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

Carl Edward Rasmussen

Due: 12:00 noon, Friday Nov 6th, 2020, online via moodle

Your answers should contain an explanation of what you do, and 2-4 central commands to achieve it (but complete listings are unnecessary). You must also give an *interpretation* of what the numerical values and graphs you provide *mean* – why are the results the way they are? **Each question should be labelled and answered separately and distinctly.** Total combined length of answers must not exceed 1000 words; clearly indicate the actual total number of words in your coursework.

You need the Gaussian Processes for Machine Learning (GPML) toolbox (version 4.2) for matlab and octave. Get the toolbox and walk through the documentation concerning regression from the Gaussian Process Web site at `www.gaussianprocess.org/gpml/code` Note, that sometimes hyperparameters are encoded using their logarithms (to avoid having to deal with *constrained* optimization for positive parameters), but you will want to report them in their natural domain. All logs are natural (ie, base e). All questions carry approximately equal weight.

a) Load data from `cw1a.mat`. Train a GP with a squared exponential covariance function, `covSEiso`. Start the log hyper-parameters at `hyp.cov = [-1 0]; hyp.lik = 0;` and minimize the negative log marginal likelihood. Show the 95% predictive error bars. Comment on the predictive error bars and the optimized hyperparameters.

b) Show that by initializing the hyperparameters differently, you can find a different local optimum for the hyperparameters. Try a range of values. Show the fit. Explain what the model is doing. Which fit is best, and why? How confident are you about this and why?

c) Train instead a GP with a periodic covariance function. Show the fit. Comment on the behaviour of the error-bars, compared to your fit from a). Do you think the data generating mechanism (apart from the noise) was really strictly periodic? How confident are you about this, and why? Explain your reasoning.

d) Generate random (essentially) noise free functions evaluated at `x = linspace(-5,5,200)';` from a GP with the following covariance function: `{@covProd, {@covPeriodic, @covSEiso}}`, with covariance hyperparameters `hyp.cov = [-0.5 0 0 2 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*eye(200)`, why? Plot some sample functions. 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, for example using `mesh(reshape(x(:,1),11,11),reshape(x(:,2),11,11),reshape(y,11,11));` Rotate the data, to get a good feel for it. Compare two GP models of the data, one with `covSEard` covariance and the other with `{@covSum, {@covSEard, @covSEard}}` covariance. For the second model be sure to break symmetry with the initial hyperparameters (eg by using `hyp.cov = 0.1*randn(6,1);`).

   Compare the models: How do the data fits compare? How do the marginal likelihoods compare? Give a careful interpretation of the relationship between data fit, marginal likelihood, and model complexity for these two models.