4F13 Probabilistic Machine Learning: Coursework #1: Gaussian Processes

Hong Ge

Due 12:00 noon, Friday Nov 7th, 2025 online via moodle

Your answers should contain an explanation of what you do and a maximum of 2-4 central commands used to achieve it (full code listings are discouraged). You must give an *interpretation* of what the numerical values and graphs you provide *mean* – why are the results the way they are, and what are the consequences? Explain your reasoning. Figures should be produced using the plotting utilities provided in the supplied Python file (ie, `gp.py`) where possible.

Each question should be labelled and answered separately, and all carry equal weight. The total length of answers must not exceed five sides of A4 (plus a cover page), using at least 11pt font and 1-inch margins. You will need the `GPy` package in Python, installable via `pip install GPy`. Documentation is at `https://gpy.readthedocs.io/en/deploy/`.

a) Load data from `cw1a.mat` using `scipy.io.loadmat()`. Train a GP with a squared exponential covariance function, `GPy.kern.RBF()`. Initialise the hyperparameters (kernel lengthscale ℓ, kernel variance σ_f^2, noise variance σ_n^2) as follows: $\ell = e^{-1}, \sigma_f^2 = 1, \sigma_n^2 = 1$. Minimise the negative log marginal likelihood. Show the 95% predictive error bars. Explain the values of the optimised hyperparameters and the shape of the predictive error bars.

b) How can you find out whether the hyper parameter optimum is unique, or whether there may be other local optima? If there are local optima, find some, and explain what the model is doing in each case. Which fit is best, and why? Quantify how confident are you about this and why?

c) Train instead a GP with a periodic covariance function, `GPy.kern.StdPeriodic()`. Compare the behaviour of the error-bars with a). Do you think the data generating mechanism (apart from the noise) was strictly periodic[1]? Carefully discuss the evindence for or against periodicity.

d) Generate random (essentially) noise-free functions evaluated at `X = np.linspace(-5.0, 5.0, 200, dtype=np.float64).reshape(-1, 1)` from a GP with covariance function defined as the product of `GPy.kern.RBF` (with $\ell = e^2, \sigma_f^2 = 1.0$) and `GPy.kern.StdPeriodic` (with $\ell = e^{-0.5}, p = 1.0, \sigma_f^2 = 1.0$). In order to apply the Cholesky decomposition to the covariance matrix, you may have to add a small diagonal matrix, for example `1e-6*np.eye(200)`, why? Plot some sample functions. Carefully explain the relationship between the properties of those random functions and the form of the covariance function.

e) Load `cw1e.mat`. This data has 2-D input and scalar output. Visualise the data, e.g., using

```
plt.plot_surface(X[:,0].reshape(11,11), X[:,1].reshape(11,11), y.reshape(11,11)).
```

Rotate the data, to get a feel for it. Compare two GP models of the data: one with a single SE covariance `GPy.kern.RBF(input_dim=2, ARD=True)`, and the other with a sum of two SE covariance functions `GPy.kern.RBF(input_dim=2, ARD=True) + GPy.kern.RBF(input_dim=2, ARD=True)`. For the second model, be sure to break symmetry with the initial hyperparameters (lengthscales and variance of each SE covariance), e.g., set them with `0.1*np.random.randn(6)`.

Compare the models: give a careful quantitative interpretation of the relationship between data fit, model complexity and marginal likelihood for each of the two models; which model is best and why, explain your reasoning.

[1] By strictly periodic, is meant a function where the exists a p such that $f(x) = f(x + np)$ for integer n and all x, not just a a function which “goes up and down”.