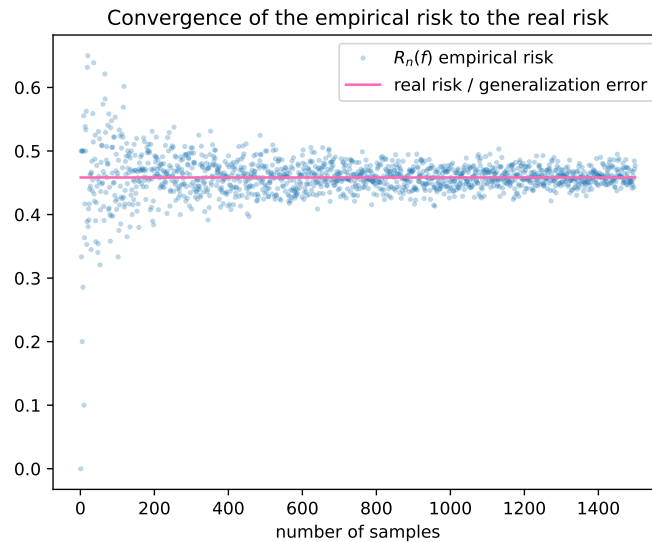


# FTML practical session 1: 2024/03/15



## TABLE DES MATIÈRES

1	Miscellaneous python	2
1.1	Environments	2
1.2	Good python habits	2
1.3	Demos	2
1.4	Operations in Python	2
2	Experimenting with the law of large numbers	2
2.1	More Python profiling	3
3	Reminders on the risks definitions	3
3.1	Risks	3
3.2	Estimating the generalization error	3
4	Overfitting and variance of the test error	4
4.1	Standard deviation of the test error	4
4.2	Overfitting	4
5	Bayes risks	5
5.0.1	Bayes estimator	5
5.1	Two examples	5
5.1.1	Problem 1 : penalty shootout	5
5.1.2	Problem 2 : prediction of the number of spotify streams	5

## INTRODUCTION

The goal of this practical session and of the next one is to experiment with some concepts that are specific and central to machine learning : namely, the law of large numbers, overfitting and risks (empirical risk and generalization error). During the lectures that will follow these two practical sessions, we will study and formalize the definitions of the different types of risks in more depth. You can do the different 3 parts in whatever order but the natural order is rather 2, 4, 5 (1 and 3 are not really exercises). You do not have to finish everything during the session.

## 1 MISCELLANEOUS PYTHON

### 1.1 Environments

To install libraries, you can use virtual environments.

<https://docs.python.org/3/library/venv.html>

The list of libraries used in the practical sessions is in `practical_sessions/requirements.txt`. You can use it to install all libraries directly with pip, e.g. with `pip install -r requirements.txt`.

### 1.2 Good python habits

You can explore these tools later.

- Formatting : <https://github.com/psf/black>
- Imports sorting : <https://github.com/PyCQA/isort>
- Code style : <https://realpython.com/python-pep8/>

### 1.3 Demos

In `practical_sessions/tp1/demos/`, you can find a couple of simples demo files to use matplotlib, numpy (if needed).

### 1.4 Operations in Python

Time complexity of elementary operations in python :

<https://wiki.python.org/moin/TimeComplexity>

## 2 EXPERIMENTING WITH THE LAW OF LARGE NUMBERS

Let us consider the same variable as in exercice P3 :  $Z_2 = Z_1$  and is  $Z_1^2$  is a uniform law in  $[1, 2]$ . We have seen that  $E[Z_2] = 7/3$ . Hence, according to the law of large numbers, the empirical average of  $n$  draws of this variable converges in probability to this expected value.

In `exercice_1/law_of_large_numbers.py`, the function `empirical_average_loop` computes the empirical average with a for loop.

- Edit the function `empirical_average_array` in order to use numpy and array operations to perform the same computation in an optimized way, only using array operations and without a loop.
- Compare the speed of the methods by monitoring the `profile.prof` profiling file, for instance using `snakeviz profile.prof`.

**References :**

- [https://en.wikipedia.org/wiki/Array\\_programming](https://en.wikipedia.org/wiki/Array_programming)
- <https://jiffyclub.github.io/snakeviz/>

Note that it is also possible to save the profiling to a text file.

**2.1 More Python profiling**

It is also possible to profile individual lines :

<https://pypi.org/project/line-profiler/>

Or to profile memory usage :

<https://pypi.org/project/memory-profiler/>

**3 REMINDERS ON THE RISKS DEFINITIONS**

In this section we just recall some definitions on the risks, used in the following exercises.

We consider a supervised learning problem,

- an input space  $\mathcal{X}$
- an output space  $\mathcal{Y}$
- a loss function  $l$
- and a dataset  $D_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$  of  $n$  samples.

**3.1 Risks**

An **estimator**  $f$  is a mapping from the input space to the output space.

**Definition 1.** Risks

Let  $l$  be a loss. The **risk** (or **statistical risk**, **generalization error**, **test error**, **risque réel**) of estimator  $f$  writes

$$R(f) = \mathbb{E}_{(X,Y) \sim \rho} [l(Y, f(X))] \quad (1)$$

Here,  $X$  is the random variable that represents the inputs, and  $Y$  the variable that represents the output.  $\rho$  is the joint law.

The **empirical risk (ER)** (risque empirique) of an estimator  $f$  writes

$$R_n(f) = \frac{1}{n} \sum_{i=1}^n l(y_i, f(x_i)) \quad (2)$$

We emphasize that the risks depends on the loss  $l$ .

In supervised learning, we only have access to the empirical risk  $R_n$  but we actually want to find an estimator  $f$  which as a small generalization error! The problem is that in real situations, we do not have access to  $\rho$ , which would allow its computation.

**3.2 Estimating the generalization error**

**However**, thanks to the law of large numbers, if we use a **fixed estimator**, that is independent on the dataset,  $f$ ,  $R_n(f) \rightarrow R(f)$  when  $n \rightarrow \infty$ . Hence, if we have a large enough number of samples  $n$ ,  $R_n(f)$  is a good estimation of  $R(f)$ . The problem will then be : how large is sufficient? the answer will depend on the context.

## 4 OVERFITTING AND VARIANCE OF THE TEST ERROR

In this exercise we experimentally study the amount of overfitting and the empirical variance of the test error as a function of the number of samples used in the datasets. To do so, we use a linear regression problem.

### 4.1 Standard deviation of the test error

In statistical learning, we consider that datasets are sampled from random variables. Hence, any quantity that is computed from a dataset is also a random variable. For instance, the **test error**, which is the empirical risk on a **test set** (a dataset that was not used at all for training) is a real random variable, that has a standard deviation. Intuitively, and according to the law of large numbers, the test error converges to the generalization error (risk) when the number of samples becomes large enough.

Using [scikit-learn](#), optimize a linear estimator on the train set, and monitor the standard deviation of the test error as a function of the number of samples used in the test set.

You can find a template file to edit `exercice_2.py` in `exercice_2/`. You should observe a plot like [Figure 1](#).

You may also experiment with the number of samples used in the train set. This parameter will also influence the test error (which is consistent with [4.2](#)).

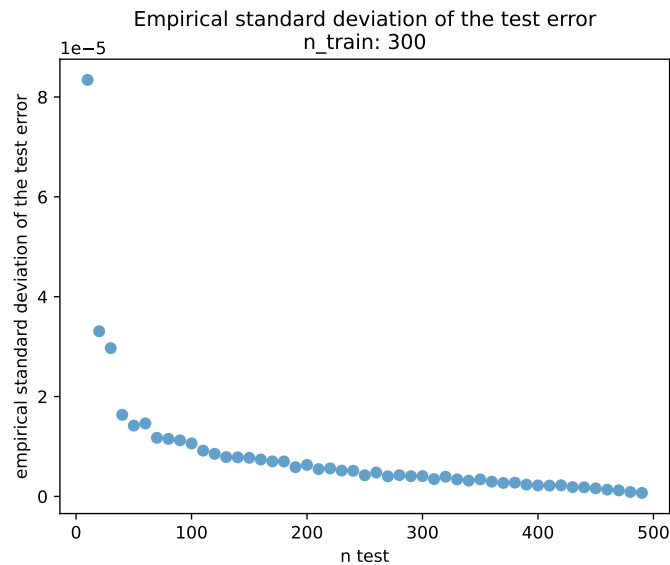


FIGURE 1 – This behavior is typical of the law of large numbers

### 4.2 Overfitting

Monitor the amount of overfitting as a function of the number of samples used in the train set.

You should observe a plot like [Figure 2](#).

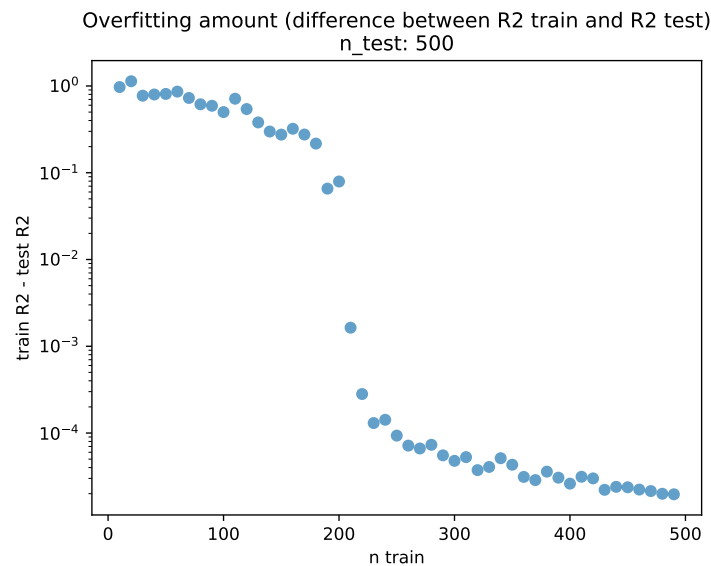


FIGURE 2 – This behavior is typical of linear regression on this kind of datasets.

## 5 BAYES RISKS

The goal of this exercise is to introduce the notion of Bayes risk.

### 5.0.1 Bayes estimator

Under some simple hypotheses, for a given supervised learning problem, there exists an optimal estimator  $f^*$  called the **Bayes estimator**, which minimizes the generalization error, given a distribution  $\rho$ . Its generalization error  $R(f^*)$  is called the **Bayes risk**.

### 5.1 Two examples

For the two following problems :

- estimate the generalization error of various estimators of your choice by simulating the random variables.
- find the Bayes estimator.

#### 5.1.1 Problem 1 : penalty shootout

We represent a football penalty shootout.  $X \in \{0, 1\}$  is the variable representing that team 1 shoots first.  $Y \in \{0, 1\}$  is the variable representing the fact that team 1 wins. We assume that :

- $X$  is uniformly distributed.
- If  $X = 1$ ,  $Y$  follows a Bernoulli law of parameter 0.6. If  $X = 0$ ,  $Y$  follows a law of parameter 0.4.
- $l$  is the 0 – 1 loss (1 if there is a mistake, 0 otherwise)

For this setting, the Bayes risk is 0.4.

#### 5.1.2 Problem 2 : prediction of the number of spotify streams

A music label is interested in predicting the number of streams of an artist, as a function of the investment. We will consider that the investment is represented by the number of persons who work with the artist during the production, which

is a proxy to the investment. This variable is noted  $X$ . More precisely, we predict the number of streams of the song on Spotify, noted  $Y$ , during the first week after release, as a function of  $X$ . We assume that :

- $X - 1 \in \mathbb{N}$  follows a binomial law of parameters  $n_X = 20$  and  $p_X = 0.2$ . Hence,  $X > 0$ .
- Given a value  $x$  of  $X$ ,  $Y$  follows a binomial law of parameters  $n_Y(x) = 3^x$  and  $p_Y(x) = 0.5$ .
- $l$  is the squared loss.

For this setting, the Bayes risk is around 627.