FTML practical session 2

1^{er} mars 2025

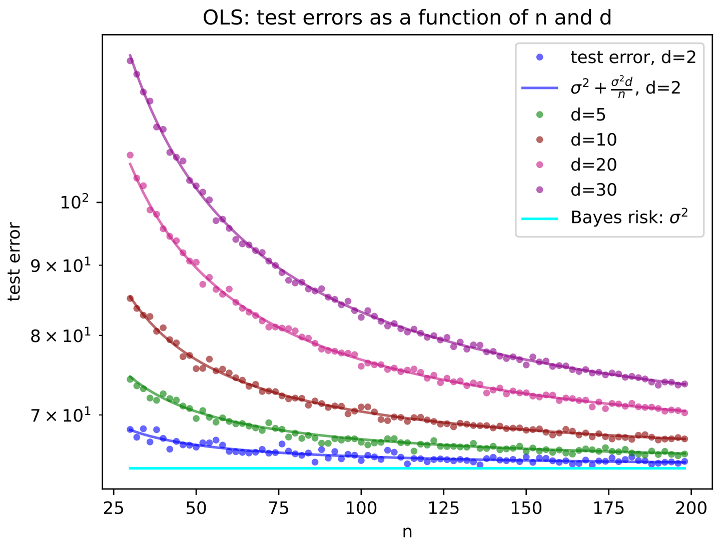

TABLE DES MATIÈRES

INTRODUCTION

The goal of this session is to keep exploring the concepts linked to risks : Bayes risk, empirical risk, overfitting. We keep exploring them in a linear regression context, but dive a bit deeper technically and start studying the influence of the dimensions (number of samples, number of features) on the statistical properties of the results. Some aspects of this session are a bit technical, it is normal if you need to take some time to understand all the objects used in the session.

There are some template files in **exercice_1_ols/** that you can use in the repo, but you can also start from scratch with your own files if you prefer.

— **main_ols.py**

— **utils_plots.py**

— **utils_algo.py**

— **constants.py**

1 LINEAR REGRESSION

A **linear model**, such as the Ordinary least squares (OLS, presented below), can be interpreted as predicting an output value (dependent variable) from combining the contributions from the d **features** of the input data (independent variables), in a linear way. This can be useful for classification as well as for regression.

1.1 A simple example of linear function

For instance, if I want to predict the amount of money that I will spend when buying some clothes, I can use a linear mapping (aka function). We assume that there are d possible clothes in a shop. If $\theta \in \mathbb{R}^d$ is a vector that contains the price of each type of clothe, and $x \in \mathbb{N}^d$ the number of each type of clothe that I buy, then I have to spend $x^T\theta$. If there exists $d = 4$ types of clothes with a price θ_i :

— socks : $\theta_1 = 2$

— T-shirts : $\theta_2 = 25$

— pants : $\theta_3 = 50$

— hats = $\theta_4 = 20$

$$\theta = \begin{pmatrix} 2 \\ 25 \\ 50 \\ 20 \end{pmatrix}$$

If I want to buy 10 socks, 2 T-shirts, 1 pants and 1 hat, then $x^T = (10, 2, 1, 1)$ and I spend

$$\begin{aligned} x^T\theta &= 10 \times 2 + 2 \times 25 + 1 \times 50 + 1 \times 20 \\ &= 140 \end{aligned} \tag{1}$$

1.2 Linear models

When doing a **linear regression** in a supervised learning context, we are using a linear function of the form $x \mapsto x^T\theta$ in order to approximate the outputs, where $x \in \mathbb{R}^d$ is an element of our input space, and $\theta \in \mathbb{R}^d$. The objective is then to find the best θ for this purpose.

Of course, not all phenomena can be approximated well in a linear way. For many tasks, neural networks and many other models outperform linear regression. However, linear regression is a foundation for more advanced modelisation that we will study in future classes (feature maps, kernel methods, neural networks, etc).

It is also important to note that linear regression can often be used as a first benchmark when comparing different kinds of models, in a supervised learning context. A model that is more difficult to train or takes a longer time to train than a linear regression should have a better performance than linear regression, in order to be worth using it!

As we will study the influence of the dimensions of the problem n (number of samples) and d (number of features of each sample), we will work with abstract datasets, instead of a given "physical dataset". However, you can always refer to the previous example in order to interpret linear models that we will use.

1.3 Formalization

Let us abstract the notations a little bit. In the OLS setting,

— $\mathcal{X} = \mathbb{R}^d$ (input space)

— $\mathcal{Y} = \mathbb{R}$ (output space)

An input sample is stored in a column vector $x_a \in \mathbb{R}^d$, with $a \in \mathbb{N}$ like

$$x_a = \begin{pmatrix} x_{a1} \\ \dots \\ x_{ai} \\ \dots \\ x_{ad} \end{pmatrix} \in \mathbb{R}^d \tag{2}$$

The estimator is a **linear mapping** parametrized by $\theta \in \mathbb{R}^d$. The prediction associated with $x \in \mathbb{R}^d$ is $f(x) = \theta^T x = x^T \theta$. If we use the squared-loss, the discrepancy between two real numbers y and y' writes $l(y, y') = (y - y')^2$. Finally, the input dataset is stored in the **design matrix** $X \in \mathbb{R}^{n \times d}$.

$$X = \begin{pmatrix} x_1^T \\ \dots \\ x_i^T \\ \dots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11}, \dots, x_{1j}, \dots x_{1d} \\ \dots \\ x_{i1}, \dots, x_{ij}, \dots x_{id} \\ \dots \\ x_{n1}, \dots, x_{nj}, \dots x_{nd} \end{pmatrix} \tag{3}$$

and the output labels are stored in a vector

$$y = \begin{pmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_n \end{pmatrix} \in \mathbb{R}^n \tag{4}$$

1.4 Empirical risk and generalization error

The **empirical risk** of an estimator θ (when we talk about an estimator, it is here equivalent to refer to θ directly or to the mapping defined by θ) writes :

$$\begin{aligned} R_n(\theta) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2 \end{aligned} \tag{5}$$

How can we write the empirical risk using only matrices and vectors, X, θ and y defined in 1.3 and an euclidean norm ?

By contrast, the **generalization error** (risque réel) of an estimator does not depend on the dataset. It is a fixed number equal to :

$$R(\theta) = E[(y - \theta^T x)^2] \tag{6}$$

where the expected value is taken over the law of $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$, the random variables representing the input and output respectively. This law is unknown, in general.

2 ORDINARY LEAST SQUARES : EMPIRICAL RISK AND OVERFITTING

In this exercice, we want to study the influence of the dimensions n and d on the amount of **overfitting** of a linear regression. In order to do so, we will need to make statistical assumptions about the data that we use.

2.1 OLS estimator

The **OLS estimator**, noted $\hat{\theta}$, is the value of θ that minimizes $R_n(\theta)$. It is thus the solution to the problem of **empirical risk minimization**, a standard approach in supervised learning. We admit the following proposition (that we will prove later in the course) :

Proposition. *Closed form solution*
*We keep the notations from 1.3. If X is injective, there exists a unique minimiser of $R_n(\theta)$, called the **OLS estimator**, given by*

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{7}$$

2.2 Excess risk the OLS estimator

Given an estimator θ, we are interested in its **excess risk**. It is the difference between its generalization error and the bayes risk. We will evaluate the excess risk of the OLS estimator as a function of n and d, for a given statistical setting, in order to observe the results of figure **??**.

2.3 Statistical setting

In order to compute these quantities, it is necessary to make statistical assumptions. We will use the **linear model**, with **fixed design**, a classical framework to analyze OLS. In this setting, we assume that X is given and fixed, and that there exists a vector $\theta^* \in \mathbb{R}^d$, such that $\forall i \in \{1, \dots, n\}$,

$$y_i = x_i^T \theta^* + \epsilon_i \tag{8}$$

where for all $i \in \{1, \dots, n\}$, ϵ_i are independent, with expectation $E[\epsilon_i] = 0$ and variance $E[\epsilon_i^2] = \sigma^2$. The ϵ_i represent a variability in the output, that is due to **noise**, or to the presence of unobserved variables. Put together in a vector $\epsilon \in \mathbb{R}^n$, this allows to write

$$y = X\theta^* + \epsilon \tag{9}$$

Considering yesterday's practical session observations, what do we expect to be the Bayes estimator ?

We admit (but you can also prove it) that the corresponding Bayes risk is σ^2.

2.4 Verification of the Bayes estimator

For a given n and d (with $d \leqslant n$), run a simulation that reproduces the fixed design, linear model statistical setting and verify that the the estimator that you have chosen achieves the Bayes risk. Verify that different estimators have worse (larger) risks. You need to generate first a X and a θ^*.

You will again need to approximate the real risk by the test error, a number of times that is sufficient to observe the convergence of the law of large numbers.

2.5 Empirical risk minimization and impact of the dimensions on the OLS overfitting

We have seen that in this statistical setting, the Bayes risk is σ^2. In practical applications, we of course do not have access to θ^* and we must find a relevant θ, based on the data only. We must resort to **empirical risk minimization**.

We want to compare the risk of the OLS estimator (obtained by empirical risk minimization) to the Bayes risk, to know how good the OLS estimator compares to the Bayes estimator. Again, remember that in practical applications, we will not have access to the Bayes estimator or to the Bayes risk.

Run a simulation that approximates the expected value of the generalization error of the OLS estimator. Remember that as the train data should be seen as random variables, the OLS estimator $\hat{\theta}$ himself should also be seen as a random variable, and so should be its generalization error. This is why we are approximating an **expected value**!
To do so, we can generate several train and test data, and average the results of the tests errors in order to approximate this exepcted value. Depending on the number of repetitions of the experiment, on which we perform the average, we should observe results simlar to that of figures 1, 2, and 3.

Remarks

— In order to generate the X matrices in various dimensions, you can use uniformly distributed entries. This should ensure that X is injective.

— We will prove during the lectures the different results mentioned in this exercice.

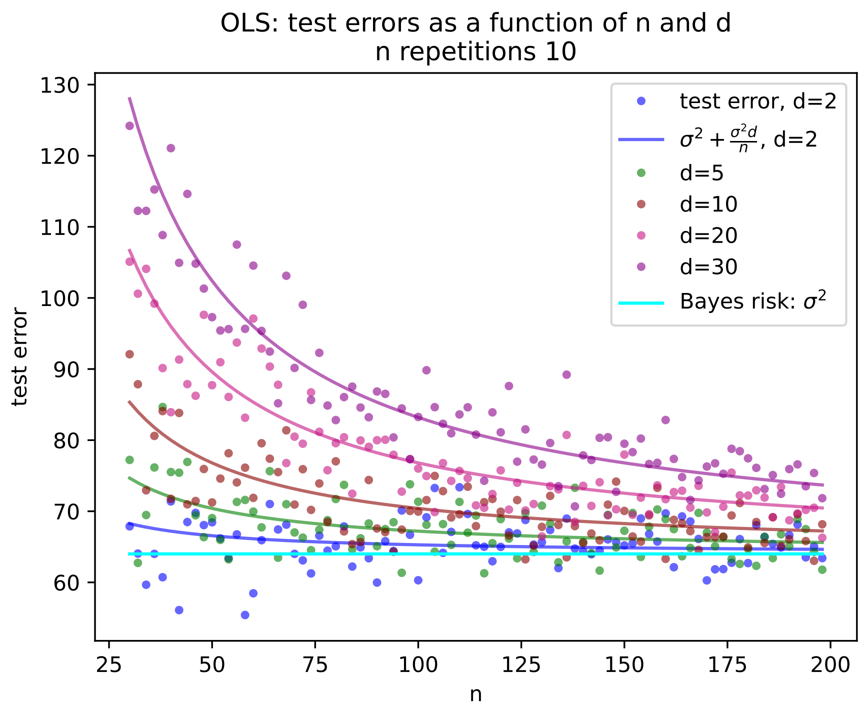

FIGURE 1 – Dependence of the test error (which approximates the generalization error (risque réel)) to the dimensions.

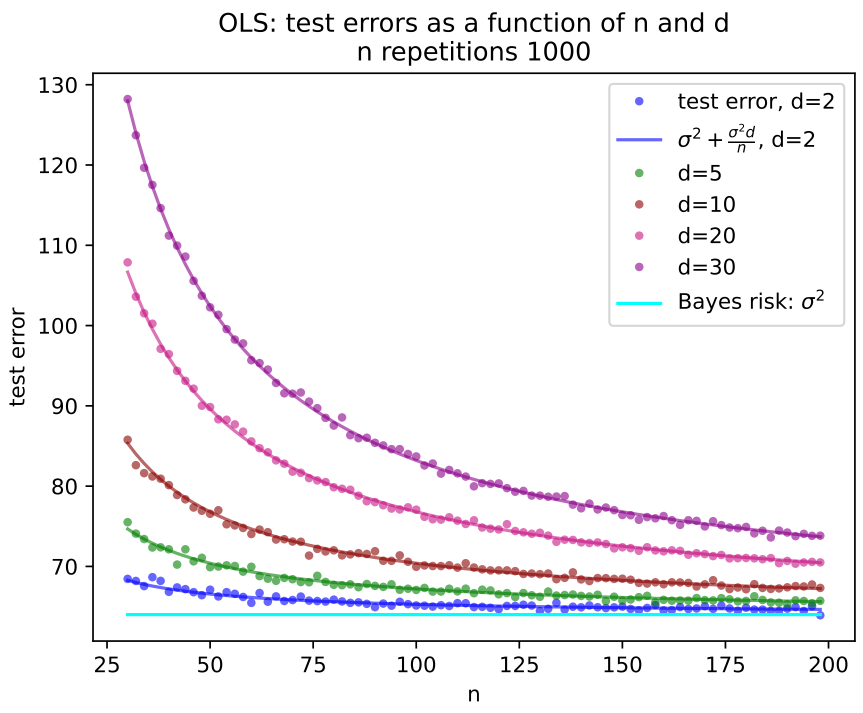

FIGURE 2 – Dependence of the test error (which approximates the generalization error (risque réel)) to the dimensions.

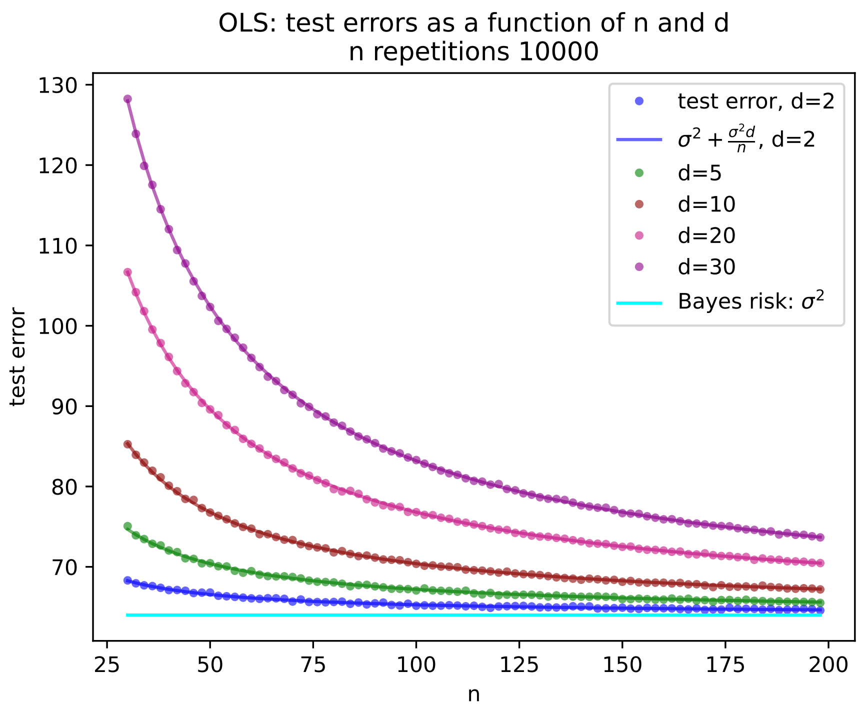

FIGURE 3 – Dependence of the test error (which approximates the generalization error (risque réel)) to the dimensions.

We can make the following, intuitive observations :

— overfitting is smaller when n is larger

— overfitting is larger when d is larger

It is possible to show that the amount of overfitting is given by $\sigma^2 \frac{d}{n}$. Let me know if you are interested in proving this result.

2.6 When $d > n$

If $d > n$, the $\sigma^2 d/n$ law is not respected !

Why can we still output an OLS estimator when $d > n$?

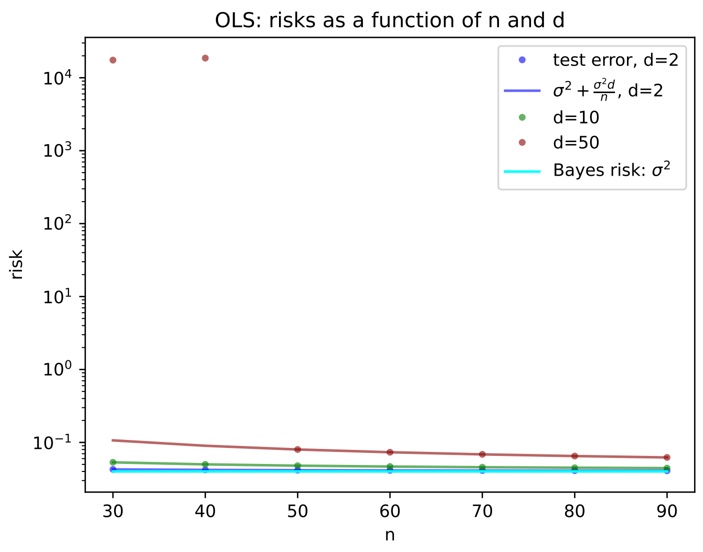

FIGURE 4 – Test errors with cases where $d > n$.

Optional : monitor the stability of the OLS estimator as a function of n and d, for instance by computing the sample standard deviation of the random variable $\|\hat{\theta} - \theta^*\|/\|\theta^*\|$ as a function of n and d. When d is closer to n, $\hat{\theta}$ can become very unstable (a small variation of the noise might lead to a large variation of $\hat{\theta}$). This behavior depends on the conditioning of $X^T X$, as we will see later.)

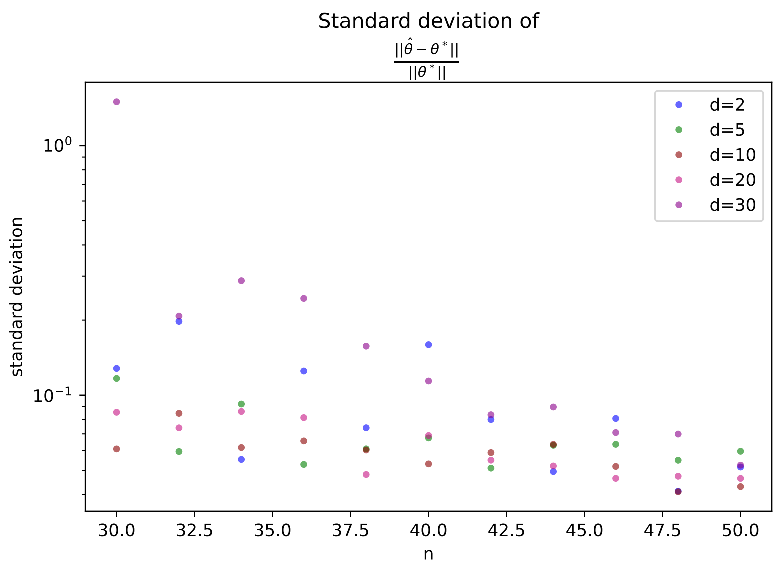

FIGURE 5 – Instability of the OLS estimator when d gets close to n.