ELEN0062
Introduction to machine learning
Project 2 : Bias and variance analysis

Prof. Pierre Geurts
Prof. Louis Wehenkel

Antoine Detry
Antoine Leroy

Academic year 2019-2020

17 November 2019

1 Bayes model and residual error in classification

1.1 Bayes model corresponding to the zero-one error loss

It is asked to find an analytical formulation of the Bayes model $h_b(x_0, x_1)$ of a binary classification problem with an output $y \in \{1, +1\}$. From the theoretical course, we have :

$$h_b(x_0, x_1) = h_b(\underline{x}) = \underset{c}{\operatorname{argmax}}\, P(y = c|\ \underline{x}) \tag{1}$$

From the statement, we can write the following conclusions about the samples x_0 and x_1 :

$$x_0^i = r^i \cos\alpha^i$$
$$x_1^i = r^i \sin\alpha^i$$

Where $r \sim \mathcal{N}(R,\, \sigma^2)$ and R is a mean which is equal to R^+ if $y = 1$ or R^- if $y =$ -1, $\sigma^2 = 0.1$ is the variance and $\alpha^i \sim \mathcal{U}(0,\, 2\pi)$. Moreover, the probability of the output y for the two classes is the same :

$$P(y = 1) = P(y = -1) = \frac{1}{2}$$

Moreover, the bayes model can be written as the following :

$$h_b(x_0, x_1) = \underset{c}{\operatorname{argmax}}\, P(y = c|\ x_0, x_1) \tag{2}$$
$$= \underset{c}{\operatorname{argmax}}\, \frac{P(y = c,\ \ x_0, x_1)}{P(x_0, x_1)} \tag{3}$$
$$= \underset{c}{\operatorname{argmax}}\, \frac{\Pr(y = c)\Pr(x_0, x_1|y = c)}{\sum_{j\in\{-1,1\}} P(x_0, x_1|\, y = j)P(y = j)} \tag{4}$$
$$= \underset{c}{\operatorname{argmax}}\, \frac{\Pr(x_0, x_1|y = c)}{\sum_{c\in\{-1,1\}} P(x_0, x_1|\, y = c)} \tag{5}$$

However, it will be easier to perform the conditional probability after performing a substitution in polar coordinates r and α :

$$\sqrt{{x_0^i}^2 + {x_1^i}^2} = r^i$$
$$\arctan x_1^i/x_0^i = \alpha^i$$

It can be seen now that only r will be dependant of the value of y as α is uniform in the polar coordinates. However, in order to remain consistent, we must transform the probabilities by considering the Jacobian matrix J $= \begin{bmatrix} \cos\alpha & -r\sin\alpha \\ \sin\alpha & r\cos\alpha \end{bmatrix}$. As a result, we can now transform the probabilities :

$$\begin{aligned}\Pr(x_0, x_1|y = c) &= \frac{1}{|J|}\Pr(r, \alpha|y = c) \\ &= \frac{1}{r}\Pr(\alpha)\Pr(r|y = c) \\ &= \frac{1}{2\pi\, r}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{r-\mu_c}{\sigma})^2}\end{aligned}$$

In addition, we can also modify equation (5).

$$h_b(x_0, x_1) = \underset{c}{\operatorname{argmax}} \frac{\Pr(x_0, x_1 | y = c)}{\sum_{c\in\{-1,1\}} P(x_0, x_1 |\, y = c)} \tag{6}$$

$$= \underset{c}{\operatorname{argmax}} \frac{\Pr(r, \alpha | y = c)}{\sum_{c\in\{-1,1\}} P(r, \alpha |\, y = c)} \tag{7}$$

$$= \underset{c}{\operatorname{argmax}} \frac{\frac{1}{2\pi\, r}\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{r-\mu_c}{\sigma})^2}}{\frac{1}{2\pi\, r}\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{r-R^+}{\sigma})^2} + \frac{1}{2\pi\, r}\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{r-R^-}{\sigma})^2}} \tag{8}$$

$$= \underset{c}{\operatorname{argmax}} \frac{e^{-\frac{1}{2}(\frac{r-\mu_c}{\sigma})^2}}{e^{-\frac{1}{2}(\frac{r-R^+}{\sigma})^2} + e^{-\frac{1}{2}(\frac{r-R^-}{\sigma})^2}} \tag{9}$$

We can conclude from equation (9) that, for example, $y = 1$ if :

$$e^{-\frac{1}{2}(\frac{r-R^+}{\sigma})^2} \geq e^{-\frac{1}{2}(\frac{r-R^-}{\sigma})^2}$$

Which can be simplified to :

$$r \geq \frac{R^+ + R^-}{2} \text{with } R^+ \geq R^- \tag{10}$$

Finally we have the following :

$$\begin{cases} h_b(x_0, x_1) = 1 \text{ if } r \geq \frac{R^+ + R^-}{2} \text{ and if } R^+ \geq R^- \\ h_b(x_0, x_1) = -1 \text{ Otherwise.} \end{cases}$$

As a conclusion, the Bayes model classifies the value of the output y to 1 if the radius is greater than or equal to the average of the expected radius for each class, otherwise it classifies y as -1.

1.2 Analytical formation of the residual error

The estimation of the residual error can be also computed by doing a change of variable in polar coordinates :

$$\begin{aligned} E_{x_0,x_1,y}\{1(y \neq h_b(x_0, x_1))\} &= E_{x_0,x_1,y}\big\{1(y \neq h_b(x_0, x_1))|y = 1\big\}P(y = 1) \\ &+ E_{x_0,x_1,y}\big\{1(y \neq h_b(x_0, x_1))|y = -1\big\}P(y = -1) \\ &= \frac{1}{2}\Bigg[E_{x_0,x_1,y}\{1(y \neq h_b(x_0, x_1))|y = 1\} \\ &+ E_{x_0,x_1,y}\{1(y \neq h_b(x_0, x_1))|y = -1\}\Bigg] \\ &= \frac{1}{2}\Bigg[\int_{-\infty}^{\infty} d\alpha \int_{-\infty}^{\infty} f(r, \alpha|y = 1)|J|dr \\ &+ \int_{-\infty}^{\infty} d\alpha \int_{-\infty}^{\infty} f(r, \alpha|y = -1)|J|dr\Bigg] \end{aligned}$$

The probability densities of the variable x_0 and x_1 are :

$$\begin{cases} f(r, \alpha|y = 1) = \frac{1}{2\pi r}\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{r-R^+}{\sigma})^2} \\ f(r, \alpha|y = -1) = \frac{1}{2\pi r}\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{r-R^-}{\sigma})^2} \end{cases}$$

Thanks to the 1.1 subsection above, the intervals of r when y = 1 and y = -1 are respectively :

$$r \in [\frac{(R^+ + R^-)}{2}, \infty[$$
$$r \in]-\infty; \frac{(R^+ + R^-)}{2}[$$

Finally, the residual errors are detected by inverting the boundaries for each class and by taking $R^+ = 2$ and $R^- = 1$:

$$E_{x_0,x_1,y}\{1(y \neq h_b(x_0,x_1))\} = \frac{1}{4\pi\sigma\sqrt{2\pi}}\Bigg[\int_0^{2\pi} d\alpha \int_{-\infty}^{\frac{(R^+ + R^-)}{2}} e^{-\frac{1}{2}(\frac{u_0 - R^+}{\sigma})^2} dr + \int_0^{2\pi} d\alpha \int_{\frac{(R^+ + R^-)}{2}}^{\infty} e^{-\frac{1}{2}(\frac{u_0 - R^-}{\sigma})^2} dr\Bigg] = 0.0569$$

2 Bias and variance of the kNN algorithm

2.1 Generalization error decomposition

We know that in regression problems, the goal is to find an estimation $\hat{y}$(x) such that the following expression is minimized. For more simplicity in the following, we write $\hat{y}(x)$ instead of the expression $\hat{y}$(x ;LS,k).

$$E_{y,x}\{(y - \hat{y}(x))^2)\}$$

A good learning algorithm should be good in average over all learning samples of size N. Thus, in practice, we want to minimize the test error expression below. This represents the test error at some point x for LS of size N with a prediction $\hat{y}$(x) for a KNN-regression with k neighbours.

$$E_{LS}\{E_{y|x}\{(y - \hat{y}(x))^2\}$$

As explained, this represents the expected prediction error. It can be decomposed in $Bias^2$, $Variance$ and an $Irreductibe\ error$ terms.

$$Var_{y|x}(y) + Bias^2_{y|x}(\hat{y}) + Var_{LS}(\hat{y}(x)) \tag{11}$$

This expression can be rewritten thanks to the Bayes model notation and transformations on the different terms.

We consider that the input values x^i are fixed and only y is random. We write $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\, \sigma^2)$. With these assumptions, we can assume the following points :

$$\begin{aligned}
E\{f(x)\} &= f(x) \\
Var\{f(x)\} &= 0 \\
E\{\epsilon\} &= 0 \\
E\{\epsilon^2\} &= \sigma^2 = Var\{\epsilon\} \\
\hat{y}(\underline{x}) &= \frac{1}{k}\sum_{l=1}^{k} y(x_l) = \frac{1}{k}\sum_{l=1}^{k}(f(x_l) + \epsilon_l) \\
Cov\{f(x), \epsilon\} &= E\{(f(x) - E\{f(x)\})(\epsilon - E\{\epsilon\})\} = 0
\end{aligned}$$

First, the Bayes model can be determined as followed.

$$\begin{aligned} h_b(x) &= E_{y|x}\{y\} \\ &= E_{\epsilon|x}\{(f(x)+\epsilon)\} \\ &= f(x) \end{aligned}$$

Secondly, we can define the *noise* as followed.

$$\begin{aligned} noise(x) &= var_{x|y}\{y\} \\ &= var\{(f(x)+\epsilon)\} \\ &= var\{f(x)\} + var\{\epsilon\} + 2f(x)\epsilon \times Cov\{f(x),\epsilon\} \\ &= 0 + \sigma^2 + 0 \\ &= \sigma^2 \end{aligned}$$

Third, we can write the $Bias^2$ as followed.

$$\begin{aligned} Bias^2(\underline{x}) &= (E_{x|y}\{y\} - E_{LS}\{\hat{y}(x)\})^2 \\ &= (E_{x|y}\{f(x)\} + 0 - E_{LS}\{\frac{1}{k}\sum_{l=1}^{k} f(x_i)\} - 0)^2 \\ &= (f(x) - \frac{1}{k}\sum_{l=1}^{k} f(x_i))^2 \end{aligned}$$

Finally, the variance term can be rewritten as followed.

$$\begin{aligned} variance\{\underline{x}\} &= var_{LS}\{\hat{y}(\underline{x})\} \\ &= var\{\frac{1}{k}\sum_{l=1}^{k}(f(x_i)+\epsilon_i)\} \\ &= \frac{1}{k^2}\sum_{l=1}^{k}(var\{\epsilon_i\}+0) \\ &= \frac{1}{k^2}(k\sigma^2) \\ &= \frac{\sigma^2}{k} \end{aligned}$$

Gathering all the terms,the generalization error of the k-Nearest Neighbours algorithm at some point x can be decomposed as

$$\sigma^2 + (f(x) - \frac{1}{k}\sum_{l=1}^{k} f(x_l))^2 + \frac{\sigma^2}{k} \tag{12}$$

2.2 Effect of k on each term of the bias-variance decomposition

As developed in the previous subsection, the test error can be decomposed in three terms.

The $Bias^2_{y|x}(\hat{y})$ reflects how the estimate varies from the mean. When the bias is high, the algorithm miss the relevant relations between inputs and output. The $Var_{y|x}(\hat{y})$ reflects the expected deviation around the mean of $\hat{y}$. When the variance is high, the model is not good in generalizing on the data because the model is overfitting. Then, the σ^2_ϵ, i.e the noise, is a

random error we can't minimize, no matter how well we estimate f(x).

As the noise is an irreductible error, what is interesting to discuss is the effect of the number of neighbours k on the bias and variance terms. Indeed, in order to minimize the test error, we have to deal with these two terms and find a trade-off between them.

Dealing bias and variance terms is about dealing with overfitting and underfitting the model. The more complex is a model, the more the variance increases and the more the bias decreases. In point of fact, when a model is more complex, it has more power to capture all data points but has thus a high variance. A simplest model is not good in capturing data points but has thus a high bias.

In the case of the KNN-algorithm, we can say that the more k increases, the more the bias increases and the more the variance decreases. Conversely, the more k decreases, the more the bias decreases and the more the variance decreases.

Let's take the (12) expression to pratically observe the effect of a k variation. We can see that k will affect the $bias^2$ and $variance$ terms. If k increases, the expression of the $Variance = \frac{\sigma^2}{k}$ decreases while the expression of the $Bias^2 = (f(x) - \frac{1}{k}\sum_{l=1}^{k} f(x_i))^2$ increases.

3 Bias and variance estimation

3.1 Protocol to estimate the residual error, the squared bias, and the variance

We consider a regression problem for which we can generate an infinite number of samples : $(X_i, Y_i)_{i=1}^{\infty}$. Let's take $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\, \sigma^2)$ is a random noise, $f(x)$ is the learning model and y is the output. The first operation is to create a certain number of datasets with a certain size uniformly in the interval of the samples and add at the end a random noise independent of the system. The number and size of each created datasets is arbitrary but it must be a good generalization of all possible datasets. Once the set of datasets are created, they are each used to train an estimator model of a given supervised learning algorithm.

When all models are trained, we can now estimate the residual error, the bias squared, and the variance on a given point x_0 for each estimator. In a regression problem, we can express these three terms as in Exercise 2 :

$$\begin{aligned} Var_{y|x}\{y\} &= E_{y|x}\{(y - E_{y|x}\{y\})^2\} = \sigma^2 \\ Bias^2_{y|x}\{\hat{y}\} &= (E_{y|x}\{y\} - E_{LS}\{\hat{y}(x)\})^2 \\ Var_{LS}\{\hat{y}(x)\} &= E_{LS}\{(\hat{y}(x) - E_{LS}\{\hat{y}(x)\})^2\} \\ E_{y|x}\{y\} &= E_{\epsilon|x}\{f(x) + \epsilon\} = f(x) + 0 \end{aligned}$$

We can notice that the residual error does not depend on the dataset but only on the variance of the noise.

3.2 Protocol to estimate the mean values of the residual error, the squared bias and the variance.

As we know the protocol to compute the squared bias, the variance and the residual error at a given point x_0, we can evaluate the mean of each term in considering a large dataset of

inputs values.

As the error at a given point, for inputs $\underline{x}$ is :

$$E_{x,y}\{(y - \hat{y}(x))^2\}$$

The error over all learning sets is :

$$\begin{aligned} E &= E_{LS}\{E_{x,y}\{(y - \hat{y}(\underline{x}))^2\}\} \\ &= E_{\underline{x}}\{E_{LS}\{E_{y|x}\{(y - \hat{y}((\underline{x}))^2\}\}\} \\ &= E_{\underline{x}}\{var_{y|x}\{y\}\} + E_{\underline{x}}\{bias^2(\underline{x})\} + E_{\underline{x}}\{var_{ls}\{\hat{y}(\underline{x})\}\} \end{aligned}$$

In practice, a good idea if we want to compute the means over the whole input space is to take envenly spaced inputs values over the interval of $\underline{x}$ to be able to cover the entire inputs space.

3.3 Protocols for a finite number of samples

In case we only have a fixed, finite number of samples, the protocol defined above is still valid. Indeed, instead of generating a finite samples randomly, we just have to take into account the finite set of N samples to train and test the models.

3.4 Estimations and plots of the residual error, the squared bias, the variance and the expected error

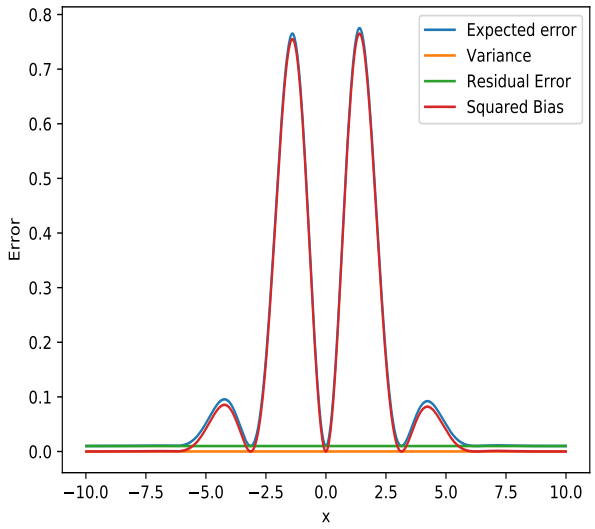

FIGURE 1 – The error of the Lasso model with $\alpha = 8$

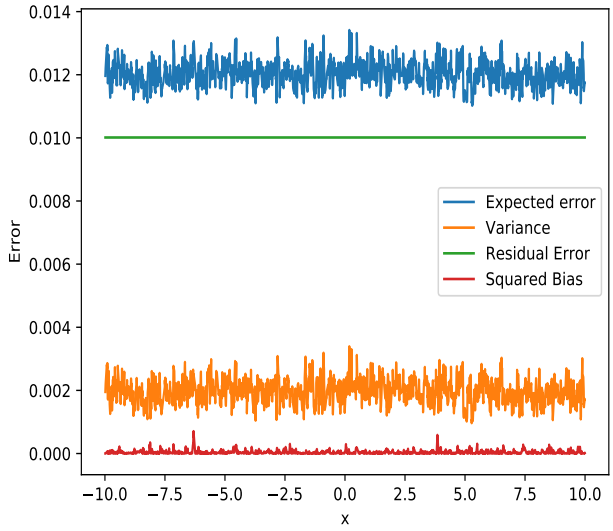

FIGURE 2 – The error of the Knn model with $K = 8$

We can use the protocols to display the estimation of residual error, squared bias and variance on two regression methods. The linear regression is the Lasso method and the linear regression will be the Knn method each over 50 datasets of 2000 features. The latter will make it possible to compare the results of question 2. We will also display the changes provided by the addition of irrelevant variables. In the context of this question, we set the alpha parameter

of the Lasso method to 8 and the number of neighbours for the Knn method to 5. Normally, we will have to observe a large error for the lasso no matter how complex it is, while the Knn errors should vary according to its complexity. In addition, we can consider the residual error as constant at 0.01, as demonstrated above.

In the case of the Lasso regression algorithm, the error is largely represented by the squared bias, which is normal since a linear regression is not at all appropriate for the output function y. On the other hand, the variance is a line almost equal to 0, so this implies that we are well in the underfitting case.

On the other hand, we can notice in the Knn method with K = 5, unlike the Lasso linear regression, that the bias is small but the variance oscillates enormously around the $\frac{\sigma^2}{k}$ value. So we are in the case of overfitting here. In addition, we can vary the number of neighbours in this algorithm, and we can see that the bias increases as the variance decreases and stabilizes more and more close to 0 as k increases. This is therefore in accordance with the explanations in question 2. Finally, we can also notice that when k is large, around 500, the error graph is almost equivalent to the graph of the Lasso linear regression algorithm. This means that above a certain k_t threshold, the Knn method acts as a linear regression method.

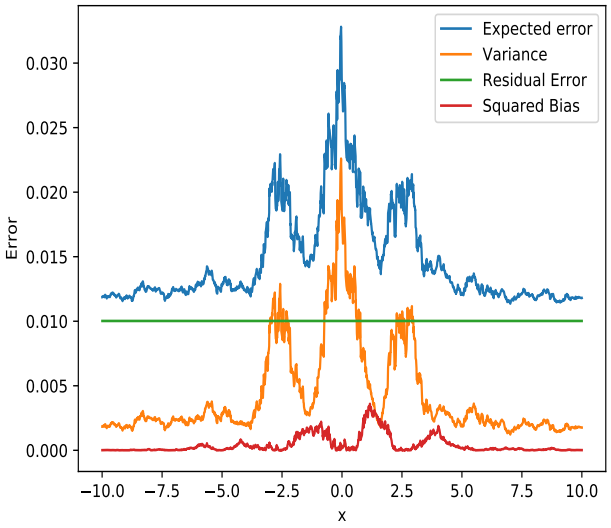

FIGURE 3 – Knn with $k = 5$ and only one irrelevant variable

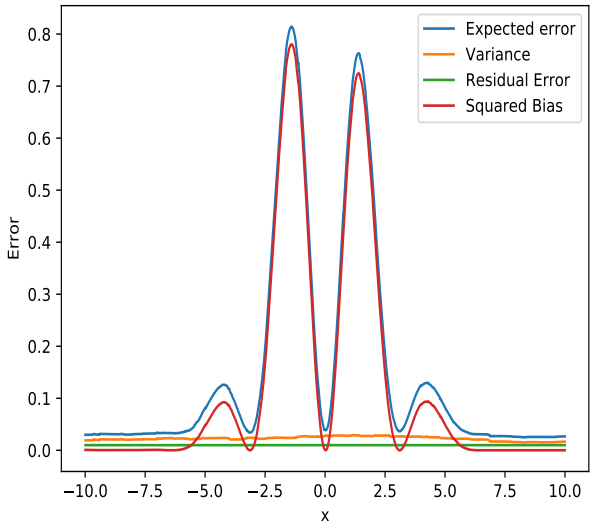

FIGURE 4 – Knn with $k = 5$ and with 100 irrelevant variable added

Finally, we can look at the influence of the number of irrelevant variables on the KNN model. We notice that the error is greater in this situation. In addition, when the number of irrelevant variables is large, the shape of the square bias and error are equivalent to that of a linear regression. This is due to the fact that models are more and more finding false relationships between relevant and non-relevant variables.

3.5 Estimations and plots of the mean values of the residual error, the squared bias, the variance and the expected error

In this section, we will report the mean value of the residual error, squared bias and variance, depending on the size of the learning set, the complexity and finally the number of irrelevant

variables.

3.5.1 In function of the size of the learning set

The plots in this section are Figures 5 and 6.

First of all, we can notice that the error varies a little and stabilizes quickly in the case of linear regression. At first the variance is large and dominates in the error and then adjusts to the value of the residual error. Then it is the square bias, which is more or less constant all along the graph, that dominates in the error. To conclude, we can see that the error is large and stabilizes around 0.2.

At fixed complexity (k=10), we first notice that the error is much lower than in the case of linear regressions. We also see that it decreases rapidly with the size of the learning set and we can therefore deduce that the error depends on the size of the learning set and therefore conclude that the KNN is more accurate when the learning set is large.

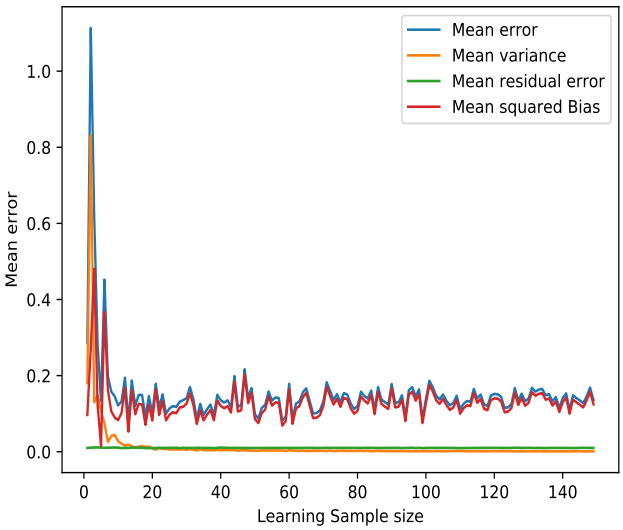

FIGURE 5 – The mean values of the Lasso model in function of the size of the learning set at fixed complexity

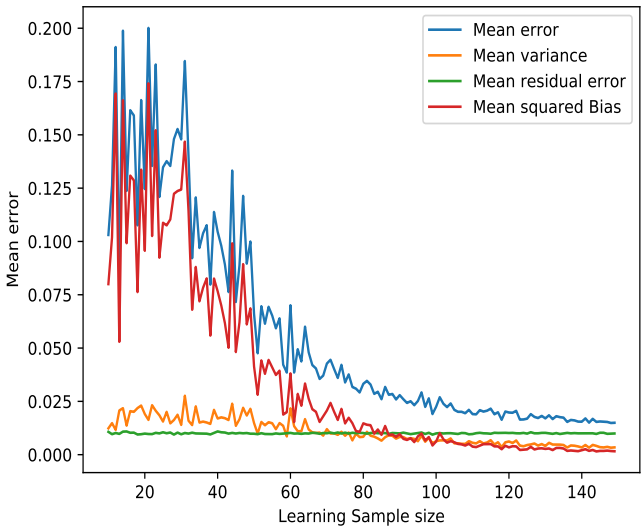

FIGURE 6 – The mean values of the Knn model in function of the size of the learning set at fixed complexity

3.5.2 In function of the model complexity

Usually, the bias decreases when the non-linear model complexity gets bigger while the variance increases.

What we can observe through the Figure 7 is that it seems that the model complexity has no clear effect on the error decomposition in a linear model. Indeed, the variance remains low while the complexity increases and the bias remains high and is the main parameters in the total mean error.

Regarding the Figure 8, analyzing the effect of complexity on a KNN model, we clearly observe that now, the more the complexity of the model increases (i.e the more the number

k of neighbors decreases, left part of the plot), the more the variance increases and the bias decreases. What is interesting to notice is the fact that it seems that increasing the complexity is relevant until a certain threshold for which the total error is minimized.

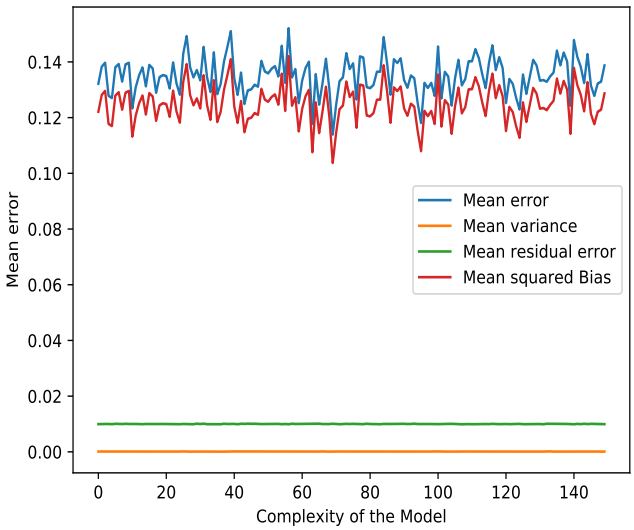

FIGURE 7 – The mean values of the Lasso model in function of the complexity model

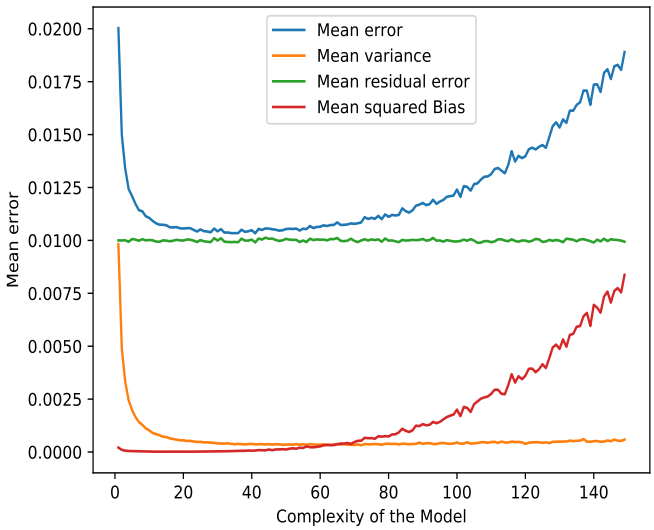

FIGURE 8 – The mean values of the Knn model in function of the complexity model

3.5.3 In function of the number of irrelevant variables added

We can look at Figure 9 and observe that in linear regressions, it seems that when we add irrelevant inputs to our model, the total error is not affected. The variance term remains low in average while the variations of the bias is independant of the irrelevant variables.

The figure 10 showing the relations between the error and added irrelevant variables in non-linear regressions seems to tell us that adding some irrelevant variables increases a lot the bias term. An intuition to explain this phenomenon could be that the model forces to find a relation between outputs and new inputs and thus don't set their coefficient to 0. In this way, the model miss to find the most relevant relations between relevant inputs and outputs.

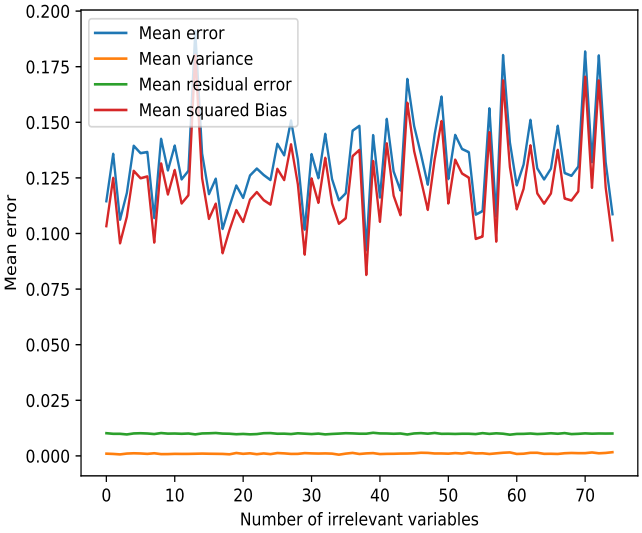

FIGURE 9 – The mean values of the Lasso model in function of the number of irrelevant variables added at fixed complexity

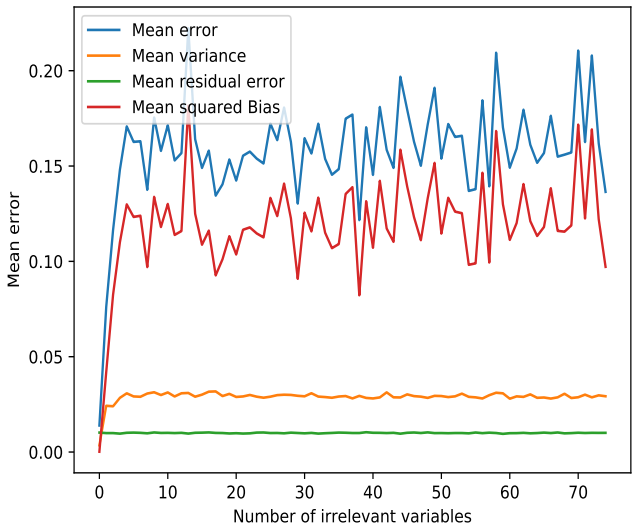

FIGURE 10 – The mean values of the Knn model in function of the number of irrelevant variables added at fixed complexity