. The RBF kernel's fitting, which lacks a strict periodic patterns, further hints that the data, while having periodic traits, isn't purely periodic.



(a) $\theta_{opt} = [0.202, 3.995, 0.909, 0.115]$ and log marginal likelihood = -6.716



(b) $\theta_{opt} = [0.035, 22.985, 0.897, 0.118]$ and log marginal likelihood = -11.903

Figure 3: Periodic covariance functions with different hyper-parameters initializations

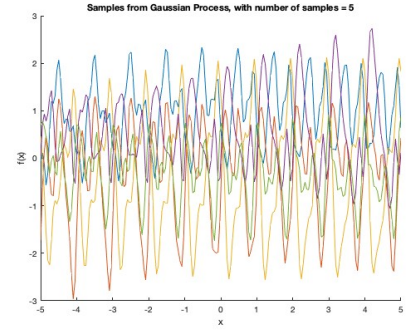
4 Exercise d)

Figure.4a and Figure.4b show the plots of different number of noise free samples generated from the Gaussian Process with product of periodic kernel and RBF kernel, which has the expression:

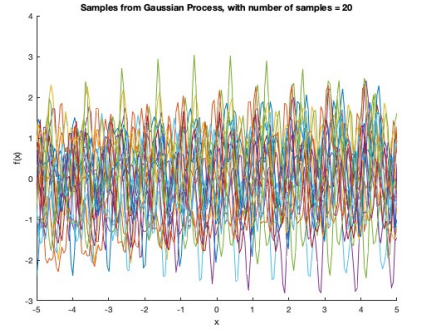
$$k(x, x') = \sigma^2_{period}\sigma^2_{RBF} \exp\left(-\frac{2\sin^2(\pi p(x - x'))}{l^2_{\text{period}}} - \frac{(x - x')^2}{2l^2_{\text{RBF}}}\right)$$

Drawing a larger number of samples, like 20, offers a more detailed depiction of the GP's intrinsic uncertainty. When sampling from a GP and performing the Cholesky decomposition, a small positive value is added to the diagonal. This ensures that all eigenvalues are shifted away from zero or negative values, reinforcing the matrix's positive definiteness, which is a vital prerequisite for the decomposition.

The hyper-parameters settings for $[l_{\text{period}}, p, \sigma_{period}, l_{\text{RBF}}, \sigma_{RBF}]$ is $[0.607, 1, 1, 7.389, 1]$. In particular, the setting $l_{periodic} < 1$ causes relatively rapid changes between the peaks in periodic kernel due to localized correlations. The peak-to-peak distance appears to be roughly 1, corresponding to $p = 1$. The amplitude, regulated by the periodic kernel's signal variance σ_{period}, is not amplified since it's set to 1. The RBF length scale, $l_{RBF} = 7.389$, causes smooth overall trends in RBF kernel. The signal variance of RBF kernel σ_{RBF} is the same as σ_{period}, which are unit and has no effects on the amplitude. In summary, the periodicity, smoothness, and amplitude of the functions are the most noticeable features by combining the two kernels.



(a) 5 random noise free samples drawn from the GP



(b) 20 random noise free samples drawn from the GP

Figure 4: Random noise free samples generation from GP with product of two covariance kernel

5 Exercise e)

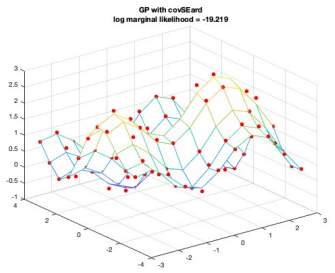
Figures.5 shows Gaussian Process fits for two different models on 2D data:

$$\text{Model 1:} \quad K_1(X, X') = \sigma_f^2 \exp\left(-\frac{1}{2}(X - X')\, D^{-2}\, (X - X')^T\right)$$

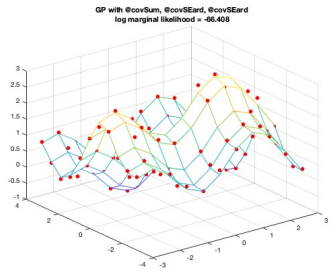
$$\text{Model 2:} \quad K_2(X, X') = \sigma_{f1}^2 \exp\left(-\frac{1}{2}(X - X')\, D_1^{-2}\, (X - X')^T\right) + \sigma_{f2}^2 \exp\left(-\frac{1}{2}(X - X')\, D_2^{-2}\, (X - X')^T\right)$$

D are diagonal matrices of the squared length scales l^2. Model 1 has a log marginal likelihood of -19.219 and Model 2 has a log marginal likelihood of -66.363. Visually, both models seem to fit the data relatively well, as the surface closely follows the observed data points. However, the squared residuals presented in the 2D bar plots highlights substantial differences. For M1, the squared residuals are generally higher, suggesting that the model might be missing some intrinsic patterns or local variations in the data. On the other hand, M2 exhibits squared residuals that are more distributed around 0, pointing towards a closer fit to the individual data points.

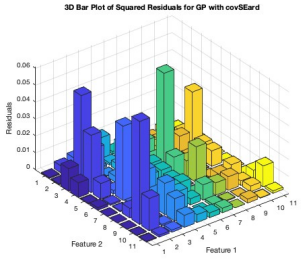
Therefore, Model 1 offers a smoother fit, capturing the overall trend but not smaller fluctuations or noise. In contrast, Model 2 is more flexible, fitting both the general trend and minor fluctuations, potentially including noise. However, Model 1 is favored due to its higher log marginal likelihood. The log marginal likelihood function inherently favors models that fit the data well while imposing penalties on those with high complexity. This aligns with the **Occam's razor principle**: a simpler model is favored when it provides comparable explanatory power. Thus, Model 1's simplicity gives it an advantage in marginal likelihood over the more complex, but potentially overfitted, Model 2.



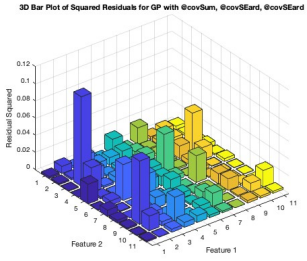
(a) Model 1 fit



(b) Model 2 fit



(c) 3D Bar Plot of Squared Residuals for Model 1



(d) 3D Bar Plot of Squared Residuals for Model 2

Figure 5: Fits of GP models with different settings on 2D data

6 Appendix

6.1 (a), (b), (c)

```
% Tasks (a), (b), (c) share the same algorithm structure
% initializations
ell = -1; sf = 0; hyp.cov = [ell, sf]; sn = 0; hyp.lik = sn;
meanfunc = []; hyp.mean = []; covfunc = @covSEiso; likfunc = @likGauss;

% optimization
hyp_opt = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);

% negative log marginal likelihood
nlml_opt = gp(hyp_opt, @infGaussLik, meanfunc, covfunc, likfunc, x, y);

% posterior mean and covariance
[m, s] = gp(hyp_opt, @infGaussLik, meanfunc, covfunc, likfunc, x, y, z);
```

6.2 (d)

```
% define the Covariance Function
covFunc = {@covProd, {@covPeriodic, @covSEiso}}; hyp.cov = [-0.5, 0, 0, 2, 0];

% covariance matrix
x = linspace(-5, 5, 200)'; K = feval(covFunc{:}, hyp.cov, x, x);

% Cholesky decomposition
K_stable = K + 1e-6 * eye(length(x)); L = chol(K_stable, 'lower');

% sampling functions and plot
num_samples = 10; f_samples = L * randn(length(x), num_samples);
```

6.3 (e)

```
clear all; close all;

data = load("cw1e.mat"); x = data.x; y = data.y;

% model 1: GP with covSEard covariance
covfunc1 = @covSEard; hyp1.cov = ones(3,1); hyp1.lik = 0.1;

% model 2: GP with {@covSum, {@covSEard, @covSEard}} covariance
covfunc2 = {@covSum, {@covSEard, @covSEard}}; hyp2.cov = 0.1 * randn(6,1); hyp2.lik = 0.1;

% global Likelihood function
likfunc = @likGauss;

% optimization
hyp1_optimized = minimize(hyp1, @gp, -100, @infGaussLik, [], covfunc1, likfunc, x, y);
hyp2_optimized = minimize(hyp2, @gp, -100, @infGaussLik, [], covfunc2, likfunc, x, y);

% Predictions for both models
[mu1, ~] = gp(hyp1_optimized, @infGaussLik, [], covfunc1, [], x, y, xTest);
[mu2, ~] = gp(hyp2_optimized, @infGaussLik, [], covfunc2, [], x, y, xTest);
```