4F13 Machine Learning: Coursework #1: Gaussian Processes

Carl Edward Rasmussen & Zoubin Ghahramani

Due: 4pm February 14th, 2013 to Laura Reed, room BNO-37

In this assignment, you'll need the Gaussian Processes for Machine Learning (GPML) toolbox 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 logs (ie, base e).

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? Hand in a maximum of 5 pages.

a) 10% : Load data from `cw1d.mat`. Train a GP with a squared exponential covariance function, `covSEiso`. Start the 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) 10% : Show that by initializing the hyperparameters differently, you can find a different local optimum for the hyperparameters. Show the fit. Explain what is going on. Which fit is best, why?

c) 10% : 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 was really periodic? Why, why not?

d) 10% : Generate 100 data points at `x = linspace(-5,5,100)';` from a GP with the following covariance function: `{@covProd, {@covPeriodic, @covSEiso}}`, with covariance hyperparameters `hyp.cov = [-0.5 0 0 2 0]`. Don't add noise to the function values. 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(100)`, why? Plot some sample functions. Explain their behaviour.

e) 10% : 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. Fit the data using a GP with covariance function `covSEard`. Comment on the fit. How much noise is there in the data?

f) 10% : Fit the data instead using `covSEiso`. Is this a better model, why, why not? What is the relative probability of the two models, e) and f)?

g) 15% : Use instead `{@covSum, {@covSEard, @covSEard}}` and be sure to break symmetry with the initial hyperparameters (eg by using `hyp.cov = 0.1*randn(6,1);`). Why is symmetry breaking necessary? Fit. Explain the model. Is the predicted function significantly different from the one obtained in e)? Is the probability of the model very different? Explain how to reconcile your last two answers.

h) 10% : Load the data from the file `mauna.mat`. This a time series of monthly average atmospheric Carbon Dioxide concentrations in parts per million (vol) measured at Mauna Loa in Hawaii. The variables are `trainyear, trainCO2, testyear, testCO2`. Train a GP model with a linear mean function and a squared exponential covariance function on the training portion of data, to predict the CO2 values from time variable. Show and comment on the fit and the hypers, and the predictions for the test data.

i) 15% : Use an additive covariance structure `{@covSum, {...}}` with what you think may be suitable components to model the data. Be careful to initialize the hyperparameters for each component to reasonable values, so that the minimizer finds a good (local) minimum for the negative log marginal likelihood. Explain, and show your fit.