LIÈGE université

Université de Liège

ELEN0062-1
Introduction to machine learning

Project :
Bias and variance analysis

DEBOR Antoine
HELD Jan

Academic year 2020-2021

1 Analytical derivations

1.1 Bayes model and residual error in classification

a. First of all, the Bayes model is given by : $h_b(\underline{x}) = h_b(x_0, x_1) = \arg\max_c P(y = c|x)$. Furthermore, the statement says that the output y is either $+1$ or -1. Hence, the Bayes model can be rewritten as :

$$h_b(x_0, x_1) = \begin{cases} +1 \ \ if \ \ P(y = 1|\underline{x}) > P(y = -1|\underline{x}) \\ -1 \ \ otherwise \end{cases} \tag{1}$$

By using the Bayes theorem, one can derive the following equations :

$$P(y = 1|\underline{x}) = \frac{P(\underline{x}|y = 1)P(y = 1)}{P(\underline{x})}$$

$$P(y = -1|\underline{x}) = \frac{P(\underline{x}|y = -1)P(y = -1)}{P(\underline{x})}.$$

The first condition in equation 1 can now be rewritten as :

$$P(y = 1|\underline{x}) > P(y = -1|\underline{x})$$

$$\Leftrightarrow \frac{P(\underline{x}|y = 1)P(y = 1)}{P(\underline{x})} > \frac{P(\underline{x}|y = -1)P(y = -1)}{P(\underline{x})}.$$

The classes are randomly chosen with an equal probability and therefore $P(y = 1) = P(y = -1) = \frac{1}{2}$. And finally, by simplifying $P(\underline{x})$ on both sides, one obtains

$$P(\underline{x}|y = 1) > P(\underline{x}|y = -1). \tag{2}$$

By using the multivariate Gaussian distribution, equation 2 can be rewritten as

$$\frac{1}{2\pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}-\underline{u_1})^T\Sigma_1^{-1}(\underline{x}-\underline{u_1})} > \frac{1}{2\pi|\Sigma_{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}-\underline{u_{-1}})^T\Sigma_{-1}^{-1}(\underline{x}-\underline{u_{-1}})}, \tag{3}$$

with

$$\underline{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \qquad \underline{u_1} = \underline{u_{-1}} = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 1 & p^+ \\ p^+ & 1 \end{bmatrix} \qquad \Sigma_{-1} = \begin{bmatrix} 1 & -p^+ \\ -p^+ & 1 \end{bmatrix}. \tag{4}$$

One can now calculate

$$|\Sigma_1| = |\Sigma_{-1}| = 1 - p^{+2} \qquad \Sigma_1^{-1} = \frac{1}{1 - p^{+2}} \begin{bmatrix} 1 & -p^+ \\ -p^+ & 1 \end{bmatrix} \qquad \Sigma_{-1}^{-1} = \frac{1}{1 - p^{+2}} \begin{bmatrix} 1 & p^+ \\ p^+ & 1 \end{bmatrix}. \tag{5}$$

By inserting terms of 4 and 5 in inequation 3, one obtains

$$e^{-\frac{1}{2}(\underline{x})^T\Sigma_1^{-1}(\underline{x})} > e^{-\frac{1}{2}(\underline{x})^T\Sigma_{-1}^{-1}(\underline{x})}$$

$$\Leftrightarrow (\underline{x})^T\Sigma_1^{-1}(\underline{x}) > (\underline{x})^T\Sigma_{-1}^{-1}(\underline{x})$$

$$\Leftrightarrow x_0 * x_1 > 0$$

Finally, statement 1 can be rewritten as :

$$\boxed{h_b(x_0, x_1) = \begin{cases} +1 \ if \ \ x_0 * x_1 > 0 \\ -1 \ otherwise \end{cases}} \tag{6}$$

b. The residual error, *i.e.* the generalization error of the previously derived Bayes model $h_b(x_0, x_1)$, is theoretically given by

$$E_{x_0,x_1,y} = \{1(y \neq h_b(x_0, x_1)\}. \tag{7}$$

Considering the definition of the mathematical expectation, equation 7 can be developed as

$$E_{x_0,x_1,y} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P((x_0, x_1) \cap y).1(h_b(x_0, x_1) \text{ is wrong}) dx_0 dx_1, \tag{8}$$

which can be written as

$$E_{x_0,x_1,y} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P((x_0, x_1)|y).P(y).1(h_b(x_0, x_1) \text{ is wrong}) dx_0 dx_1 \tag{9}$$

by using the Bayes formula. One can then separate equation 9 into two parts since $h_b(x_0, x_1)$ is wrong in two distinct cases :

$$\begin{aligned} E_{x_0,x_1,y} = & \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P((x_0, x_1)|y = 1).P(y = 1).1(h_b(x_0, x_1) = -1) dx_0 dx_1 \\ & + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P((x_0, x_1)|y = -1).P(y = -1).1(h_b(x_0, x_1) = 1) dx_0 dx_1, \end{aligned} \tag{10}$$

and, considering the definition of $h_b(x_0, x_1)$, the integration intervals can be reduced as follow :

$$\begin{aligned} E_{x_0,x_1,y} = & \int_0^{+\infty} dx_0 \int_{-\infty}^{0} P((x_0, x_1)|y = 1).P(y = 1) dx_1 \\ & + \int_{-\infty}^{0} dx_0 \int_0^{+\infty} P((x_0, x_1)|y = 1).P(y = 1) dx_1 \\ & + \int_0^{+\infty} dx_0 \int_0^{+\infty} P((x_0, x_1)|y = -1).P(y = -1) dx_1 \\ & + \int_{-\infty}^{0} dx_0 \int_{-\infty}^{0} P((x_0, x_1)|y = -1).P(y = -1) dx_1. \end{aligned} \tag{11}$$

Finally, recalling that

$$P((x_0, x_1)|y = 1) = \frac{1}{2\pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T \Sigma_1^{-1} \underline{x}},$$

$$P((x_0, x_1)|y = -1) = \frac{1}{2\pi|\Sigma_{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T \Sigma_{-1}^{-1} \underline{x}}$$

and

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

with $|\Sigma_1| = |\Sigma_{-1}| = 1-\rho^+$, $\Sigma_1^{-1} = \frac{1}{1-\rho^+}\begin{pmatrix} 1 & -\rho^+ \\ -\rho^+ & 1 \end{pmatrix}$ and $\Sigma_{-1}^{-1} = \frac{1}{1-\rho^+}\begin{pmatrix} 1 & \rho^+ \\ \rho^+ & 1 \end{pmatrix}$, the residual error writes

$$\boxed{\begin{aligned} E_{x0,x1,y} = &\frac{1}{2}\int_0^{+\infty} dx_0 \int_{-\infty}^{0} \frac{1}{2\pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_1^{-1}\underline{x}} dx_1 \\ &+ \frac{1}{2}\int_{-\infty}^{0} dx_0 \int_{0}^{+\infty} \frac{1}{2\pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_1^{-1}\underline{x}} dx_1 \\ &+ \frac{1}{2}\int_{0}^{+\infty} dx_0 \int_{0}^{+\infty} \frac{1}{2\pi|\Sigma_{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_{-1}^{-1}\underline{x}} dx_1 \\ &+ \frac{1}{2}\int_{-\infty}^{0} dx_0 \int_{-\infty}^{0} \frac{1}{2\pi|\Sigma_{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_{-1}^{-1}\underline{x}} dx_1. \end{aligned}} \tag{12}$$

One can also notice that this formula is symmetrical with respect to x_0 and x_1, which makes it possible to write it as

$$\begin{aligned} E_{x0,x1,y} = &\int_0^{+\infty} dx_0 \int_{-\infty}^{0} \frac{1}{2\pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_1^{-1}\underline{x}} dx_1 \\ &+ \int_{0}^{+\infty} dx_0 \int_{0}^{+\infty} \frac{1}{2\pi|\Sigma_{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}\underline{x}^T\Sigma_{-1}^{-1}\underline{x}} dx_1 \end{aligned} \tag{13}$$

dividing the integration span by two (which could be of good practice for a numerical integration).

For $\rho^+ = 0.75$, this integral is computed in the file `Q1b.m` and is equal to $E_{x_0,x_1,y} = 0.2301$. This is a coherent value since it is way smaller than 0.5, the value of the residual error if the model selects at random the output, regardless of the inputs.

Empirically, this value is computed in the file `Q1.1.b.py`. To compute $E_{x_0,x_1,y}$, one generates 100 times independently 20000 samples $\mathbf{x}^i = (x_0^i, x_1^i)$ by first selecting their class y^i at random (with an equal probability for each class), and then drawing their values from the given multivariate Gaussian distribution. Then, one computes the generalization error of the Bayes model for each of the 100 sets of samples. Finally, one takes the average of the generalization error of the Bayes model over the 100 repetitions and one gets $E_{x_0,x_1,y} = 0.23035$, which is, as expected, close to the value previously computed with the analytical formula.

1.2 Bias and variance of ridge regression

a. For the ordinary least-square regression, the vector $\boldsymbol{w}$ is defined as $\boldsymbol{w}_{OLS} = \underset{\boldsymbol{w}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^T\boldsymbol{w}\right)^2$. The argument of the min function can be vectorially written as

$$OLS(\boldsymbol{w}) = (\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) \tag{14}$$

and setting its first derivative to 0

$$\frac{\partial}{\partial \boldsymbol{w}} OLS(\boldsymbol{w}) = 2(\mathbf{X}^T\mathbf{X})\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y} = 0 \tag{15}$$

in order to find the minimum value leads to

$$\begin{aligned} 2(\mathbf{X}^T\mathbf{X})\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y} &= 0 \\ \iff (\mathbf{X}^T\mathbf{X})\boldsymbol{w} &= 2\mathbf{X}^T\boldsymbol{y} \\ \iff (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\boldsymbol{w} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}, \end{aligned} \tag{16}$$

which finally gives

$$\boldsymbol{w}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{17}$$

the optimal vector $\boldsymbol{w}$ in the ordinary least-square sense. Assuming that $\mathbf{X}$ is orthogonal, $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ and this product is thus non singular and invertible.

For the ridge regression, the vector $\boldsymbol{w}$ is defined as $\boldsymbol{w}_R = \underset{\boldsymbol{w}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^T\boldsymbol{w}\right)^2 + \lambda \boldsymbol{w}^T\boldsymbol{w}$. The argument of the min function can be vectorially written as

$$RR(\boldsymbol{w}, \lambda) = (\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w} \tag{18}$$

and setting its first derivative to 0

$$\frac{\partial}{\partial \boldsymbol{w}} RR(\boldsymbol{w}, \lambda) = 2(\mathbf{X}^T\mathbf{X})\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y} + 2\lambda\boldsymbol{w} = 0 \tag{19}$$

in order to find the minimum value leads to

$$\begin{aligned} 2(\mathbf{X}^T\mathbf{X})\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y} + 2\lambda\boldsymbol{w} &= 0 \\ \iff (\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)\boldsymbol{w} &= \mathbf{X}^T\boldsymbol{y} \\ \iff (\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)^{-1}(\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)\boldsymbol{w} &= (\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)^{-1}\mathbf{X}^T\boldsymbol{y}, \end{aligned} \tag{20}$$

which finally gives

$$\boldsymbol{w}_{RR} = (\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{21}$$

the optimal vector $\boldsymbol{w}$ in the ridge regression sense. Again, assuming that $\mathbf{X}$ is orthogonal, $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ and $(\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda)$ is thus non singular and invertible.

Assuming that $\mathbf{X}$ is orthogonal, one can write

$$\boldsymbol{w}_{OLS} = \mathbf{X}^T\boldsymbol{y}, \tag{22}$$

and

$$\boldsymbol{w}_{RR} = (\mathbf{I} + \mathbf{I}\lambda)^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{23}$$

which eventually lead to

$$\boldsymbol{w}_{RR} = \frac{1}{1+\lambda}\boldsymbol{w}_{OLS}. \tag{24}$$

b. i. The bias of $\boldsymbol{x}^T\boldsymbol{w}_{OLS}$ is given by

$$\begin{aligned} bias_{OLS}(\boldsymbol{x}) &= h_b(\boldsymbol{x}) - E_{LS}\{\hat{y}(\boldsymbol{x})\} \\ &= h_b(\boldsymbol{x}) - E_{LS}\{\boldsymbol{x}^T\boldsymbol{w}_{OLS}\} \end{aligned} \tag{25}$$

and the bias of $\frac{\boldsymbol{x}^T\boldsymbol{w}_{OLS}}{1+\lambda}$ is given by

$$\begin{aligned} bias_{pseudo-R}(\boldsymbol{x}) &= h_b(\boldsymbol{x}) - E_{LS}\{\hat{y}(\boldsymbol{x})\} \\ &= h_b(\boldsymbol{x}) - E_{LS}\{\frac{\boldsymbol{x}^T\boldsymbol{w}_{OLS}}{1+\lambda}\} \\ &= h_b(\boldsymbol{x}) - \frac{1}{1+\lambda}E_{LS}\{\boldsymbol{x}^T\boldsymbol{w}_{OLS}\}. \end{aligned} \tag{26}$$

The relation between these two bias then writes

$$\boxed{bias_{pseudo-R}(\boldsymbol{x}) = \frac{bias_{OLS}(\boldsymbol{x})}{1+\lambda} + \frac{\lambda}{1+\lambda} h_b(\boldsymbol{x}).} \tag{27}$$

The variance of $\boldsymbol{x}^T \boldsymbol{w}_{OLS}$ is given by

$$\begin{aligned} var_{OLS}(\boldsymbol{x}) &= E_{LS}\{(\hat{y}(\boldsymbol{x}) - E_{LS}\{\hat{y}(\boldsymbol{x})\})^2\} \\ &= E_{LS}\{(\boldsymbol{x}^T \boldsymbol{w}_{OLS} - E_{LS}\{\boldsymbol{x}^T \boldsymbol{w}_{OLS}\})^2\} \end{aligned} \tag{28}$$

and the variance of $\frac{\boldsymbol{x}^T \boldsymbol{w}_{OLS}}{1+\lambda}$ is given by

$$\begin{aligned} var_{pseudo-R}(\boldsymbol{x}) &= E_{LS}\{(\hat{y}(\boldsymbol{x}) - E_{LS}\{\hat{y}(\boldsymbol{x})\})^2\} \\ &= E_{LS}\{(\frac{\boldsymbol{x}^T \boldsymbol{w}_{OLS}}{1+\lambda} - E_{LS}\{\frac{\boldsymbol{x}^T \boldsymbol{w}_{OLS}}{1+\lambda}\})^2\} \\ &= E_{LS}\{(\frac{1}{1+\lambda}(\boldsymbol{x}^T \boldsymbol{w}_{OLS} - E_{LS}\{\boldsymbol{x}^T \boldsymbol{w}_{OLS}\}))^2\} \\ &= \frac{1}{(1+\lambda)^2} E_{LS}\{(\boldsymbol{x}^T \boldsymbol{w}_{OLS} - E_{LS}\{\boldsymbol{x}^T \boldsymbol{w}_{OLS}\})^2\}. \end{aligned} \tag{29}$$

The relation between these two variances then writes

$$\boxed{var_{pseudo-R}(\boldsymbol{x}) = \frac{1}{(1+\lambda)^2} var_{OLS}(\boldsymbol{x}).} \tag{30}$$

ii. The impact of λ on bias and variance can be explained on the basis of these formulas by setting λ to extreme values, with $\lambda \geq 0$.
For $\lambda = 0$, it is clear from equations (27) and (30) that both

$$bias_{OLS}(\boldsymbol{x}) = bias_{pseudo-R}(\boldsymbol{x}) \tag{31}$$

and

$$var_{OLS}(\boldsymbol{x}) = var_{pseudo-R}(\boldsymbol{x}). \tag{32}$$

This is not surprising since the two training ways differ only by the presence of λ; for $\lambda = 0$, both algorithms are equivalent.
At the opposite, for $\lambda \to +\infty$, one can observe

$$bias_{pseudo-R}(\boldsymbol{x}) \to h_b(\boldsymbol{x}) \tag{33}$$

and

$$var_{pseudo-R}(\boldsymbol{x}) \to 0. \tag{34}$$

By increasing the value of λ, one can thus easily see that, on one hand, the variance asymptotically decreases towards 0. This comes from the fact that λ corresponds to a penalizing factor, penalizing large 2-norm vectors $\boldsymbol{w}$ [1]. For $\lambda \to +\infty$, the penalizing term dominates and the optimal vector is thus $\mathbf{0}$, regardless of the inputs of the different learning samples. As a conclusion, λ impacts the variance by reducing the range of vectors $\boldsymbol{w}$ that can be optimal from a learning sample to another, thus reducing the variance.
On the other hand, the bias asymptotically goes towards the value of $h_b(\boldsymbol{x})$, which, according to the definition of the bias, corresponds to the average model $E_{LS}\{\hat{y}(\boldsymbol{x})\}$ tending

1. $\boldsymbol{w}^T \boldsymbol{w}$ equivalently writes $\|\boldsymbol{w}\|_2^2$

towards 0. This is certainly not a wanted behavior for a model, and it comes from the fact that, again, the penalizing term dominates for $\lambda \to +\infty$, leading to the only optimal vector $\mathbf{0}$, thus leading to a zero average model. As a conclusion, λ impacts the bias by increasing the latter, since too large values of λ prevent the models from correctly fitting the data.

For large values of λ, one can thus observe a high bias and a low variance, meaning that large values of λ lead to underfitting the data.

2 Empirical analyses

a. Analytically, for the considered problem and at a given point x_0,

- the residual error is given by

$$\begin{aligned} noise(x_0) &= E_{y|x_0}\{(y - h_b(x_0))^2\} \\ &= E_{y|x_0}\{(y - E_{y|x_0}\{y\})^2\} \\ &= E_{y|x_0}\{(f(x_0) + \epsilon - E_{y|x_0}\{f(x_0) + \epsilon\})^2\} \\ &= E_{y|x_0}\{(f(x_0) + \epsilon - E_{y|x_0}\{f(x_0)\} - E_{y|x_0}\{\epsilon\})^2\} \\ &= E_{y|x_0}\{(f(x_0) + \epsilon - E_{y|x_0}\{f(x_0)\} - \mu_\epsilon)^2\} \\ &= E_{y|x_0}\{(f(x_0) + \epsilon - E_{y|x_0}\{f(x_0)\})^2\} \\ &= E_{y|x_0}\{(f(x_0) + \epsilon - f(x_0))^2\} \\ &= E_{y|x_0}\{(\epsilon)^2\} \\ &= E_{y|x_0}\{(\epsilon - \mu_\epsilon)^2\} \\ &= \boxed{\sigma^2} \end{aligned} \tag{35}$$

  with σ^2 the variance of ϵ and $\mu_\epsilon = 0$ its mean as given in the statement, and $h_b(x) = E_{y|x}\{y\}$ the Bayes model.

- the squared bias is given by

$$\begin{aligned} bias^2(x_0) &= (h_b(x_0) - E_{LS}\{\hat{y}(x_0)\})^2 \\ &= (E_{y|x_0}\{y\} - E_{LS}\{\hat{y}(x_0)\})^2 \\ &= \boxed{(f(x_0) - E_{LS}\{\mathcal{A}(x_0)\})^2} \end{aligned} \tag{36}$$

  with $\mathcal{A}$ the supervised learning algorithm as given in the statement.

- the variance is given by

$$\begin{aligned} variance(x_0) &= E_{LS}\{(\hat{y}(x_0) - E_{LS}\{\hat{y}(x_0)\})^2\} \\ &= \boxed{E_{LS}\{(\mathcal{A}(x_0) - E_{LS}\{\mathcal{A}(x_0)\})^2\}}. \end{aligned} \tag{37}$$

Since the expected error is equal to the sum of the previous three quantities, it is given by

$$\boxed{expected\ error(x_0) = \sigma^2 + (f(x_0) - E_{LS}\{\mathcal{A}(x_0)\})^2 + E_{LS}\{(\mathcal{A}(x_0) - E_{LS}\{\mathcal{A}(x_0)\})^2\}.} \tag{38}$$

b. Assuming that one can generate samples for a given value x, an experimental protocol to estimate the different quantities of the previous section at a given point x_0 is presented in the following.

First, on the basis of f and the noise distribution, generate a great number M_1 of samples $(y(x_0), x_0)_i$.
Second, on the basis of f and the noise distribution again, generate a great number M_2 of learning samples LS_j of size N_1 for x in a particular range containing x_0.
Since a computer can not deal with continuous variables, the range of values for x must contain a fixed integer number of values N_2 and always contain the value x_0. M_1 and M_2 must be large enough to deal with the fact that one can not generate an infinite number of learning samples but that the results must converge, while it must stay reasonable with respect to the computers computation capabilities.

i. To estimate the value of the Bayes model $h_b(x_0)$, average the value of $y(x_0) = f(x_0) + \epsilon$ over all the generated samples $(y(x_0), x_0)_i$.
To estimate the residual error, now that $h_b(x_0)$ has been estimated, compute $(y(x_0) - h_b(x_0))^2$ for each of the generated samples $(y(x_0), x_0)_i$ and average that quantity over all those samples.

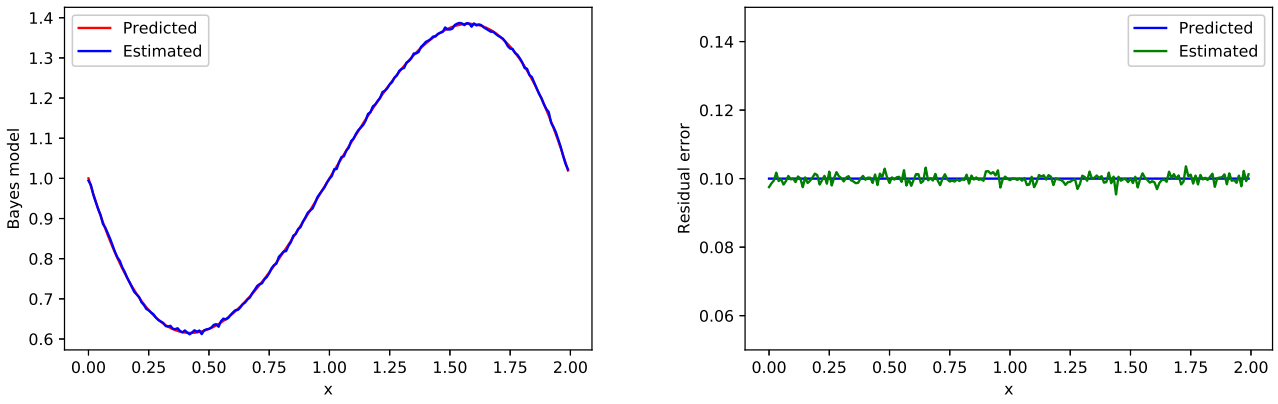
ii. To estimate the squared bias, first fit the algorithm $\mathcal{A}$ on each of the generated learning samples LS_j. Second, average the value predicted by each model $\mathcal{A}(x_0)$ over all LS_j. Finally, subtract this average value from the value $h_b(x_0)$ previously computed and square.

iii. To estimate the variance, now that the average predicted value over all LS_j has been computed, compute, for each model and thus for each LS_j, the difference between the predicted value $\mathcal{A}(x_0)$ and this average value. Finally, average those differences over all LS_j.

iv. To estimate the expected error, compute the sum of the estimated residual error, squared bias and variance.

N.B. : In the rest of the project, one uses $M_1 = 10000$ and $M_2 = 1000$. The learning samples are of size $N_1 = 30$. The asked quantities are moreover evaluated for $N_2 = 200$ evenly spaced values of $x \in [0, 2]$, which provides a resolution of the order of one hundredth.

c. The graphs of the estimated Bayes model and the estimated residual error are obtained [2] following the protocol described in section 2.b. and can be seen in figures 1a and 1b respectively ; these two estimated quantities are superposed with the ones analytically computed with the expressions from section 2.a..



(a) (b)

FIGURE 1 – Estimated and predicted Bayes model (a) and residual error (b)

2. The dedicated Python script can be found in the Jupyter notebook file `Q2.cde.ipynb`. Figures from sections 2.d and 2.e are also generated with this script.

It is clear from figure 1a that the estimated values of the Bayes model nearly perfectly match the analytical values. From figure 1b, one can also see that, even if the estimated residual error is not constant over the range of x values, the amplitude of the observed ripples is of the order of 10^{-3}, which is very small comparing to the magnitude of the analytical value $\sigma^2 = 0.1$.

Therefore, for the sake of simplicity and to make the estimations more stable, one will from now use the analytical expressions $f(x)$ and σ^2 rather than the empirical estimates computed in this section for the Bayes model and for the residual error respectively.

d. For learning samples of size $N = 30$ and for complexity values $m = 0, \ldots, 5$, the graphs of the estimated square bias, variance and expected error of the model $\hat{y}_m$ can be seen in figures 2, 3 and 4 respectively.

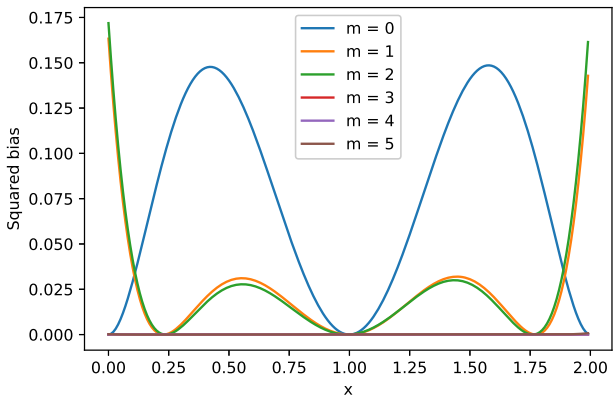


FIGURE 2 – Estimated squared bias

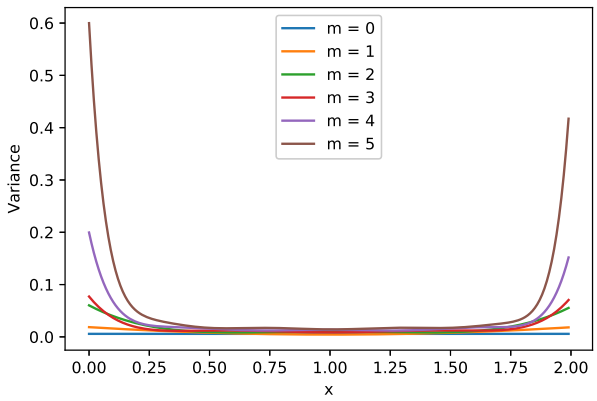


FIGURE 3 – Estimated variance

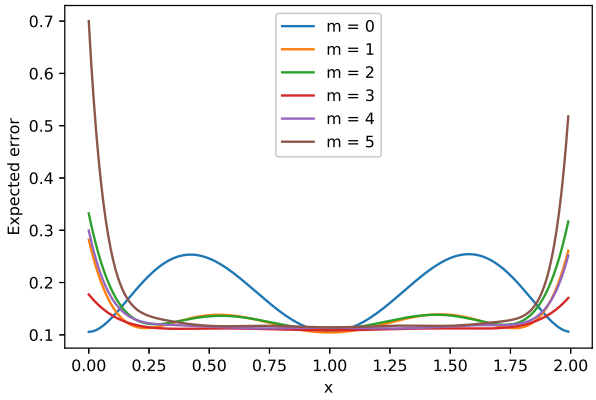


FIGURE 4 – Estimated expected error

These figures are obtained using the protocol described in section 2.b.. The different models are fitted using ordinary least-square as implemented by the function `LinearRegression` in Scikit-learn, whose parameter *fit_intercept* is set to *False*[3].

3. If $fit\_intercept = True$, the y-intercept, corresponding to a_0 in the considered case, will be determined by the line of best fit, rather than by ordinary least-square.

From figure 2, one can observe that increasing the complexity globally decreases the estimated squared bias. For values of m greater or equal to 3, the estimated squared bias can be considered to be 0 for all $x \in [0, 2]$. This means that, for each considered value of $x \in [0, 2]$, the average predicted value perfectly matches the Bayes model. Since the degree of the original polynomial $f(x)$ is equal to 3, this observation is not surprising as the models could not well approximate $f(x)$ with a degree lower than 3.
It is however interesting to focus on some particular values for x as, for lower degrees of complexity, the estimated squared bias is not uniform over the input space. Indeed, depending on the shape of the actual function to approximate, a given level of complexity will lead to models making better predictions for certain values of x than for other ones.
For $\boxed{x = 0}$, the squared bias can be considered as equal to 0 for $m = 0$ in addition to the previously observed values. This can be explained by looking at the shape of the Bayes model in figure 1a. Indeed, fitting a model for $m = 0$ by ordinary least-squares is likely to lead to a constant value for the predicted output corresponding to an horizontal line symmetrically splitting the Bayes model in two parts. This horizontal line will thus in average nearly perfectly match the Bayes model for $x = 0$, $x = 1$ and $x = 2$. For values of m different from 0 and different from the values leading to a nearly constant and zero bias, one can observe a high squared bias, meaning that the first and second degree models do not well match the Bayes model in average for $x = 0$.
For $\boxed{x = 0.5}$, one can now observe that the squared bias is quite high for $m = 0$ and lower but non zero for $m = 1, 2$. For the same reason as depict for $x = 0$, the horizontal line for $m = 0$ gives a really bad approximation of the output at $x = 0.5$. The first and second degree polynomials correspond to lower bias. Indeed, first order ones correspond to an oblique line sloping such that they take into account the two "lobes" of the Bayes model and their opposite concavities. Therefore, they are expected to provide better predictions for $x = 0.5$ than order 0 models. The second order ones provide an average squared bias only slightly lower than the first order ones, meaning that at $x = 0.5$, models of theses two degrees are quite close to each other.
For $\boxed{x = 1}$, all complexity levels correspond to a zero squared bias. This means that for each value of m, the models are in average very close to the actual value of the Bayes model for $x = 1$. Since in the range $x \in [0, 2]$ the Bayes model has an obvious anti-symmetry with respect to $x = 1$, it is not surprising that fitting a linear model with ordinary least-squares will in average lead to a perfect prediction for $x = 1$.
For $\boxed{x = 1.75}$, one can observe that the only value of m leading to a non zero squared bias is $m = 0$, which can be again explained with the horizontal line image. For $m = 1, 2$, which do not correspond to a constant and nearly zero bias for all x, this means that the first and second order polynomials are in average shaped such that the output for $x = 1.75$ is well predicted, which thus means that such shapes are optimal in the ordinary least-squares sense.

From figure 3, one can observe that increasing the complexity tends to have the opposite effect on the estimated variance compared to that observed for the bias. Indeed, although this might not be clear in the figure for central values of x, the estimated variance globally increases with complexity.
Again, it is interesting to focus on particular values of x.
For $\boxed{x = 0}$, the level of complexity has a clear impact on the estimated variance. Indeed, it is clear that increasing the latter leads to an increase in variance. This is not surprising since increasing the number of coefficients provides a wider range of approximation curves. For $x = 0$, predictions seem to vary a lot from a learning sample to another as the complexity increases. This is due to the fact that as an boundary value of the domain, $x = 0$ is a very sensitive value when interpolating; at this boundary, the greatest power of x becomes predominant, which

leads to this high sensitivity.
For $\boxed{x = 0.5}$ and $\boxed{x = 1}$, the level of complexity only has a slight impact on the estimate variance. For $\boxed{x = 1.75}$, the impact is more pronounced but remains small comparing to the effect observed for $x = 0$. For these three values, the variance remains in the same order of magnitude for the different values of m, while still increasing a bit with it. This means that these values are less impacted by the differences between the learning samples, regardless of complexity. On average, for a given value of m, the different models are close to the average one for these values of x. These values are less sensitive than $x = 0$.

From figure 4, one can not fetch any global tendency. Indeed, the estimated expected error combines the impacts of both previously analyzed quantities, which are different. One can however focus on the same particular values of x as before.
For $\boxed{x = 0}$, one can observe that the lowest complexity level corresponds to the smallest expected error, while the highest complexity level corresponds to the highest expected error. For intermediate values of m, the lowest expected error is obtained for $m = 3$.
For $\boxed{x = 0.5}$, a value of m equal to $m = 0$ is clearly the worst. $m = 1$ and $m = 2$ lead to a smaller expected error but are still above the lowest expected error provided by values of m greater or equal to 3.
For $\boxed{x = 1}$, the expected error is quite small for all values of m. This indeed corresponds to both low bias and low variance, meaning that regardless of the complexity, the considered linear model fitted by ordinary least-square is good at predicting the output for $x = 1$.
For $\boxed{x = 1.75}$, the different complexity levels are in the same range of error, except for $m = 0$, which again corresponds to a high expected error.
Taking into account the different curves for the different studied specific points, it seems that the complexity level ensuring a globally low level of expected error is $m = 3$, which seems coherent regarding the degree of the original polynomial $f(x)$.

e. The mean values of the previous quantities over the input space are estimated by the average values of the previous estimated quantities over the 200 values of x used for the different plots of sections 2.c. and 2.d.. These estimated mean values, plotted for increasing values of m, are represented in figures 5, 6 and 7.

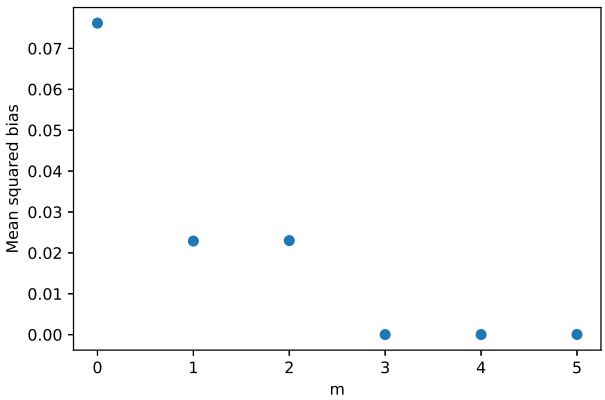


FIGURE 5 – Mean estimated squared bias

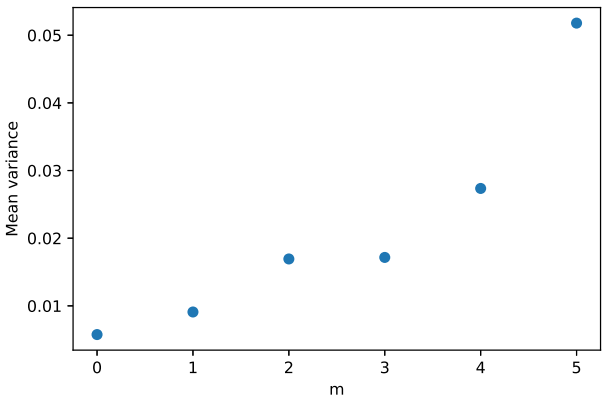


FIGURE 6 – Mean estimated variance

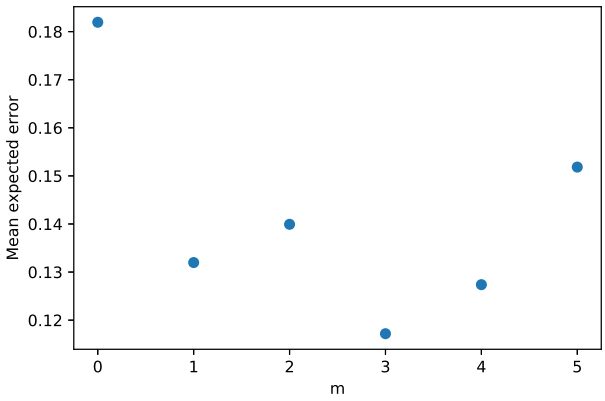


FIGURE 7 – Mean estimated expected error

On one hand, one can first observe from figure 5 that, as already said in the previous section, the mean value of the squared bias globally decreases with the complexity m of the model. For values of m larger or equal to 3, one can again see that the mean value of the squared bias is nearly equal to 0.These two observations are not surprising. Indeed, by increasing the complexity of the model, one can hope to better approximate the actual $x - y$ linking relationship than with a lower one, since $f(x)$ is cubic. Therefore, it seems coherent to observe that, in average over the input space, the error between the Bayes model and the average model decreases with the complexity. The fact that one observes no more clear improvement (in the bias sense) for values of m greater than 3 is related to the actual degree of $f(x)$, which is 3. Increasing m above this value can intuitively not generate models with a smaller average difference between the Bayes model and the average one.

Second, one can observe in figure 6 that, on the other hand and as said in the previous section, the mean variance is an increasing function of the complexity. Again, this result is not surprising, since increasing the complexity intuitively allows to tune more the coefficients of the model, causing it to differ in a wider range from one learning sample to another. In the opposite, a low value of m leads to a smaller degree of freedom ; in particular, for $m = 0$, the model consists in only one constant coefficient.

In figure 7, one can finally observe the estimated coupled contribution of both the squared bias and the variance, in average over the input space [4]. In the expected error sense, the optimal value for m is 3, which seems coherent regarding the actual degree of $f(x)$. In the opposite, the values $m = 0$ and $m = 1$ on one hand, and $m = 4$ and $m = 5$ on the other hand correspond to larger values of the expected error. Indeed, from figures 5 and 6, the first two values correspond to both a high bias and a low variance compared to the optimal complexity, meaning that models with such a complexity are likely to underfit the data. The other two values correspond to a low bias and a high variance compared to the optimal complexity, which means that models with such a complexity are likely to overfit the data. The complexity value $m = 3$ is thus the value which balance bias and variance in the considered problem. As previously said, an imbalance towards variance (*i.e.* a high variance) leads to overfitting while an imbalance towards bias (*i.e.* a high bias) leads to underfitting.
One can also note that a value of $m = 2$ does not correspond to a low bias or a low variance, leading to a quite large value of the mean expected error.

4. The estimated error also takes into account the effect of the residual error. Since the latter is considered here as constant, one does not consider it in the analysis.

A synthetic representation of this bias and variance trade-off is represented in figure 8 [5].

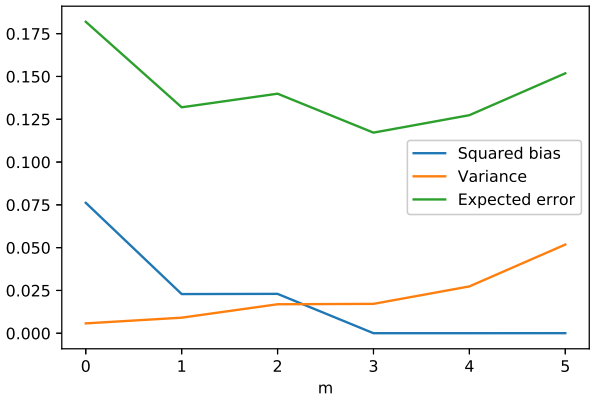


FIGURE 8 – Models comparison

It is clear from this figure that one model can not have both a low bias and a low variance at the same time, in average over the input space. The mean expected error allows a quick comparison between the different levels of complexity in order to identify the level balancing bias and variance and thus providing a trade-off between these two quantities.

f. Setting $m = 5$ and using ridge regression as implemented in the function `Ridge` [6] in Scikit-learn to fit the models, the same quantities as before can be estimated and averaged over the input space [7]. One can thus plot these mean quantities with respect to the regularisation level λ (for $\lambda \in [0, 2]$, with a number of points equal to 200), as it can be seen in figures 9, 10 and 11.

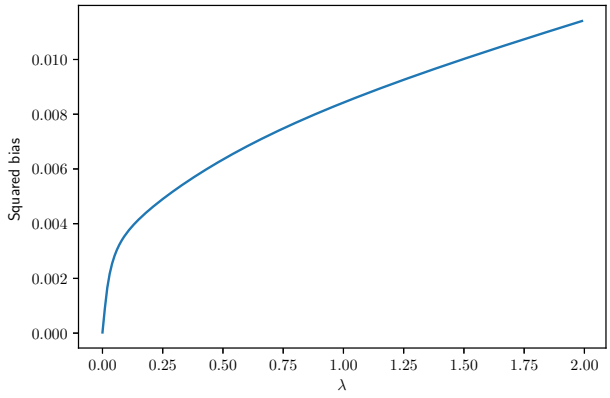


FIGURE 9 – Mean estimated squared bias

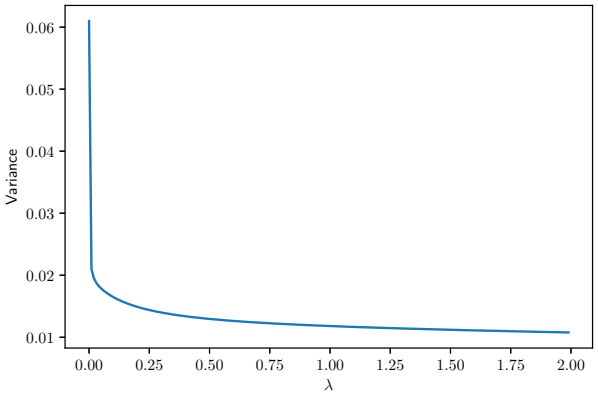


FIGURE 10 – Mean estimated variance

5. Although m is a discrete variable, the different points of the different curves are here linked for the sake of the representation.

6. Again, the parameter *fit_intercept* is set to *False* for the same reason depict in section 2.d..

7. The dedicated Python script can be found in the Jupyter notebook file `Q2.f.ipynb`.

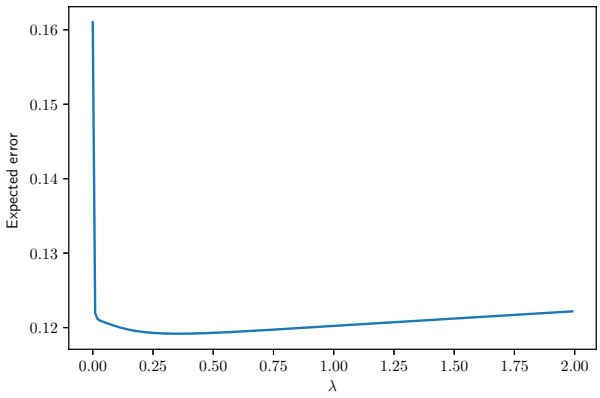


FIGURE 11 – Mean estimated expected error

From figures 9 and 10, one can observe what has been predicted in section 1.b.ii [8], *i.e.* an increase of the bias and a decrease of the variance with respect to λ. The opposite effect of complexity is thus observed.
From figure 11, one can observe that there exists an optimal value for λ in the expected error sense. This optimal value, located around $\lambda = 0.3$, is the value providing a trade-off between bias and variance, while other values of λ lead to an imbalance between those two quantities : low values of λ lead to low bias and large variance (hence to overfitting) while large values of λ lead to high bias and small variance (hence to underfitting).

For a given complexity value m, it is thus possible to identify a value for λ minimizing the mean expected error ; it would be interesting to analyze both the impact of m and λ to seek for the best combination.

8. Although assumptions and simplification were made in section 1.b.ii.