CX 4803 CML – Spring 2022

Assignment 4

Please show your work as if you were explaining your solution to another student. In particular, please **show all your code** *used to generate the solutions. Submit your solutions as a single pdf file on Canvas. Include your program listings in your pdf file.*

1. Given data (x_i, y_i), $i = 1, \dots, n$, we wish to use Gaussian process regression to predict values $\hat{y}_i$ for x_i, $i = n+1, \dots, n+N$, where $N = 10$. In other words, instead of making a single prediction, we wish to make N predictions. We can do this by considering the vector

$$\begin{bmatrix} y \\ \hat{y} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \hat{y}_{n+1} \\ \vdots \\ \hat{y}_{n+N} \end{bmatrix} \sim N(0, \Sigma)$$

as a sample from a multivariate Gaussian distribution with mean zero and a covariance matrix Σ defined by a covariance function $k(\cdot, \cdot)$.

The covariance matrix Σ can be written as a block 2×2 matrix,

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

   (a) Give expressions for Σ_{11}, Σ_{12}, Σ_{21}, and Σ_{22}. Define any symbols you use.

   (b) Give an expression for $p(\hat{y}|y)$. Be sure to define any symbols you use.

2. The program `demo_gauproc_regr.m` on Canvas plots the **mean** of $p(\hat{y}|y)$ where $\hat{y}$ is the prediction for a large number of values of x between 0 and 1. (This allows a curve to be drawn, rather than a prediction at a single point.)

   Modify the program (or rewrite it from scratch), so that the program also plots 5 example curves from the distribution $p(\hat{y}|y)$ where $\hat{y}$ is the prediction for a large number of values of x between 0 and 1.

   This question is analogous to Figure 3.9 in Bishop, but for Gaussian process regression. Do not use the Bayesian linear regression to complete this question. Hint: this question is related to question 1.

3. The expensive computational step in Gaussian process regression involves the inverse of $C = K + \beta^{-1}I$, where K and β were defined in class. In fact, the inverse of C is never actually computed. Instead, one computes the Cholesky factorization of C, i.e., $C = LL^T$, where L is the lower triangular Cholesky factor. A quantity such as $w = C^{-1}y$ can be written as

$$\begin{aligned} w &= C^{-1}y \\ &= (LL^T)^{-1}y \\ &= L^{-T}(L^{-1}y) \end{aligned}$$

which shows you can first compute $v = L^{-1}y$ and then compute $w = L^{-T}v$. Computing the Cholesky factorization of C is computationally much cheaper than computing C^{-1}.

Modify your code from the previous question to make use of this information to make your code more efficient.

4. A data set is provided in the file `a4data` on Canvas. The data consists of measurements at 705 points in 2-D. The file contains 3 columns. The first column is x_1 for each point. The second column is x_2 for each point. The third column is the measurement.

   Use Gaussian process regression to predict values at three points:

point	prediction
(30,30)	?
(25,5)	?
(4,4)	?

   You should use a Gaussian kernel function. Experiment with different values of the kernel function standard deviation s and also the noise variance $1/\beta$ that will allow you to obtain "better" predictions. To test what are good values of s and β, you could try to hold back some of the data (i.e., not use it for training) and use this held back data to test your Gaussian process regression (since you know the measured values for held back data).

   Be sure to explain what you did and also show your code for Gaussian process regression for this 2-D data.