Assignment 8: Gaussian Process Regression

Unsupervised Learning

Zoubin Ghahramani & Carl Edward Rasmussen

Due: Tues 11 Dec, 2001

We consider a univariate regression case, ie we are modeling the scalar y as a function of (conditional on) the scalar x. We suggest that you use the following covariance function:

$$C(x^{(p)}, x^{(q)}) = v \exp(-w(x^{(p)} - x^{(q)})^2) + u\delta_{p,q}$$

with hyperparameters v, w, u. Once you are done with this covariance function, you can try using other, more complicated covariance functions.

1. Write a little piece of code to draw random samples from a Gaussian Process. Plot the functions for various values of the hyperparameters, both without noise ($u = 0$) and with noise ($u > 0$). In the moise-free case, you may need to add a tiny amound of noise (say $u = 10^{-6}$) for numerical reasons. Investigate the role of each of the three hyperparameters.

2. Write a matlab function that computes minus the log evidence and its (three) partial derivatives wrt the hyperparameters. Since the hyperparameters themselves must be positive, you should reparamterize the hypers (using a log transformation).

3. Download the motorcycle data from the course website, and plot the data. Use the `checkgrad.m` (from the `minimize` code) and the motorcycle data to check that the partial derivatives in your function are computed correctly.

4. Using `minimize` and your derivative function, find the maximum likelihood (maximum evidence) hyperparameters for the motorcycle data. Verify that the hyperparameters take on values that seem apropriate for the data.

5. Write code to make (test) predictions. Plot the predictive (test) distribution (eg, by plotting the predictive mean and plus/minus two standard error errorbars). Comment on the success and failures of the model.

6. Bonus: Use the Metropolis algorithm to integrate over the posterior (assume a flat prior for the log hyperparameters) distribution of the hyperparameters.