Exercise sheet: Gaussian Processes

The following exercises have different levels of difficulty indicated by (*), (**), (***). An exercise with (*) is a simple exercise requiring less time to solve compared to an exercise with (***), which is a more complex exercise. Don't worry if you can't do a (***) exercise, as these are beyond what will be expected in the exam. They are instead intended to encourage further reading and deeper understanding.

1. (*) If someone is using an exponentiated quadratic covariance function, and they increase the lengthscale, what effect will this increase have on the covariance between two points (e.g. at $x_1 = 1$ and $x_2 = 3$)?

   Answer:

   The covariance will increase, as every point will become more related to other points.

2. (*) For an arbitrary choice of covariance function: Does the covariance between points always get smaller as the two points are placed further apart?

   Answer:

   No. For example a periodic kernel will have a higher covariance between points one-period apart.

3. (*) Are covariances always positive? (do all covariance functions lead to positive covariances?)

   Answer:

   No. An example: the linear kernel, $k(x_1, x_2) = x_1 x_2$, which will have a negative covariance between points on either side of the origin. The periodic kernel is another example.

4. (*) What is the mean and covariance of $p(y_2)$ if,

$$\begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{12}^\top} & \boldsymbol{\Sigma_{22}} \end{bmatrix}\right)$$

   Answer:

This is marginalisation, and to do that to a Gaussian we just include the components of the Gaussian along that axis: $p(\boldsymbol{y_2}) = N\left(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_{22}}\right)$

5. (**) We have two observations:

$$X = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ 1 \end{bmatrix}$$

We don't currently know y_1, yet. We will use the exponentiated quadratic with a lengthscale of 2 (and kernel variance of 1) to compute a prediction at $x_* = 3$.

$$k(x_1, x_2) = \exp\left(\frac{-(x_1 - x_2)^2}{2l^2}\right)$$

a. Compute $\boldsymbol{k}_{*f}$ and K_{ff}^{-1}.
b. Compute the product $\boldsymbol{k}_{*f} K_{ff}^{-1}$ and so write down an expression for the posterior mean, in the form of a matrix times $\boldsymbol{y}$. Substitute in the value of y_2 that we know, thereby having a (linear) expression for y_* in terms of y_1.
c. You will find that the prediction, y_* linearly dependent on y_1, in a negative direction: So as y_1 gets larger, y_* gets smaller. Explain why. What is the intuition for this result?

Answer:

a. Just to one decimal place for simplicity,

$$\boldsymbol{k}_{*f} = \begin{bmatrix} 0.6 & 0.9 \end{bmatrix}.$$

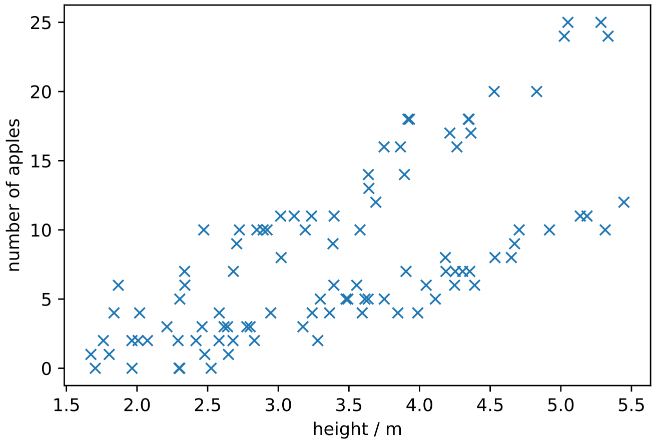
$$K_{ff}^{-1} = \begin{bmatrix} 4.5 & -4 \\ -4 & 4.5 \end{bmatrix}.$$

b. So $y_* = \begin{bmatrix} -0.8 & 1.6 \end{bmatrix} \boldsymbol{y}$. Substituting in $y_1 = 1$ and multiplying, we get:

$$y_* = 1.6 - 0.8 y_1.$$

c. The training point at x_2 acts like a fulcrum. As y_1 increases on one side of the fulcrum, the value at x_* on the other side will get smaller.

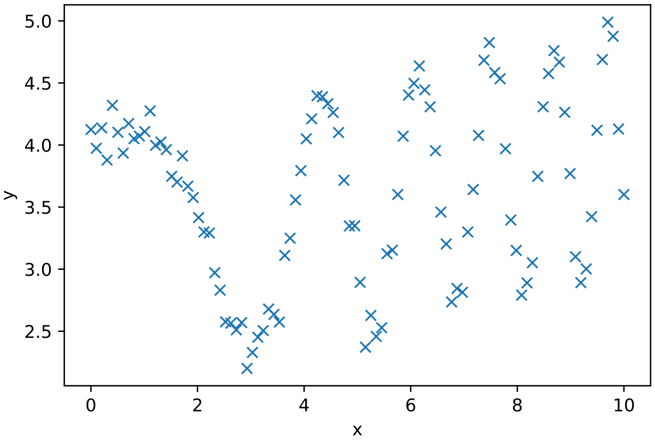
6. (*) In an orchard there are two types of apple tree (you don't know which is which). You measure the number of apples for different trees and plot them against the tree heights. What would be the problem(s) with modelling this with standard Gaussian process regression, with a Gaussian likelihood?



Answer:

The most obvious problem is that the output is multiplemodal, while a Guassian likelihood assumes that the output (for a given input) is a Gaussian distribution, when it clearly consists of two peaks. Other issues are that the outputs are positive only (the output of the model will have density everywhere) and are discrete (the model output is continuous). This could be better modelled using a mixture model (with an indicator vector deciding which class a tree belongs in, and two GP regressors fitting each set of trees.

7. (**) You have some data you want to fit with a Gaussian process. What would be the problem with using a standard exponentiated quadratic kernel?



Answer:

The lengthscale seems to get shorter as x gets bigger. The standard EQ kernel doesn't handle this, so would be a poor choice of kernel in this case. This is an example of where we would need a *non-stationary kernel*.

non-stationary kernel: One in which we can't just write down the kernel function as a function of the relative difference, i.e. $k(x_1 - x_2)$, but instead we need to know the absolute values of the inputs (not just their relative locations). See chapter 4 of C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning.

8. (***) What makes a valid kernel? The definition is that it must be positive semidefinite. Specifically, a kernel is said to be positive semidefinite if[1],

$$\int k(\boldsymbol{x}_1, \boldsymbol{x}_2) f(\boldsymbol{x}_1) f(\boldsymbol{x}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \geq 0$$

this is sort of the infinite version of the finite version for a matrix, $\boldsymbol{x}^\top K_{xx} \boldsymbol{x} \geq 0 \; \forall \; \boldsymbol{x} \in \mathcal{R}^D$. Where K_{xx} is the covariance matrix between our points in $\boldsymbol{x}$. We'll use this finite version as it's easier to work with. The linear kernel (for one dimensional x) is $k(x_1, x_2) = x_1 x_2$. Show that this is a valid kernel. Hint: Write out the covariance matrix between an arbitrary pair of points, $[x_1, x_2]$ and compute $\boldsymbol{x}^\top K_{xx} \boldsymbol{x}$, expanding out the expression so it is the simple sum of terms with various combinations of x_1 and x_2. Think about if any of the terms can be negative?

Answer:

The covariance matrix will be:

$$K_{xx} = \begin{bmatrix} x_1 x_1 & x_1 x_2 \\ x_2 x_1 & x_2 x_2 \end{bmatrix}.$$

So the product of $\boldsymbol{x}^\top K_{xx} \boldsymbol{x}$ is:

$$K_{xx} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} x_1 x_1 & x_1 x_2 \\ x_2 x_1 & x_2 x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Multiplying all this out we get:

$$\begin{bmatrix} x_1 x_1 x_1 + x_2 x_2 x_1 & x_1 x_1 x_2 + x_2 x_2 x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$x_1^4 + x_1^2 x_2^2 + x_1^2 x_2^2 + x_2^4.$$

We note that x_1 and x_2 are both real, so x_1^2, x_1^4, x_2^2 and x_2^4 are all non-negative. And so we can say that their products and sums are also non-negative, so the inequality holds:

$$x_1^4 + x_1^2 x_2^2 + x_1^2 x_2^2 + x_2^4 \geq 0.$$

[1] I've been a bit loose with the notation here, see p80 of C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, for more precise notation and further discussion.