Supervised Learning 2004 — Assignment 3

Ed Snelson <snelson@gatsby.ucl.ac.uk>

Due Monday 29th November 2004

Please hand in a print out of any code you write for this assignment, and its output. Each question carries equal marks. While discussion is always encouraged, you should do this assignment by yourself.

Q1 Deriving Gaussian process prediction

We have a data set $\{\mathbf{x}_n, t_n\}_{n=1}^N$, a new input $\mathbf{x}^*$, and we want to predict the distribution of the corresponding output t^*. Modelling the data with a GP, we know that the joint distribution over outputs is a multivariate Gaussian with zero mean and covariance K_+:

$$p(\mathbf{t}, t^*) = \mathcal{N}(\mathbf{0}, K_+) \ ,$$

where $\mathbf{t}$ is the vector of training outputs. The covariance can be broken down into block form:

$$K_+ = \begin{bmatrix} K & \mathbf{k} \\ \mathbf{k}^\top & k^* \end{bmatrix} \ ,$$

where K is the $N \times N$ covariance matrix of the training data only.

Show that the mean and variance of the predictive distribution are given by:

$$\mathbb{E}[t^*] = \mathbf{k}^\top K^{-1} \mathbf{t}$$
$$\mathrm{var}[t^*] = k^* - \mathbf{k}^\top K^{-1} \mathbf{k} \ ,$$

showing every line of your working.

You will need to use the following result for the inverse of K_+:

$$K_+^{-1} = \begin{bmatrix} M & \mathbf{m} \\ \mathbf{m}^\top & m \end{bmatrix}$$

where

$$m = 1/\left(k^* - \mathbf{k}^\top K^{-1} \mathbf{k}\right)$$
$$\mathbf{m} = -m K^{-1} \mathbf{k}$$
$$M = K^{-1} + \frac{1}{m} \mathbf{m}\mathbf{m}^\top \ .$$

Hints. Expand out the terms in the exponential of the joint distribution using the block inverse formula given above, but only keep terms involving t^*, as these are the only relevant ones. First obtain the variance from the $(t^*)^2$ term; then obtain the mean from the t^* term. Don't worry about normalising constants, as you know the final distribution must be Gaussian.

Q2 Generating a covariance matrix

Write code to generate a covariance matrix from a given set of 1D inputs $\{x_n\}_{n=1}^N$, using the following Gaussian covariance function:

$$C(x, x') = v_0 \exp\left[-\alpha(x - x')^2\right] \ .$$

Remember: $C_{ij} = C(x_i, x_j)$. Add noise of variance v_1 to the diagonal elements:

$$K = C + v_1 I \ ,$$

where I is the identity matrix.

Test your code by outputting the covariance matrix K for the small set of inputs $\{-1.5,\ -0.2,\ 1.0,\ 1.7,\ 3.0\}$, using the following parameters: $\alpha = 0.4$, $v_0 = 2$, $v_1 = 0.1$. Report this matrix to 3 decimal places, and submit your code. Also convince yourself that your code is working by checking by hand at least a few of the entries of the matrix, as you will need this code for the next questions.

Q3 Finding the parameters

Ideally one would find maximum likelihood parameters using gradient ascent on the log likelihood. In this question we are going to search through a discrete set of parameters, and pick the best ones.

Write code that outputs the log likelihood for given parameters $\{\alpha, v_0, v_1\}$:

$$\begin{aligned}\mathcal{L}(\alpha, v_0, v_1) &= \log p(\mathbf{t}|\{x_n\}, \alpha, v_0, v_1) \\ &= -\frac{1}{2}\log\det K - \frac{1}{2}\mathbf{t}^\top K^{-1}\mathbf{t} - \frac{N}{2}\log(2\pi) \ ,\end{aligned}$$

using your routine from Q2 to calculate the covariance matrix.

Download the data from the course web-site. This is a simple 1D regression data set; the first column contains 51 inputs $\{x_n\}$, and the second column contains corresponding outputs $\{t_n\}$.

Calculate the log likelihood for the same parameters as used in Q2.

Write code to loop through calculating the log likelihood for every combination of parameters where each parameter can take on the values:

$$10^{-2}, 10^{-1.5}, 10^{-1}, \ldots, 10^{1.5}, 10^{2} \ .$$

Don't submit values of the log likelihood for all these combinations as there are $9^3 = 729$ of them! Instead find the parameter setting which gives the maximum log likelihood, and report the values of α, v_0, and v_1, as well as the corresponding log likelihood.

Plot the data points, $\{t_n\}$ against $\{x_n\}$. By looking at the plot, check and explain why the parameters you have obtained are of the right sort of magnitude for this data.

Q4 Implementing GP prediction

Write code that implements the equations you derived in Q1 for GP prediction. Your code should take as input: a set of training data $\{x_n, t_n\}_{n=1}^N$, a set of input points for which predictions are to be made $\{x_m^*\}_{m=1}^M$, and parameters for the covariance. Your code should output the predictive means and variances that correspond to the inputs $\{x_m^*\}$.

Use your code to make predictions for the training data of Q3 at the following set of new inputs:

$$\{x_m^*\} = \{-10, -9.9, -9.8, \ldots, 9.8, 9.9, 10\} \ ,$$

with the 'maximum likelihood' parameters you found in Q3. If you could not do Q3, simply use the setting of the parameters from Q2.

Plot the mean predictions $\{\mathbb{E}[t_m^*]\}$ against $\{x_m^*\}$ over the top of the training data you plotted in Q3. Also plot points at 2 standard deviations either side of these mean predictions. You should obtain a nice smooth fit to the data.

Hints. The simplest (but not the most efficient) method is to use your code from Q2 to generate the $(N+M) \times (N+M)$ covariance matrix of all the training inputs and the predictive inputs together. You can then pick out the appropriate $\mathbf{k}$ vectors and k^* values from this one big matrix (remember the k^* values appear on the diagonal). You can also pick out the training data only $N \times N$ covariance K, and note you only need to calculate K^{-1} once when predicting on the entire set of new inputs.