Fondamentaux de l'Apprentissage Automatique

Lecturer: Yann Chevaleyre
Scribe: KALAA Yassine

Lecture n°1 #
28/09/2023

1 Introduction

In this lecture, we introduce introduce the concepts of empirical risk and true risk. These two notions are key to machine learning, especially when it comes for supervised learning, which means that the algorithm aims at predicting outputs from a labeled data set. We look at their uses and limitations, and give few examples.

2 Supervised Learning : a Formal Setting

Supervised learning is based on learning from a data set to provide predictions, our output values. Let us start by formalizing it.

For every learner, we have :

Spaces

There are three spaces :

- The input space : $\mathcal{X}$
- The label space : $\mathcal{Y}$
- The prediction space : $\hat{\mathcal{Y}}$

Examples

For a linear regression, we have :

- $\mathcal{X} = \mathbb{R}^d$
- $\mathcal{Y} = \mathbb{R}$
- $\hat{\mathcal{Y}} = \mathbb{R}$

For a bi-class classification problem, we have :

- $\mathcal{X} = \mathbb{R}^d$
- $\mathcal{Y} = \{0, 1\}$
- $\hat{\mathcal{Y}} = \{0, 1\}$

For a bi-class learning problem where we predict probabilities for each class, we have :

- $\mathcal{X} = \mathbb{R}^d$
- $\mathcal{Y} = \{0, 1\}$
- $\hat{\mathcal{Y}} = [0, 1]$

A data set as input :

$$S = \{(x_1, y_1) \dots (x_N, y_N)\}$$

A Loss Function :

The aim of the loss function is to evaluate how the prediction $\hat{y}$ fits the label y.

$$\begin{aligned}\ell : \hat{\mathcal{Y}} \times \mathcal{Y} &\to \mathbb{R}\\ (\hat{y}, y) &\mapsto \ell(\hat{y}, y)\end{aligned}$$

From the spaces that the learner works on, the loss function it uses and the data set given as input, the learner outputs a Prediction Function, also called a classifier.

This prediction function gets as input $x \in \mathcal{X}$ and outputs $\hat{y} \in \hat{\mathcal{Y}}$:

$$\begin{aligned}f : \mathcal{X} &\to \hat{\mathcal{Y}}\\ x &\mapsto f(x)\end{aligned}$$

Important Loss Functions

Least Square Regression method :

Spaces : $\hat{\mathcal{Y}} = \mathcal{Y} = \mathbb{R}$
Square loss :

$$\ell^{sq}(\hat{y}, y) = (\hat{y} - y)^2$$

Multiclass Classification method :

Spaces : $\hat{y} = y = \{1, \ldots, k\}$
0-1 loss :

$$\ell^{0/1}(\hat{y}, y) = \mathbf{1}_{[\hat{y} \neq y]} := \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases} \text{, also called the indicator function.}$$

Now that we have established what we are working with, and given some examples, we can ask ourselves how we can we get the best prediction model.

Indeed, we encounter an issue here : the loss function ℓ evaluates a single prediction and not all the predictions on the hole data set.

That is why we define the **Empirical Risk**.

3 Empirical Risk, True Risk and ERM

3.1 Empirical Risk

Definition 1. *Let $S = ((x_1, y_1), \ldots, (x_N, y_N))$. The empirical risk of $f : X \to \hat{\dagger}$ with respect to S is $\hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i)$.*

Definition 2. *A learning algorithm is an ERM (Empirical Risk Minimizing) algorithm if it outputs a function $\hat{f}$ minimizing the empirical risk :*

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}_N(f) \text{ where } \hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(f\left(x_i\right), y_i\right)$$

Remark

— Note that the empirical risk always depends on the given data set S.

— A few algorithms are strictly ERM, particularly because of its limitations that we will develop later on. For example, the ordinary linear regression method is strictly ERM (e.g. ordinary linear regression).

— On an another hand k-Nearest neighbors is not ERM because the class of function $\mathcal{F}$ does not exist.

Examples

Let us check the minimization problem of an ERM algorithm in some cases :

For the **linear regression** with squared loss :

We are looking for a linear function, thus, we have :

$$\mathcal{F} = \left\{ x \mapsto a^\top x + b, (a, b) \in \mathbb{R}^d \times \mathbb{R} \right\}$$

and

$$\hat{f}_{\hat{a},\hat{b}} = \underset{\mathcal{F}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_i \left(ax_i + b - y_i\right)^2 \right)$$

For the **linear classification** with 0-1 loss :

Here again, we are looking for a linear function, so :

$$\mathcal{F} = \left\{ x \mapsto a^\top x + b, (a, b) \in \mathbb{R}^d \times \mathbb{R} \right\}$$

and

$$\hat{f}_{\hat{a},\hat{b}} = \underset{\mathcal{F}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_i \mathbf{1}_{[g(x_i) \neq y_i]} \right) \text{ with } g(x_i) := \begin{cases} 1 & \text{if } a^\top x_i + b \geq y \\ 0 & \text{otherwise} \end{cases}$$

3.2 Overfitting

Definition 3. *We say that a model overfits when training performance is high but performance on other data from the same source is low.*

With ERM algorithms, overfitting often occurs if precautions are not taken.

Let us give an example to illustrate overfitting and thus, see one the limitations of ERM algorithms. For that, we take :

— A polynomial model $f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$

- $\mathcal{X} = \mathcal{Y} = \hat{\dagger} = \mathbb{R}$
- The square loss $\ell^{sq}(\hat{y}, y) = (\hat{y} - y)^2$
- Learning algorithms find the best parameters $w_0, w_1, \ldots, w_M$.
- M is a hyper-parameter that is set by an expert

The green curve is the one we want to predict : $(Y = \sin(2\pi x) + \epsilon)$

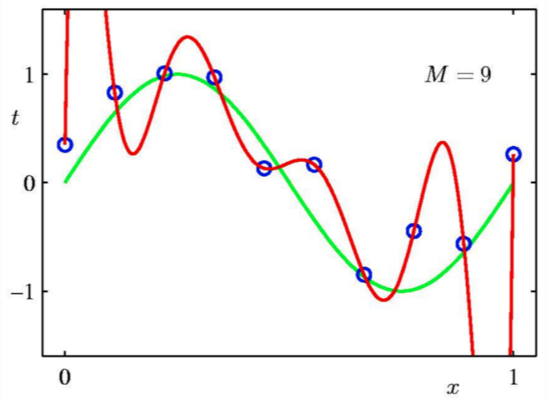


Figure 1 – Overfitting

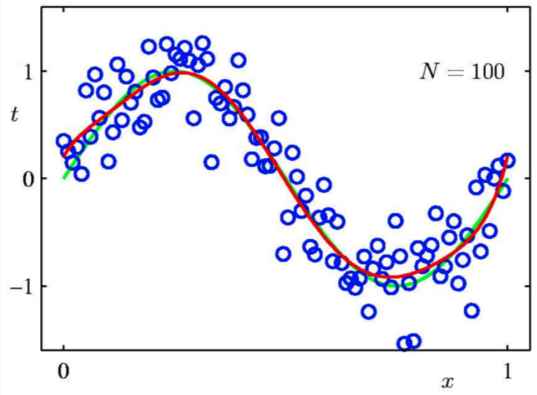


Figure 2 – Fixing overfitting by providing more data

As we can see in Figure 1, our curve fits all the data, the Empirical Risk is minimized as it is equal to 0, however the red curve does not fit the green one. This is what we call overfitting. In this case, it is due to our hyper-parameter M that too big compared to the number of data set points we have.

An idea to fix overfitting, while training the model, is to provide more data points as we can see in the Figure 2. This gives significant result, however, it is not always possible to have more data.

In this case, we can also use a regularization term additionally to our Empirical Risk. This aims at putting additional constraints on the parameters to promote simple models over complex ones. In this particular example, this would lower M and provide a better prediction curve.

Finally, the best case way to validate a model stay the testing. Usually, we use the train-test split method to keep a part of our data set for testing as we never test the model with training data !

3.3 True Risk

As we saw in the previous section, the Empirical Risk is not a good indicator to show if the model is correct or not ; it was equal to 0 as the curve was not fitting the true one in Figure 1.

Measuring performance of a classifier : Necessary assumptions

— We are interested in the performance of our classifier on future data coming from the same source. What does "same source" mean ?

— In machine learning, we make the following weak assumption : we assume there exists an unknown data generating distribution P over $\mathcal{X} \times \mathcal{Y}$.

— All input/output pairs (x, y), including pairs from the data set and future data, are generated i.i.d. from a distribution P

Definition 4. *The risk of a prediction function $f : X \to \hat{y}$ is*

$$R(f) = \mathbb{E}_{X,Y}[\ell(f(X), Y)] = \int_{X,y} \ell(f(X), Y) dP(X, Y)$$

In words, it is the expected loss of f on a new example (X, Y) drawn randomly from P, i.e. having the same distribution as our data set.

Remark

— What we really want is a classifier minimizing the true risk.

— In general, the true risk of a function cannot be computed.

— ERM is good if and only if the true risk is close to the empirical risk.

Risk vs Empirical risk

Let $S = ((x_1, y_1), \ldots, (x_N, y_N))$ be drawn independent and identically distributed from P.

Let's draw some inspiration from the Strong Law of Large Numbers : If $z, z_1, \ldots, z_n$ are i.i.d. with expected value $\mathbb{E}z$, then with probability one,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} z_i = \mathbb{E}z$$

By the Strong Law of Large Numbers, if f is independent from S then

$$\lim_{N \to \infty} \hat{R}_N(f) = R(f)$$

ERM seeks to find the $\hat{f}$ that minimizes $\hat{R}_N(f)$ with a strong dependency on the data set S, so independence does not hold in general. Thus, $\hat{R}_N(\hat{f}) \nrightarrow R(\hat{f})$.

Recap on ERM

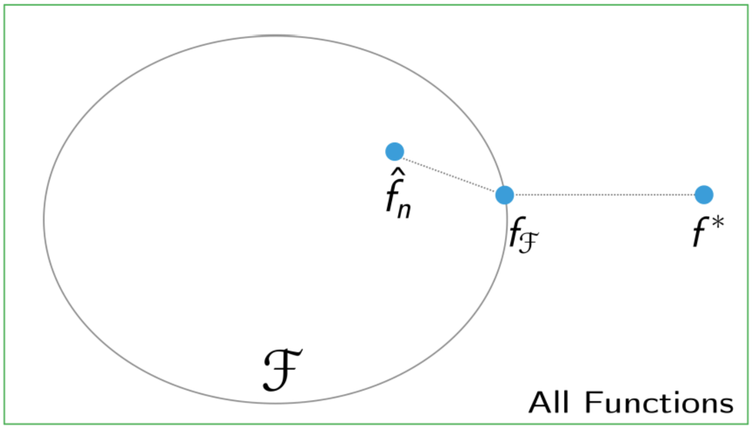
— The risk that interests us is the True Risk

— We saw ERM that can overfit because it optimizes $\hat{R}(f)$ which can be far from $R(f)$.

— So we can wonder : If the number of examples N is big enough, will $R(\hat{f})$ converge to the best possible classifier in the class $\mathcal{F}$? What is the "best classifier" ?

4 Decomposing the risk of ERM

FIGURE 3 – Decomposition of the Risks



The Figure 3 illustrates well the decomposition of the Risks. First, we make as we can only output measurable function from the learner, this leads to the Bayes error. We make another error by supposing that $f \in \mathcal{F}$, we restrain the learner on a certain class of function, this leads to approximation error. Finally, because we are using ERM algorithms, we make another error, the estimation error.

$$f^* = \arg\min_{f} \mathbb{E}_{X,Y} \ell(f(X), Y)$$

$$f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{X,Y} \ell(f(X), Y)$$

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f\left(x_i\right), y_i\right)$$

Definition 5. *Approximation Error (of $\mathcal{F}$) $= R\left(f_{\mathcal{F}}\right) - R\left(f^*\right)$*

Definition 6. *Estimation error (of $\hat{f}_n$ in $\mathcal{F}$) $= R\left(\hat{f}_n\right) - R\left(f_{\mathcal{F}}\right)$*

4.1 Error Decomposition

As we saw, the risk of ERM $\hat{f}_n$ can be decomposed :

$$R\left(\hat{f}_n\right) = \underbrace{R\left(\hat{f}_n\right) - R\left(f_{\mathcal{F}}\right)}_{\text{estimation error}} + \underbrace{R\,(f_{\mathcal{F}}^{\mathcal{F}} - R\left(f^*\right)}_{\text{approximation error}} + \underbrace{R\left(f^*\right)}_{\text{bayes error}}$$

Thus, by taking a bigger $\mathcal{F}$ the approximation error decreases but the estimation error increases.

4.2 The Bayes Predictor

Definition 7. *A Bayes predictor $f^* : X \to \hat{y}$ is a function that achieves the minimal risk among all measurable functions :*

$$f^* \in \underset{f \in \{ \text{ measurable functions } \}}{\arg\min} R(f),$$

where the minimum is taken over all functions from $\mathcal{X}$ to $\hat{\mathcal{Y}}$.

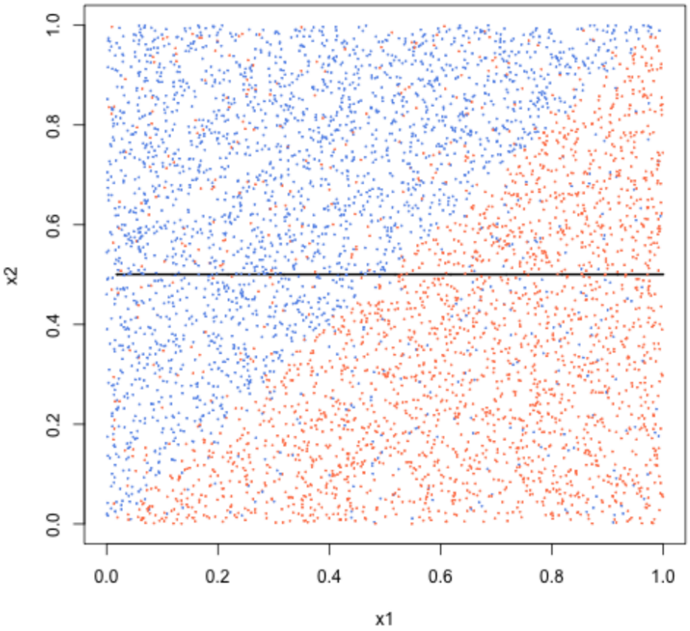
Remark

— The risk of a Bayes prediction function is called the Bayes risk.

— A Bayes prediction function is often called the **target function** : it is the best possible function we can possibly produce

An illustration of the notion of Bayes risk with $\ell^{0/1}$

FIGURE 4 – Decomposition of the Risks



$$y = \{\text{blue, orange }\}$$

$$P_x = \text{ Uniform } \left([0,1]^2\right)$$

$$\mathbb{P} \left(\text{orange } \mid x^{(1)} > x^{(2)}\right) = 0.9$$

$$\mathbb{P} \left(\text{ orange } \mid x^{(1)} \leqslant x^{(2)}\right) = 0.1$$

$$\text{Risk of } f(x) = \begin{cases} \text{orange} & \text{if } x^{(1)} > x^{(2)} \\ \text{blue} & \text{otherwise} \end{cases}$$

The idea here is to split the cloud points into 4 quarters.

Note that each quarter of our data set can be said to be equiprobable, because S was drawn i.i.d. We write Q_i the i-th quarter of the figure. By the law of total probability, we have :

$$\mathbb{P}(error) = \mathbb{P}\left(\text{ error } \mid Q_1\right)\mathbb{P}\left(Q_1\right) + \mathbb{P}(\text{ error } \mid Q2)\mathbb{P}\left(Q_2\right) + \mathbb{P}\left(\text{ error } \mid Q_3\right)\mathbb{P}\left(Q_3\right) + \mathbb{P}\left(\text{ error } \mid Q_4\right)\mathbb{P}\left(Q_4\right)$$

Thus :

$$\mathbb{P}(error) = 0.5 \times 0.25 + 0.1 \times 0.25 + 0.5 \times 0.25 + 0.1 \times 0.25) = 0.3$$

Each quarter represents a $\frac{1}{4}$ portion of the hole data set, and the probability of an error in a quarter depends on the distribution of blue and red points.

Finally, the Approximation Error for $\mathcal{F}_1$ (the class function of 1 class classifier is $\mathcal{R}\left(f_{\mathcal{F}}\right) - \mathcal{R}\left(f^*\right) = 0.3 - 0.1 = 0.2$

4.2.1 The Expectation and its properties : reminder

Lemma 1. *For any predicate C, we have $\mathbb{E}[1(C)] = \mathbb{P}(C)$ where $1(\cdot)$ is the indicator function*

Démonstration. $\mathbb{E}[1(C)] = 1 \times \mathbb{P}(C) + 0 \times (1 - \mathbb{P}(C)) = \mathbb{P}(C)$ □

As usual, $\mathcal{X}$ and $\mathcal{Y}$ are the domain of the random variables X and Y. Let $g(X, Y)$ be an arbitrary real-valued function. - If X and y are continuous spaces, assuming their distribution P admits a joint probability density function $p(X, Y)$, then :

- The expectation of $g(X, Y)$ is :

$$\mathbb{E}_{X,Y}[g(X, Y)] = \int_{X \times Y} g(X, Y) p(X, Y) dX dY$$

- The conditional expectation of $g(X, Y)$ given X is

$$\mathbb{E}_Y[g(X, Y) \mid X] = \int_y g(X, Y) p(Y \mid X) dY$$

- Finally, the Law of total expectation [1] states that :

$$\mathbb{E}_{X,Y}[g(X, Y)] = \mathbb{E}_X\left[\mathbb{E}_Y[g(X, Y) \mid X]\right]$$

- The Law of total expectation states that :

$$\mathbb{E}_{X,Y}[g(X, Y)] = \mathbb{E}_X\left[\mathbb{E}_Y[g(X, Y) \mid X]\right]$$

Démonstration.

$$\begin{aligned}
\mathbb{E}_{X,Y}[g(X,Y)] &= \int_{X\times y} g(X,Y)p(X,Y)dXdY \\
&= \int_{X\times y} g(X,Y)p(Y \mid X)p(X)dXdY \\
&= \int_{X} \left(\int_{y} g(X,Y)p(Y \mid X)dY \right) p(X)dX \\
&= \mathbb{E}_{X}\left[\mathbb{E}_{Y}[g(X,Y) \mid X]\right]
\end{aligned}$$

□

- If X is continuous and y discrete, and if $p(X,Y)$ is there joint density function, then :

- The expectation of $g(X,Y)$ is :

$$\mathbb{E}_{X,Y}[g(X,Y)] = \sum_{Y\in Y} \int_{X} g(X,Y)p(X,Y)dX$$

- The conditional expectation over Y of $g(X,Y)$ given X is :

$$\mathbb{E}_{Y}[g(X,Y) \mid X] = \sum_{Y\in Y} g(X,Y)p(Y \mid X)dY$$

- Finally, the Law of total expectation still is (same proof) :

$$\mathbb{E}_{X,Y}[g(X,Y)] = \mathbb{E}_{X}\left[\mathbb{E}_{Y}[g(X,Y) \mid X]\right]$$

4.2.2 Bayes Predictor for Binary Classifier

Spaces are $\hat{\mathcal{Y}} = \mathcal{Y} = \{0,1\}$ and we use the $0-1$ loss :

$$\ell(\hat{y},y) = \mathbf{1}(\hat{y} \neq y) := \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases}$$

Theorem 1. *In this case, the Risk is :*

$$\begin{aligned}
R(f) &= \mathbb{E}[1(f(X) \neq Y)] = 0 \cdot \mathbb{P}(f(X) = Y) + 1 \cdot \mathbb{P}(f(X) \neq Y) \\
&= \mathbb{P}(f(X) \neq Y)
\end{aligned}$$

$$\begin{aligned}
R(f) &= \mathbb{E}[1(f(X) \neq Y)] = 0 \cdot \mathbb{P}(f(X) = Y) + 1 \cdot \mathbb{P}(f(X) \neq Y) \\
&= \mathbb{P}(f(X) \neq Y)
\end{aligned}$$

Which is just the misclassification error rate.

Bayes prediction function is just the assignment to the most likely class :

$$f^*(x) \in \underset{c\in\{0,1\}}{\arg\max} P(Y = c \mid X = x)$$

Démonstration. We define $\eta(X) = P(Y = 1 \mid X)$

$$\begin{aligned} R(f) &= \mathbb{E}_{X,Y}[1(f(X) \neq Y)] \\ &= \mathbb{E}_X\left[\mathbb{E}_Y[1(f(X) \neq Y) \mid X]\right] \\ &= \mathbb{E}_X[P(f(X) \neq Y \mid X)] \\ &= \mathbb{E}_X[P(Y = 0 \mid X) \cdot 1(f(X) = 1) + P(Y = 1 \mid X) \cdot 1(f(X) = 0)] \\ &= \mathbb{E}_X[(1 - \eta(X)) \cdot 1(f(X) = 1) + \eta(X) \cdot 1(f(X) = 0)] \end{aligned}$$

- Thus, the Bayes Predictor f^* which minimizes $R(\cdot)$ will be

$$f^*(x) = \begin{cases} 1 & \text{if } 1 - \eta(x) \leqslant \eta(x) \\ 0 & \text{otherwise.} \end{cases} \quad \underset{c \in \{0,1\}}{\arg\max} P(Y = c \mid X = x)$$

- And $R\left(f^*\right) = \mathbb{E}_X[\min(\eta(X), 1 - \eta(X))]$ □

4.2.3 Bayes Predictor for Least Squares Regression

Bayes Predictor for Least Squares Regression - Spaces : $\hat{\mathcal{Y}} = \mathcal{Y} = \mathbb{R}$

- Square loss :

$$\ell(\hat{y}, y) = (\hat{y} - y)^2$$

Theorem 2. *In this case, the Risk is :*

$$\begin{aligned} R(f) &= \mathbb{E}\left[(f(X) - Y)^2\right] \\ &= \mathbb{E}\left[(f(X) - \mathbb{E}[Y \mid X])^2\right] + \mathbb{E}\left[(Y - \mathbb{E}[Y \mid X])^2\right] \end{aligned}$$

Démonstration. The Bayes Risk in the regression case :

$$\begin{aligned} R(f) &= \mathbb{E}_{X,Y}\left[(f(X) - Y)^2\right] \\ &= \mathbb{E}_X\left[\mathbb{E}_Y\left[(f(X) - Y)^2 \mid X\right]\right] \\ &= \mathbb{E}_X\left[\mathbb{E}_Y\left[(f(X) - \mathbb{E}[Y \mid X] + \mathbb{E}[Y \mid X] - Y)^2 \mid X\right]\right] \\ &= \mathbb{E}_X\left[\mathbb{E}_Y[(f(X) - \mathbb{E}[Y \mid X])^2 + (\mathbb{E}[Y \mid X] - Y)^2 + \underbrace{(f(X) - \mathbb{E}[Y \mid X])(\mathbb{E}[Y \mid X] - Y)}_{=0} \mid X]\right. \end{aligned}$$

The third term is null because given $X, (f(X) - \mathbb{E}[Y \mid X])$ is constant and $\mathbb{E}_Y[\mathbb{E}[Y \mid X] - Y \mid X]$ is null. Thus, the Bayes predictor f^* which minimizes $R(\cdot)$ will be :

$$f^*(X) = \mathbb{E}[Y \mid X = X]$$

and

$$R\left(f^*\right) = \mathbb{E}_X\left[\mathbb{E}_Y\left[(\mathbb{E}[Y \mid X] - Y)^2 \mid X\right]\right] = \mathbb{E}_X[\operatorname{var}(Y \mid X)]$$

□

5 Analyze of the Risk of the 1-nearest neighbor

Let $h_S^{NN}(x)$ be the 1-nearest neighbor (1NN) classifier taking neighbors from S. Does $\mathcal{R}\left(h_S^{NN}(x)\right) \to \mathcal{R}\left(f^*\right)$ when N tends to infinity.

Important simplification :

We will consider the two class setting where $\mathbb{P}(y = 1 \mid X) = P(y = 1)$.

We study $\mathbb{E}_S\left[\mathcal{R}\left(h_S^{NN}(x)\right)\right]$.

Where $S \sim P_S$ is the distribution of our data set.

We define Y_{NN} as the class of the Nearest Neighbor of x in S

$$\mathbb{E}_S\left[\mathcal{R}\left(h_S^{NN}(x)\right)\right] = \mathbb{E}_{S\sim P_S}\left[\mathbb{E}_X\left[\mathbb{P}_Y(y = 0 \mid x) \cdot \mathbf{1}\left(Y_{NN} = 1\right) + \mathbb{P}_Y(y = 1 \mid x) \cdot \mathbf{1}\left(Y_{NN} = 0\right)\right]\right]$$

$\mathbb{P}(y = 0 \mid x)$ does not depend on S, and S only affects the Nearest Neighbor calculation, and can thus reposition the expectations.

$$\begin{aligned}
\mathbb{E}_S\left[\mathcal{R}\left(h_S^{NN}(x)\right)\right] &= \mathbb{E}_X\left[\mathbb{P}(y = 0 \mid x) \cdot \mathbb{E}_S\left[\mathbf{1}\left(Y_{NN} = 1\right) \mid x\right] + \mathbb{P}(y = 1 \mid x) \cdot \mathbb{E}_S\left[\mathbf{1}\left(Y_{NN} = 0\right) \mid x\right]\right] \\
&= \mathbb{E}_X\left[\mathbb{P}(y = 0 \mid x) \cdot \mathbb{P}\left(Y_{NN} = 1 \mid x\right) + \mathbb{P}(y = 1 \mid x) \cdot \mathbb{P}\left(Y_{NN} = 0 \mid x\right)\right] \\
&= \mathbb{E}_X[p(1-p) + (1-p)p] \\
&= 2p(1-p)
\end{aligned}$$

Thus, as $N \to \infty$,

$$\mathbb{E}_S\left[\mathcal{R}\left(h_S^{NN}(x)\right)\right] \to 2p(1-p)$$

However if we do not make theses assumptions, in the general case, we have :

$$\mathbb{E}_S\left[\mathcal{R}\left(h_S^{NN}(x)\right)\right] \to \mathbb{E}_X[2\mathbb{P}(y = 1 \mid x)(1 - \mathbb{P}(y = 1 \mid x))]$$

Recalling the Bayes Error of in binary classification, it comes :

$$\mathcal{R}\left(f^*\right) = \mathbb{E}_X\left[\min\left(P_Y(Y = 1 \mid X), P_Y(Y = 0 \mid X)\right)\right]$$

Thus 1NN is not Bayes Consistent, indeed the True Risk does not converge to the Bayes Risk.

6 Conclusion

We saw that the best way to compute the performance of a classifier is by using the True Risk. However, it is not easy, in general, to calculate it. Besides, our learning models can not perfectly fit any data set having the same distribution as the data set we are training on. Thus, the goal is to be closer as possible to the target function and thus closer as possible to the Bayes Error.