Lab 1 – Linear and GP regression

Short course on Statistical modelling for optimization

N. Durrande - J.C. Croix, Universidad Tecnológica de Pereira, 2016

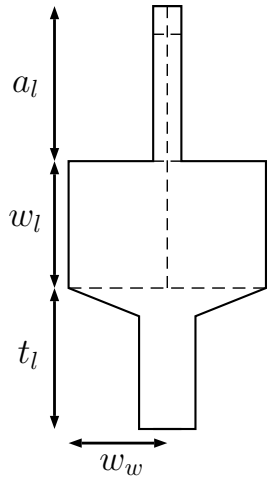
This lab session is composed of two independent parts: the first one on linear regression and the second one on Gaussian process regression. Although they both are equally important for the course, it is not possible to do tomorrow's lab if you have not answered Question 6 of Section 1. Make sure you do!

One easy way to get a Python distribution with the required packages is to use the Anaconda environment from Continuum Analytics. A free version of Anaconda can be downloaded and installed from `https://store.continuum.io`. Installing Anaconda will also provide the program "spyder" which is a useful scientific python development environment.

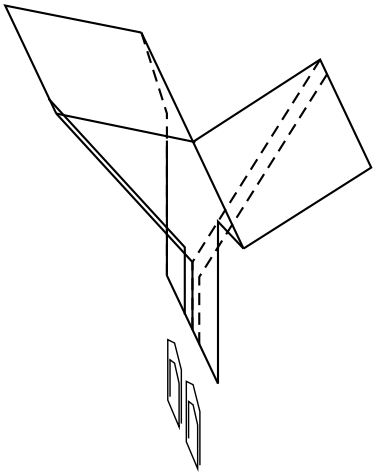
1 Linear regression

The aim of this section is to use data gathered from previous experiments on paper helicopters to gather some information on the influence of the various variables and to define the region of the input space that contains the optima.

Data The file *lab1_data.csv* contains the input and output for 30 experiments. The first 4 columns correspond respectively to the parameters w_l, w_w, t_l, a_l and the last 2 give the falling time from a 7m hight (each experiment is run twice).



Before Folding


After Folding

Code An incomplete implementation of the lab session is given in the file *lab1_LR.py*. This sample of code contains the data loading and some elements of a linear regression program.

Q1. Do some basic data analysis and plotting on X and F. Give an estimate of the observation noise.

Q2. Complete the function `LR` from the file *lab1_LR.py* such that this function returns the estimate $\hat{\beta}$ and its covariance matrix.

Q3. Complete the function `predLR` and build a first regression model based on a constant, a linear effect (say w_l) and a quadratic effect for the same variable. Plot the model using the function `plotModel` and compute the R^2. Do some variables seem more influential than others?

Q4. Repeat the previous question with more than one variable. You may use the function `pvalue` to test if the effect of some predictors is statistically significant.

Q5. Try replacing (or adding) some (or all) of the inputs by some physical quantities such as the surface of the wings, their angle, the overall length of the helicopter before being fold, ... Can you improve the model and gather some information about the physics of the phenomenon?

Q6. For the parametrization of the problem of your choice, define a hypercube that can be used to refine the search of the optimal helicopter. Do not forget the physical constraint... the helicopter should fit on an A4 paper sheet! The maximum dimensions, including printers margins, are thus 27cm$\times$18cm.

2 Gaussian process regression

The code sample *lab1_GPR.py* will help you for this section. It contains some of the most classical covariance functions (ie kernels). Feel free to add new ones!

Q1. Can you recognize the kernels that are already coded? Give the functions a proper name. Plot various of them and study the influence of the parameters. Why do some of them do not have a lengthcale parameter?

Q2. Complete the function `sampleGP(x,kern,n,**kwargs)`. This function should return n samples evaluated at x of a centred GP with kernel *kern*. The kernel parameters σ^2 and θ may be specified by the user in the input `**kwargs`.

Q3. Plot sample paths for various kernels and parameters. What is the influence of the variance and lengthscale parameters?

Q4. Finish writing the function `predGPR`. It should return the vector of conditional mean $m(x)$ and the conditional covariance matrix $c(x,x)$.

Q5. Build a model based on a 5 observations of the toy function `ftest` (input space is $(0,1)$). Plot the model using `plotModel` and study the influence of the kernel, its parameters and the DoE X. Which kernel, parameter values, DoE seem to give the best model? To answer this question, compare the model predictions with actual values of the function on a test set.