Short Course on Statistical Modeling for Optimization, lab sessions report

Juan David Gil   Gustavo Gutiérrez

1 Introduction

This is a report on the course *Statistical Modelling for Optimization* held at the *Universidad Tecnológica de Pereira* from June 28th to July 1st. The practical part of course consisted of four lab sessions with a common goal of optimizing the flying time of a paper helicopter model. In this document we report our findings and conclusions of the lab sessions. The notebooks of each lab session are publicly available[1].

2 Lab session 1

2.1 Linear regression

For this part we were asked to analyze the data corresponding to two realizations of the helicopter experiment.

Noise between realizations of the experiment. Figure 1a shows the plot between the times of the experiments for every realization. The average variance between the time of the experiments is 0.0425 s. This indicates that there is not much noise present in the data.

Relations in the input data. According to the provided data, we can appreciate some relation between different variables. For instance, the wing length and the falling times seems to be linearly correlated according to figure 2. The same correlation but this time less clear is present in figure 3 with the arm length. Finally, the arm and tail lengths might also be correlated as shown in figure 1b.

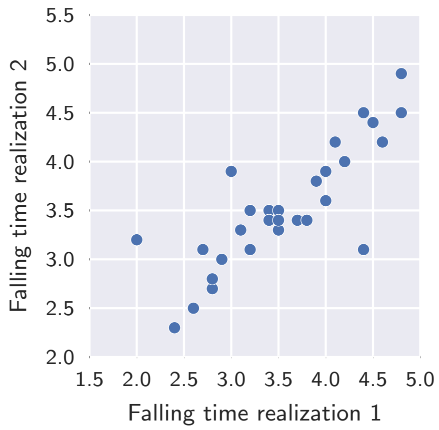
Despite of the described relations among the different variables, there are no strong correlations among the variables of the model. Linear correlations are just apparent but not clearly strong.

Linear regression. A first model $m(X)$ was approximated by using a linear regression. The basis functions were $\phi(x) = [1, x_0, x_0^2]$. Those functions were based on effects on the wing length (x_0). However, the coefficient of determination (R^2) for the obtained model was 0.237. A better model is obtained using the basis functions $\phi(x) = [1, x, x_3^2, \sin(x_3), \cos(x_0)]$. R^2 for the new model is 0.778 which (graphically) fits better the data according to figure 4.

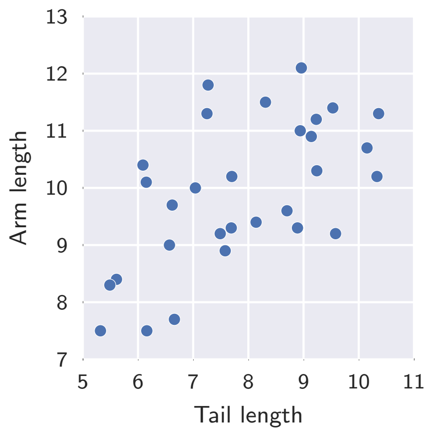
Several functions were tried as basis. The ones with more effect are the constant function (1), the linear effect on all the variables (x), the cosine effect on the wing length ($\cos(x_0)$). However the other functions also contribute to improve R^2.

P-values. The contribution of the basis functions to the linear regression are confirmed by the following vector of p-values: $\langle 0.0, 0.0, 0.0, 0.0, 0.02, .01, 0.01, 0.0 \rangle$ in summary there is a high probability that all the basis functions contribute to the regression as each value is close to 0.0.

[1] `https://github.com/juangil/SMFO`



(a) Comparison between the times measured in the two realizations.



(b) Comparison between the tail and the arm lengths.

Figure 1

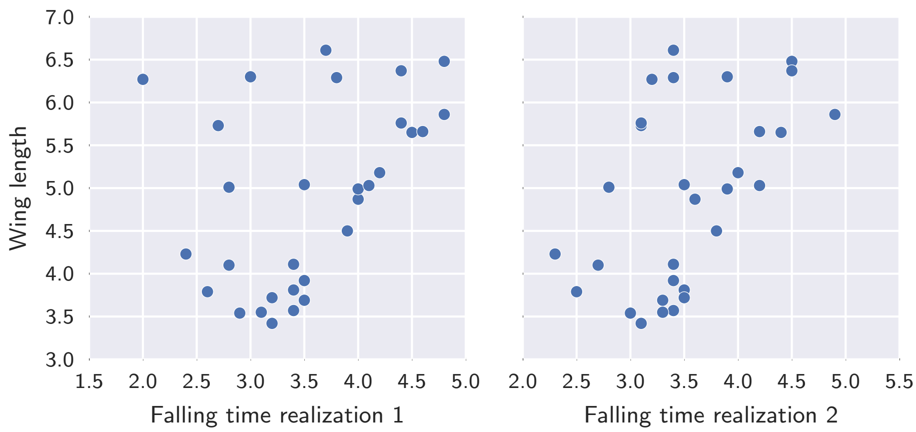


Figure 2: Correlation between the wing length (in centimeters) and the falling times (in seconds) of the two realizations.

Figure 3: Correlation between the arm length (in centimeters) and the falling times (in seconds) of the two realizations.

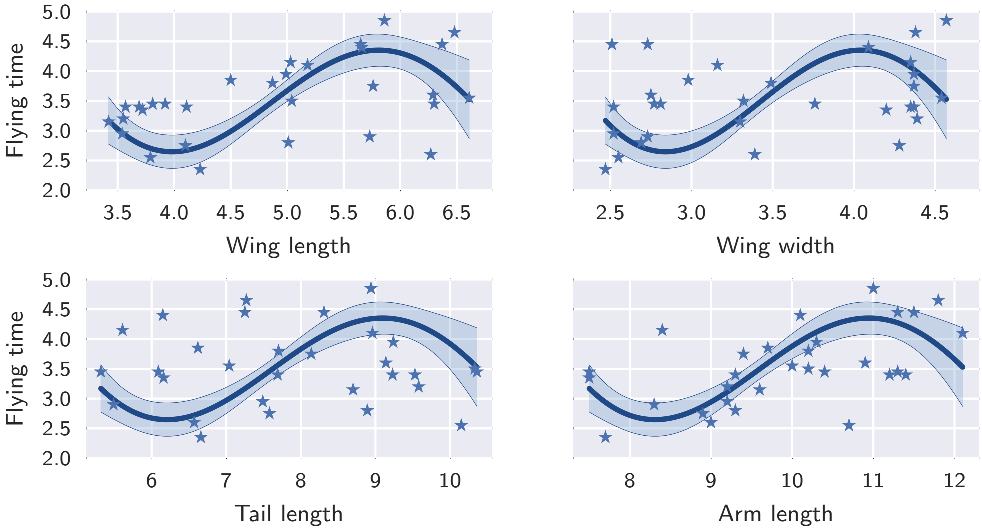


Figure 4: Linear regression based prediction.

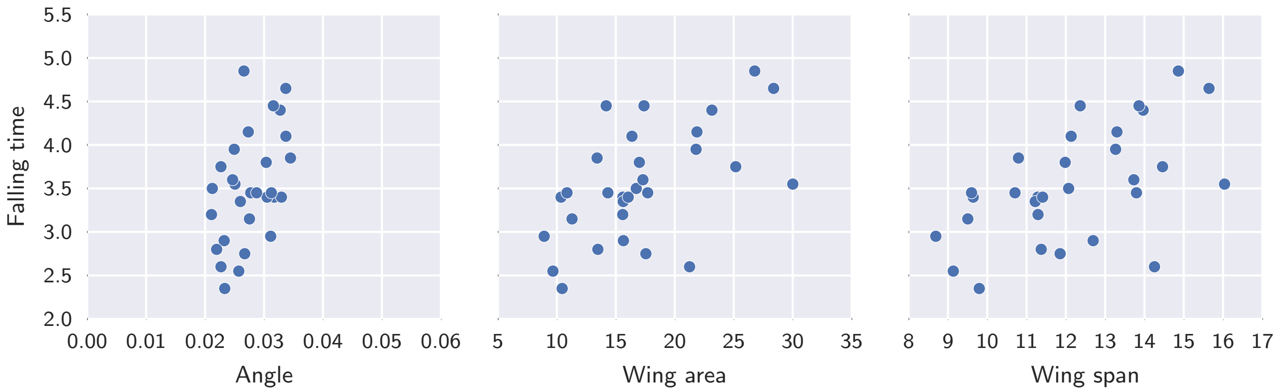


Figure 5: New physical measures that might affect the experiment. Left: angle (in degrees) between wing and tail. Middle: area of the wing (in cm^2). Right: wingspan (in cm).

Influence of other physical quantities. In this point we are interested in the effect of other physical properties in the experiment. Specifically, we were suggested to observe the angle between the wing and the tail and additionally we computed the wing area and the wingspan (distance between the farthest points of the two wings). Figure 5 presents the relation between those measures and the flying time. The figure shows a clear linear trend between the angle and the flight time. Moreover, the slope of that trend suggest the angle as a good candidate to optimize. The other measures are not as promising. There is no clear trend in the wing area nor in the wingspan.

2.2 Gaussian process regression

Kernel names. In the first point we are presented a several kernel implementations and asked to identify them. The first two implementations are actually the same and both represent the *Matern32* kernel. The third is the implementation of the constant kernel. The fourth implementation is for the *Brownian* kernel. The last implementation corresponds to the *Gaussian* kernel.

Sampling from a Gaussian process and plots. The sampling function is completed by evaluating the provided kernel on the input data. After that n random samples are taken from a multivariate normal with the provided mean (μ) and the kernel as the covariance parameter. Figure 6 presents samplings

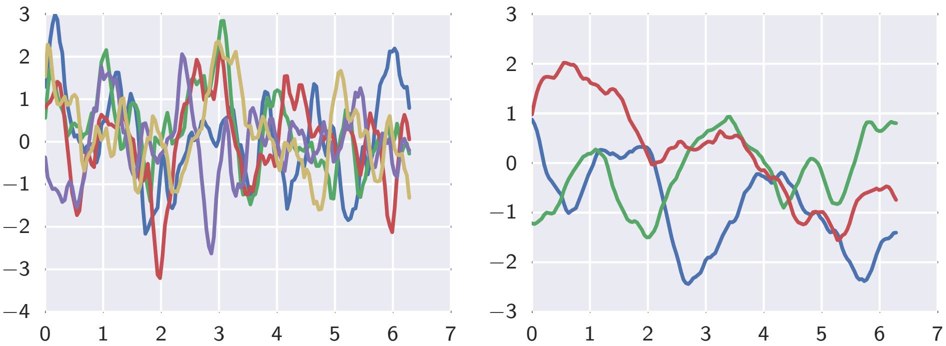


Figure 6: Sample from a Gaussian process with kernel evaluated using the wing length of the helicopter. The kernel for the left sampling of five functions is *Matern32* with $\sigma^2 = 1.0$ and $\theta = 0.2$. The second plot is the same kernel, this time with only three functions and $\theta = 0.8$

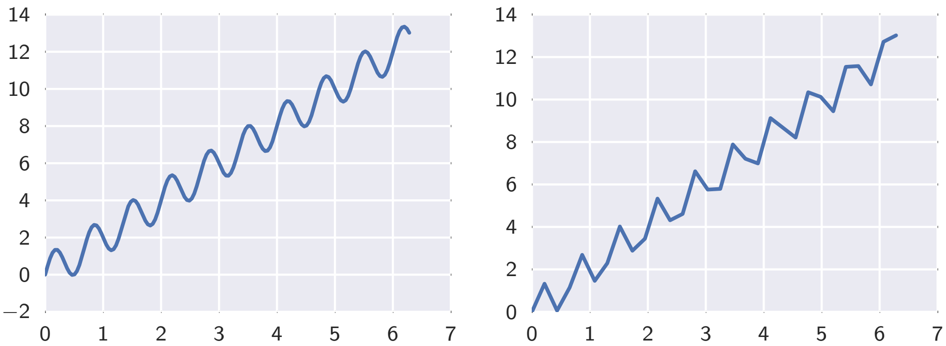


Figure 7: Plots for $f(x) = \sin(3\pi x)$ using 150 (left) and 30 (right) samples.

of the same kernel with different parameters. The plot on the left uses $\theta = 0.8$ which leads to a wider "waves".

Gaussian process regression for prediction of a toy function. In this point we are asked to approximate the function $f(x) = \sin(3\pi x)$ plotted in figure 7.

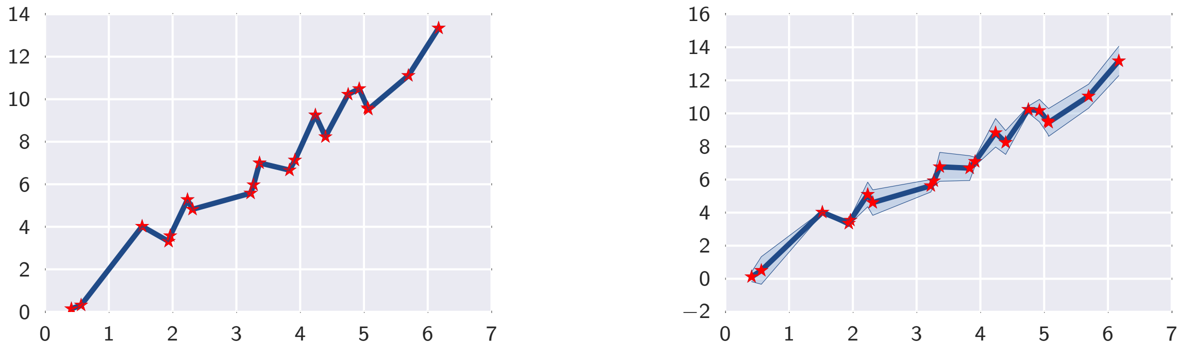
The prediction was performed using the *Matern32* kernel as it provided the best results. The model and its variance are presented in figure 8. 20 points were predicted using the two trained models. The quality of the predictions is measured by the sum of variances between real and estimated values. The respective quantities were: 0.000132 and 0.095207 which let us conclude that the model trained with 150 samples is better than the one trained with just 30. This is also appreciated in the plots by looking at the variance. The light blue area is quite noticeable for the second model.

3 Lab session 2

In this lab we applied the theory behind the design of experiments (DoE) to generate an appropriate data set of helicopter models and measure the resulting falling times for them. The objective was to experiment with models properly filling the filling space properties.

Choosing the best design of experiment. Filling space properties for a DoE were measured using several measures that were defined and explained in the lecture sessions as follows:

MaxiMin: The minimum distance between two points should be large, then the larger the measure the better the DoE.



(a) Predictions for model trained with 150 samples.

(b) Predictions for model trained with 30 samples.

Figure 8: GP predictions for data sampled from $f(x) = \sin(3\pi x)$ of 20 points (in red) using a *Matern32* kernel.

MiniMax: The maximum distance between any point in the space and a point in DoE should be small, hence, the smaller the measure the better the DoE.

Discrepancy: The less the discrepancy the better the DoE, converging to the uniform distribution.

The methods for using *latin hypercube* and *k-means* were implemented to generate the experiments. Their results were compared with a random sampling using discrepancy methods and visual inspection. Table 1 presents the quality evaluation for each DoE. For our setting, the latin hypercube exhibited the best behavior ranking on the second place for all the criteria. The random approach obtained the best evaluations on discrepancy and *MaxiMin* but got the worst rank for *MiniMax*.

DoE	Discrepancy	MaxiMin	MiniMax
Random	0.122	0.542	0.662
Latin Hypercube	0.2	0.527	0.611
K-means	0.527	0.424	0.474

Table 1: DoE quality evaluation

K-means outperforms the latin hypercube method on the MiniMax criterion. However, according to figures 9a and 9b, latin hypercube has a better projection capacity than k-means. In consequence we chose the latin hypercube as the method for the DoE.

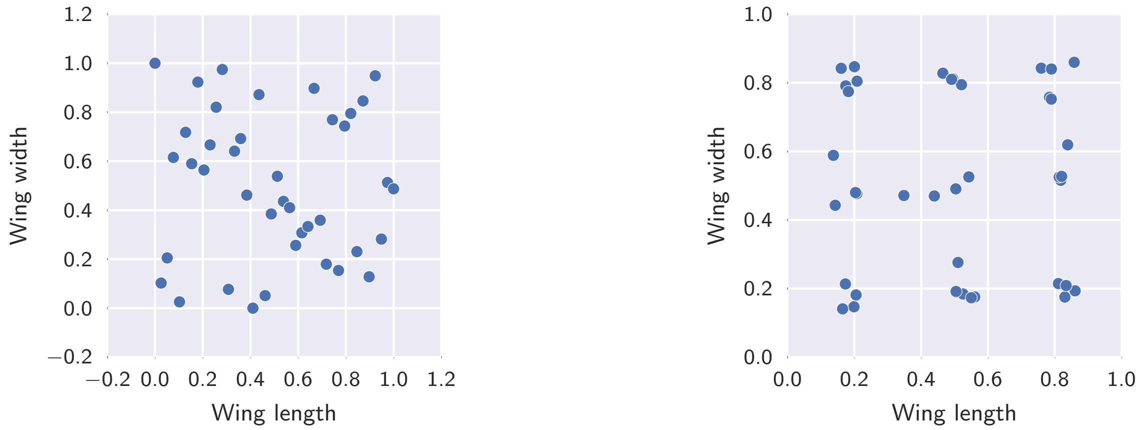
After this process it was necessary to transform the DoE to the printing space in which it had to fulfill several constraints to be printed.

4 Lab session 3

The latin latin hypercube method was used to generate 40 models of the helicopter experiment. Each helicopter was thrown 2 times from 5 to 6 meters height. Two falling times for each throw were registered in seconds by different observers.

4.1 Analyzing Data

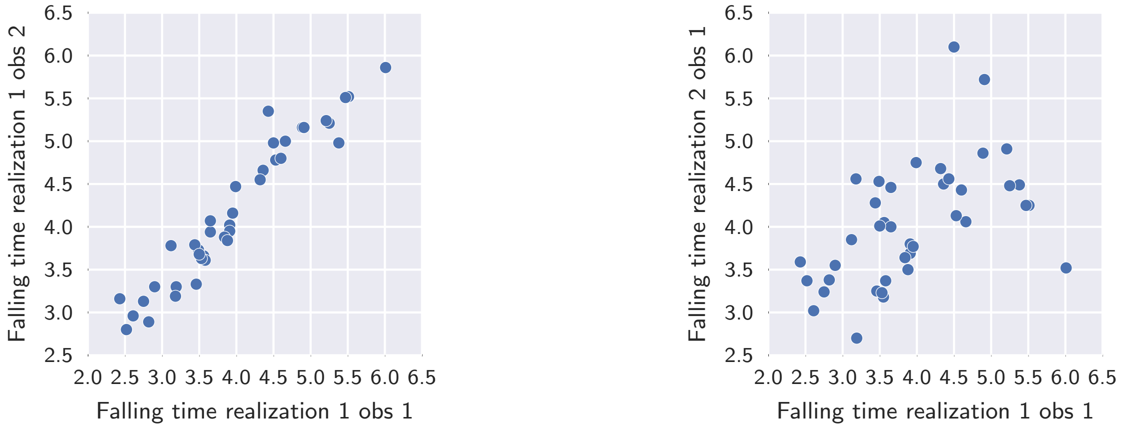
The collected data was further analyzed to inspect the observation noise and possible correlations among the model variables. Figures 10a and 10b presents the correlations between the falling times registered by the two observers for every throw. There is no much noise between the observers of each throw. However there is considerable noise between the different throws. Despite of that, in a first setting a Gaussian Process model was fitted using only two falling times corresponding to the average times of the two throws.



(a) Wing length vs. wing width for the latin hypercube

(b) Wing length vs. wing width for k-means.

Figure 9: Plots of DoE projected in a 2D space.



(a) Comparison between the two times registered for one throw.

(b) Comparison between the two times from different throws.

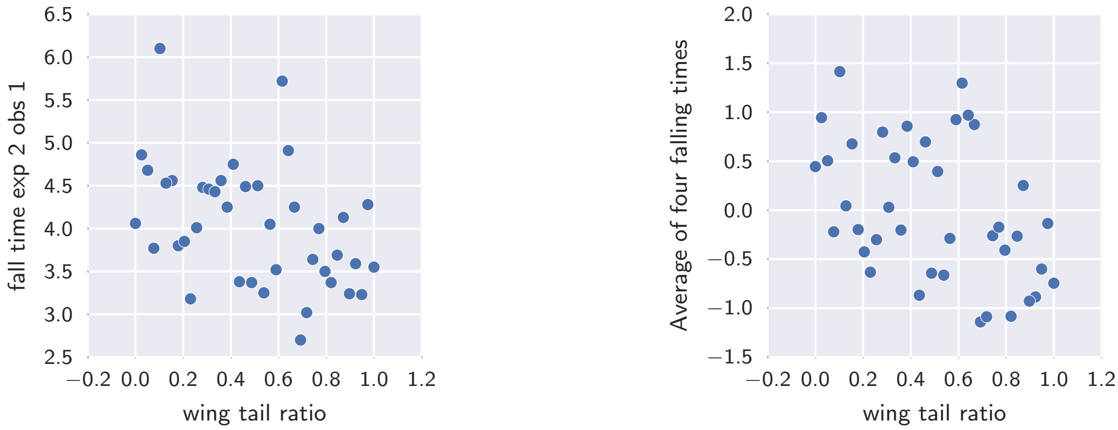
Figure 10: Plots of time measurements made by two observers(obs)

In that setting the leave one out function had to be modified in order to retrieve the two observations for the same set of parameters. The power prediction of the model was computed by a $Q2$ quality measure. The result was around 0.1 which is pretty bad as the value should be above 0.5 and is better when close to 1.0.

The conclusion on the obtained value for $Q2$ was that the gathered data was very noisy and did not expose the correlations between the model variables and the falling time. Some prior conclusions from a former lab exposed an increasing linear correlation between the wing angle and the falling time. This kind of correlation was also present with the wing length to tail ratio. In figure 11a the linear correlation between the ratio and the falling time is not as clear as in figure 11b. While in the first figure only the measurement of one observer was plotted in the second figure the average of all four measurements was taken, exposing a better correlation, as expected.

4.2 Fitting a GP model with the average

Another Assumption made to explain the $Q2$ value was the simplicity of the model, initially a GP with a Matern32 kernel was used. Thus, after modifying the time variable to be the average of the four measurements and adding some other 2 RBF kernels the $Q2$ measure increased in 0.4 giving exactly a value of 0.58 approximately. The $Q2$ measure was enough for us to follow with the optimization stage. With the new

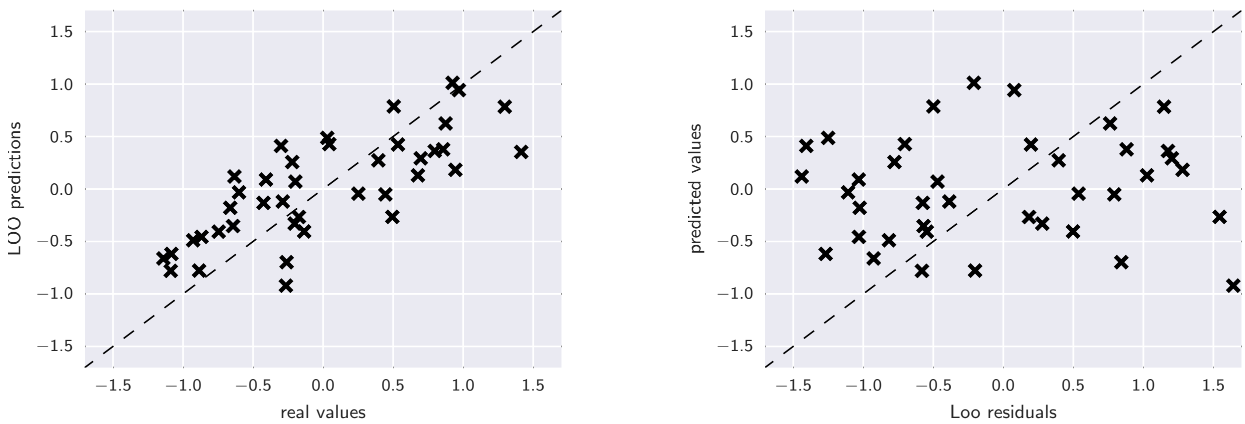


(a) Wing length to tail ratio against one of four falling times.

(b) Wing length to tail ratio against average of four falling times.

Figure 11: Plots of time measurements given the wing tail ratio.

data was not necessary to modify the leave one out function to delete the repeated point with the other time observation. In the figure 12a a plot is made showing the real falling times against the GP predictions, following a linear tendency which also is an evidence that the model is suitable to make predictions.



(a) Actual falling time values vs. *loo* predictions.

(b) *Loo* residuals vs. mean predicted value.

For further model validation the standardized residuals of the leave one out predictions where also computed and plotted against the mean predicted value(standardized for this case), the plot shows something not very similar to a linear tendency according to a normal distribution $\mathcal{N}(0, 1)$ as will be expected for better fitted models.

5 Lab session 4

The previous Lab session was about dealing with GP regression and modifying the measurement variables and the model variables in order to find a GP model suitable to explain data. In this Lab session the objective was to take that trained GP model and optimizing it with one or a mixture of the techniques reviewed in class. In general the optimization process of the trained model can be resumed as follows:

1. Generate N ($N = 1000$ for this case) random points $H = \{h_1, \ldots, h_N\}$ of your input variables, which can be between some interval in which the optimal value could be found or in all input space.

2. It is important to multiply the falling times by -1 because the optimization is a minimization and not a maximization.

3. Take each sample point h_i and compute a function to evaluate the improvement $I(h_i)$ of this sample over the best possible time value in the model observations. This can be done in two ways, either using an optimization process to find the value that maximizes the improvement function I from h_i or just evaluating $I(h_i)$.

4. from all the N points choose the point h_i with the best value for I.

5. The chosen point represents a new helicopter to be printed, then this new helicopter must be throw four times (in our setting).

6. After throwing the helicopter, the new data has to be incorporated to the model observations in order to train the model again.

7. Go to step one to start again until the flight time of the helicopter is considered to be acceptable or before running out of time.

5.1 Improvement function I

The improvement function is a way to quantify the improvement of the optimal value of the model given a new sample from the input space, for this case the improvement function used was the expected improvement which is a model based optimization method, suitable for the GP model trained for this case. The idea behind the expected improvement is to find the point which maximizes the next function:

$$EI(x) = \sqrt{c(x,x)}(u(x)cdf(x) + pdf(u(x))), \text{ where } u(x) = \frac{min(F) - m(x)}{\sqrt{c(x,x)}}$$

5.2 Optimization Results

Given that the time for running the optimization was limited, only one optimized helicopter was printed and tested. Other implementation issue was found in the Expected Improvement function optimization step, which did not allow the *EI* to be fully optimized in order to get a better estimate, so the printed helicopter was subject to that matter. Taking the trained model and applying the method explained before a new optimal helicopter was found with the following dimensions:

$$X^* = (wl = 6.500, ww = 4.453, tl = 7.952, al = 12.472)$$

The dimensions given for the new optimal helicopter made sense with some prior assumptions, for example that a big arm length means a bigger wing angle which makes an helicopter suitable to fly longer, however the wing length to tail ratio is close to one, which does not give a good flying time according to figure 11b. Nevertheless the printed helicopter was thrown from the same place where the observed data was gathered.

In average the new optimal helicopter took 5.025 seconds to touch the floor, which is better than all the observed data except for one which had a falling time of 5.4625, the reason for this is that the conditions in which the new optimal helicopter were different as the conditions of the gathered data set, besides the measurement error between realizations also affects the measurement given that the average of four measurements was took. Despite of that fact the trained model predicts a falling time for the new point X^* which is the best falling time on the dataset.