

# TD1 - Machine Learning

Mathieu Emily - mathieu.emily@agrocampus-ouest.fr

---

**Objectives :** Bayes classifier, kNN, train and test dataset

---

In this example, we consider a toy situation where the  $f$  function is known. Of course, such situation is not relevant in practice since our goal is to estimate  $f$ . However, to introduce the first concepts of machine learning it is more convenient to use known situation where we know the truth!

For that purpose, we start by simulating our dataset.

1. Plot the curve defined by :

$$\left(x, \frac{3 + \cos(4\pi * x)}{8} + \frac{x^2}{2}\right)$$

with  $x \in [0, 1]$ .


Let  $f$  be defined as follows :

$$\forall (x_1, x_2) \in [0, 1] \times [0, 1] : f(x_1, x_2) = x_2 - ((3 + \cos(4\pi * x_1))/8 + (x_1^2)/2)$$

## 1 Simulation of a dataset

Let  $Y$  the output :

$$\begin{aligned}\mathbb{P}[Y = 1 | X_1 = x_1, X_2 = x_2] &= \max(0, \min(1, 0.5 - f(x_1, x_2))) \\ &= 1 - \mathbb{P}[Y = 0 | X_1 = x_1, X_2 = x_2]\end{aligned}\tag{1}$$

1. Generate a random sample of  $n = 100$  observations where the  $i$ th observation is a triplet  $(y_i, x_{i,1}, x_{i,2})$  with  $(x_{i,1}, x_{i,2}) \in [0, 1]^2$  and  $y_i$  is a realization of  $Y$  given  $(X_1 = x_{i,1}, X_2 = x_{i,2})$ . Store the simulated dataset in an  object called `data.train`, where `data.train` is a `data.frame` with 3 columns named as `X1`, `X2` and `class`.
2. Plot the observations in the  $(X_1, X_2)$  coordinate system and color the points according to their class. Add the curve obtained in the first question in the plot.

## 2 The Bayes classifier

**Définition 1** *The Bayes classifier assigns each observation to the most likely class, given its predictor values*

1. Suppose that you don't know the `class` column in the `data.train` dataset but you know Equation 1, propose the Bayes classifier's and apply it to the `data.train` dataset. The corresponding predicted class can be stored in variable `pred.train.Bayes`.
2. Can you evaluate the performance of your classifier ?

### 3 the kNN classifier

The kNN is a non-parametric method that aim at estimating the conditional distribution of  $Y$  given  $X = (x_1, \dots, x_p)$ .

**Définition 2** Let  $x_0 = (x_1^0, \dots, x_p^0)$  be a point in the observed space and  $K$  an integer. The kNN first identifies the  $K$  observed points that are closest to  $x_0$ , represented by  $\mathcal{N}_0$ . It then estimates the conditional probability for class  $j$  as the fraction of points in  $\mathcal{N}_0$  whose response values equal  $j$  :

$$\mathbb{P}[Y = j|X = x_0] = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{I}(y_i = j)$$

Finally, kNN applies the Bayes rule and classifies the test observation  $x_0$  to the class with the largest probability.

1. Use the function `knn` (package `class`) to learn the kNN classifier with  $k = 20$ ,  $k = 10$ ,  $k = 5$ ,  $k = 3$ ,  $k = 2$  and  $k = 1$ .
2. Which classifier is the best ?
3. Is the best kNN classifier better than the Bayes classifier ?

### 4 On the use of a test dataset

1. Using the same procedure as for the `data.train` dataset, simulate the `data.test` dataset composed of 5000 observations.
2. Use the Bayes classifier and the various kNN classifiers trained in the previous question to predict the observations of the `data.test` dataset.
3. Which classifier is the best ?
4. Do we get similar conclusions as for the `data.train` dataset.

### 5 On the irreducible error

Let  $g$  be defined as follows :

$$\forall (x_1, x_2, d) \in [0, 1] \times [0, 1] \times \mathbb{R}^+ : g(x_1, x_2, d) = \text{sign}(f(x_1, x_2)) |f(x_1, x_2)|^d$$

1. Is the separability between the two classes is increasing or decreasing with  $d$ ?
2. How does the error rate of the Bayes classifier vary with  $d$ ?