SL : FOUNDATIONS

I. INTRODUCTION

- **Learning** : " A process based on experience in which a subject (the learner) improves his understanding of some phenomenon or his ability to design an adequate response when facing a new situation ".

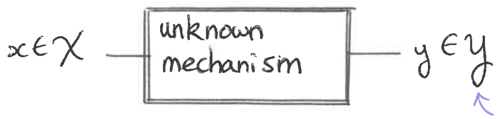
  ↖ inference step : infer general properties from particular examples

- In the SUPERVISED LEARNING framework, the available experience gathered is represented by a collection of pairs $\{(x_1, y_1), \dots, (x_n, y_n)\}$, called the LEARNING SAMPLE.

  the x_i's belong to some set $\mathcal{X}$, called the INPUT SPACE, or the FEATURE SPACE ; often a subset of $\mathbb{R}^d$, $d \geq 1$. [d = number of features / dim of $\mathcal{X}$]

  To each x_i corresponds an $y_i \in \mathcal{Y} \subset \mathbb{R}$, called the RESPONSE VARIABLE, or the LABEL of x_i.

- The goal of a supervised learning task is to guess the unknown label y of a new unlabelled feature point x, given only the knowledge of the learning sample.



↖ Depending on the nature of $\mathcal{Y}$, we are facing a different problem :

(i) $\mathcal{Y} = \{0, 1\}$ (or $\{-1, 1\}$) (or $\{1, 2, \dots, K\}$) = CLASSIFICATION TASK : each item is assigned a category (if two categories : we have BINARY CLASSIF)

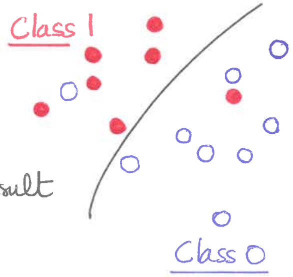
× **Examples** :

↘ text or document classification
x = frequency of words in a document
y = topic, such as sports, business, music, ...

↘ image classification
x = pixel values
y = categories such as landscape, portrait, animal, ...

↘ digit recognition
x = pixel values
y = the digit

↘ medical diagnosis
x = patient medical test result
y = patient disease state



(ii) $\mathcal{Y} = \mathbb{R}$ (or a subset of $\mathbb{R}$) = REGRESSION PROBLEM : predict a real value for each feature point.

× **Examples** :

↘ x = body measurements (weight, age, ...)
y = body fat level

↘ x = air pollution concentration measurements & other indicators
y = housing value

↘ x = environmental variables (temp & salinity of water...)
y = species abundance and richness

Other learning tasks include :

- RANKING : order items according to some criterion (Ex : web search)
- CLUSTERING : partition feature points into distinct regions : an unsupervised learning (UL) task : no labels.
- DIMENSION REDUCTION : reduce the dimensionality d of

the input space $\mathcal{X} = \mathbb{R}^d$. Often used as a pre-processing step in a supervised learning task; or for data compression.

In addition to the supervised & unsupervised learning framework, common machine learning tasks also include:

→ SEMI-SUPERVISED LEARNING : the learner receives both labeled & unlabeled training data. A common scenario in applications where unlabeled data is easily accessible, while labeled data is expensive. (examples include webpage classification: manually labeling a webpage is long & tedious, while millions of web pages are available)

→ ON-LINE LEARNING : the online -scenario involves multiple rounds. At each round the learner receives an unlabeled data, makes a prediction, and then finds out the true label. The goal is to minimize the total cumulative loss over all rounds (instances & their labels may be chosen in an adversarial way) (example: spam e-mail filtering; computational advertising)

→ REINFORCEMENT LEARNING: the learner takes actions in an environment in order to maximize some cumulative reward. The focus is on on-line performance, and the learner is faced with the exploration vs exploitation dilemma: exploring unknown actions & gain knowledge vs exploiting information already collected (multi-armed bandit problem)

→ ACTIVE LEARNING : a particular case of semi-supervised learning, where the learner is able to query an oracle to obtain the desired label of an unlabeled feature point x (computational biology applications). The task is to decide which data points should be labeled.

The role of probability theory & mathematical statistics : the inference step in the learning process leads to consider the idea of modeling the unknown. Similarity between a new pair (x,y) and existing pairs $(x_1,y_1),\dots,(x_n,y_n)$ is understood in a probabilistic way, assuming that (x,y) and the (x_i,y_i)s are drawn independently from the same (unknown) probability distribution. The inference step requires the estimation of this unknown probability & standard statistical techniques apply.

- A simple model : the input points are assumed to be independently drawn from a common probability distribution $\mathbb{P}_X$. As to the labels, we assume the existence of a correct labeling function $f : \mathcal{X} \to \mathcal{Y}$, unknown to the learner.

- A more realistic model : Pairs (x_i, y_i) & (x,y) are independently drawn from a joint distribution $\mathbb{P}_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$. In other words, $(x_1,y_1),\dots,(x_n,y_n)$ and (x,y) are observed values (aka realizations) of independent and identically distributed (iid) random variables $(X_1,Y_1),\dots,(X_n,Y_n)$ and (X,Y). In the remainder, we refer to the collection of RVs

$$\mathcal{L}_n = \{(X_1,Y_1),\dots,(X_n,Y_n)\}$$

as the LEARNING SAMPLE. We write $(X_i,Y_i) \sim P_{X,Y}$

Task: predicting the unknown (and random) label Y of a new feature point X. This task is carried out by constructing a PREDICTOR $f_n : \mathcal{X} \to \mathbb{R}$, based on $\mathcal{L}_n$, where it is understood that $f_n(X)$ stands as a guess for the unknown label Y of X. Note that in the case of binary classification f_n takes (or can take) values in $\mathbb{R}$, instead of $\{0,1\}$, for reasons that will become clear later

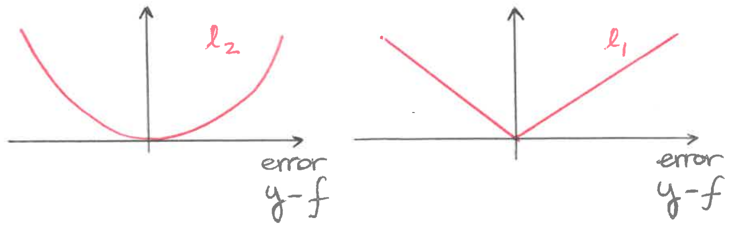
Let $f : \mathcal{X} \to \mathbb{R}$ be a given predictor (for now, f is not necessarily constructed using $\mathcal{L}_n$; but just some function $\mathcal{X} \to \mathcal{Y}$). We assess the performance / accuracy of f by means of a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}_+$, which quantifies the cost of predicting $f(x)$ when the true outcome is y, given by $\ell(y, f(x))$

we will often abuse notation, and write $\ell(y, f)$

Examples:

(i) Square loss. Useful with continuous labels. It is defined as $\ell_2(y,f) = (y-f)^2$ = square of the error $y - f(x)$. A popular choice in many regression problems.

(ii) A more exotic loss is the p-power loss $\ell_p(y,f) = |y-f|^p$, and in particular, the absolute loss $\ell_1(y-f) = |y-f|$.



(iii) For binary labels $y \in \{0,1\}$, a suitable choice is the 0-1 loss, defined by $\ell_{0\text{-}1}(y,f) = \mathbb{1}(y \neq f)$

taking values in $\{0,1\}$

$$= \begin{cases} 1 & \text{if } y \neq f \\ 0 & \text{otherwise.} \end{cases}$$

Alternatively, an assymetric loss function may be used. For example, in spam classification, we prefer to label a spam email as legitimate, than the other way around.

A suitable loss function would be:

$$\ell(y,f) = \begin{cases} 10 & \text{if } y=0,\ f(x)=1 \\ 1 & \text{if } y=1,\ f(x)=0 \\ 0 & \text{otherwise}, \end{cases}$$

where $y=1$ denotes a spam, and $y=0$ otherwise.

(iv) Multiclass classification problems: each training point belongs to one of K categories. The response variable Y belongs to $\mathcal{Y} = \{0,1\}^K$: $Y = (Y_1, .., Y_K)$, where $Y_k = 1$ if and only if X belongs to category k, and zero otherwise.

Put • $p_k(x) := \mathbb{P}(Y_k = 1 \mid X = x)$

$$= \frac{\exp\{f_k(x)\}}{\sum_{j=1}^{K} \exp\{f_j(x)\}}, \quad (k = 1, .., K)$$

Then classify a new x for which $p_k(x)$ is the largest

convenient notation since $p_k(x) \geq 0$, and must sum to one.

• $f(x) = (f_1(x), \dots, f_K(x))$.

Use the multinomial deviance (minus the log likelihood) as a loss function:

$$\ell(y,f) = -\sum_{k=1}^{K} y_k \log p_k(x)$$

$$= -\sum_k y_k f_k(x) + \underbrace{\sum_k y_k}_{=1} \log\Big(\sum_j e^{f_j(x)}\Big)$$

$$= -\sum_{k=1}^{K} y_k f_k(x) + \log\Big(\sum_{j=1}^{K} \exp\{f_j(x)\}\Big).$$

The task of finding $f = (f_1, .., f_K)$ minimizing ℓ is usually under the additional constraint that $\sum_{j=1}^{K} f_j = 0$ (since adding a constant value to all the f_j does not change p_k).

The quantity $\ell(Y, f(X))$ is a random variable, and depends on the value taken by (X, Y). It may be large for some values of (X, Y), but small for others. To decide if f does a good job or not, you may select f so that $\ell(Y, f(X))$ remains small with high probability. A more tractable approach is to construct f such that $\ell(Y, f(X))$ is small on average, and define the EXPECTED LOSS or RISK as

$$\boxed{R(f) = \mathbb{E}\,\ell(Y, f(X))} = \int \ell(y, f(x))\, d\mathbb{P}_{X,Y}(x,y)$$

RISK of f.

We are interested in functions f with small risk, and in particular in the function f^* with the smallest risk (over the space of all measurable functions)

$$\boxed{f^* \in \underset{f}{\operatorname{argmin}}\, R(f) \quad ; \text{ with risk } R^* := R(f^*).}$$

'argmin' should be understood as follows

$$\underset{f}{\operatorname{argmin}}\, R(f) = \left\{ f \text{ measurable} : R(f) = \min_{g \text{ meas.}} R(g) \right\}$$

The minimizer is not necessarily unique.

The optimal risk R^* stands as a benchmark to which the risk of any predictor f should be compared. In particular, we are interested in the EXCESS RISK,

$$\boxed{\mathcal{E}(f) := R(f) - R^*}$$

EXCESS RISK of f.

We next derive the expression of the optimal predictor f^* in two important cases: regression problem with square loss, and binary classification task with 0-1 loss.

- Optimality of the regression function.

Theorem Let $(X, Y) \sim \mathbb{P}_{X,Y}$, such that $\mathbb{E}\,Y^2 < \infty$, and let $r(x) := \mathbb{E}(Y \mid X = x)$ be the conditional expectation of Y given $X = x$.
Then r minimizes the quadratic risk: $R(r) = R^*$,
where
$$R(f) := \mathbb{E}\{(Y - f(X))^2\} = \mathbb{E}\{\ell_2(Y, f(X))\}, \quad f \text{ measurable}.$$

proof: Let $f : \mathcal{X} \to \mathcal{Y}$ measurable, such that $f(X)$ is square integrable. Then

$$\begin{aligned}
R(f) &= \mathbb{E}\{(Y - f(X))^2\} \\
&= \mathbb{E}_X \mathbb{E}\{(Y - f(X))^2 \mid X\} \\
&= \mathbb{E}_X \mathbb{E}\{(Y - r(X) + r(X) - f(X))^2 \mid X\}
\end{aligned}$$

$r(X) = \mathbb{E}(Y \mid X)$

$$\begin{aligned}
&= \mathbb{E}_X \mathbb{E}\{(Y - r(X))^2 \mid X\} \\
&\quad + 2\,\mathbb{E}_X \overbrace{\mathbb{E}\{(Y - r(X))(r(X) - f(X)) \mid X\}}^{=0} \\
&\quad + \mathbb{E}_X \mathbb{E}\{(f(X) - r(X))^2 \mid X\} \\
&= \mathbb{E}\{(Y - r(X))^2\} + \mathbb{E}\{(f(X) - r(X))^2\} \\
&= R(r) + \underbrace{\int (f(x) - r(x))^2\, d\mathbb{P}_X(x)}_{}
\end{aligned}$$

non-negative, and equal to zero if and only if $f(x) = r(x)$, which concludes the proof ▦

× Remark: For the choice of a square loss, the excess risk $\mathcal{E}(f)$ takes a convenient form. We established in the proof of the theorem that for a fixed function f,

$$\mathcal{E}(f) = R(f) - R(r) = \int_{\mathbb{R}^d} (f(x) - r(x))^2 \, \mathbb{P}_X(dx)$$

$$= \mathbb{E}\{(f(X) - r(X))^2\}$$

$$= \mathbb{E}\{\ell_2(f(X), r(X))\}$$

$= $ expected ℓ_2-error of f.

× Geometrical considerations:

Let $\mathcal{L}^2 =$ space of square integrable RVs, endowed with the inner product $\langle X, Y\rangle := \mathbb{E}(XY)$, and norm $\|X\|^2 = \langle X, X\rangle$.

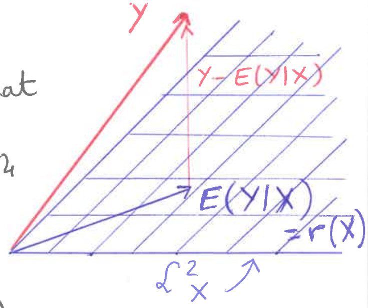
The distance between X and Y is $d(X,Y) = \|X - Y\|^2$

$$= \mathbb{E}(X-Y)^2.$$

Consider $\mathcal{L}^2_X \subset \mathcal{L}^2$, the subspace of $\mathcal{L}^2$ consisting of all square integrable functions of X: $\mathcal{L}^2_X := \{\varphi(X) \mid \mathbb{E}\varphi(X)^2 < \infty\}$.
$X \in \mathcal{L}^2$

The best element in $\mathcal{L}^2_X$ approximating Y is the one that minimizes $d(Y, \varphi(X)) = \mathbb{E}(Y - \varphi(X))^2$, and is given by the conditional expectation $\mathbb{E}(Y|X)$, since we saw that

$$\mathbb{E}(Y - \varphi(X))^2$$
$$= \mathbb{E}(Y - r(X))^2 + \mathbb{E}(r(X) - \varphi(X))^2$$



⇒ 'vectors' $\mathbb{E}(Y|X)$ and $Y - \mathbb{E}(Y|X)$ are orthogonal: $\mathbb{E}(Y|X)$ is the orthogonal projection of Y onto $\mathcal{L}^2_X$.

- Optimality in binary classification.

In binary classification with 0-1 loss $\ell(y,f) = \mathbb{1}(y \neq f)$, the risk of f is

$$R(f) = \mathbb{E}\,\ell(Y, f(X)) = \mathbb{E}\,\mathbb{1}(Y \neq f(X)) = \mathbb{P}(Y \neq f(X))$$

Risk minimization ⇔ Minimizing the misclassification probability

Def = BAYE'S RISK is the infimum of the risk of all classifiers $R^* = \inf_f R(f) = \inf_f \mathbb{P}(Y \neq f(X))$.

Theorem: Let $(X, Y) \sim \mathbb{P}_{X,Y}$, where $Y \in \{0,1\}$.
We define BAYE'S CLASSIFIER as

$$f^*(x) := \begin{cases} 1 & \text{if } r(x) \geqslant 1/2 \\ 0 & \text{otherwise,} \end{cases}$$

where $r(x) = \mathbb{P}(Y=1 \mid X=x) = \mathbb{E}(Y \mid X=x)$

Then $R(f^*) = R^* =$ Baye's risk.

proof: Let $f: \mathcal{X} \to \mathcal{Y}$ be a measurable function.
We want to show that $R(f) - R(f^*) \geqslant 0$.
Since

$$R(f) - R(f^*) = \mathbb{P}(Y \neq f(X)) - \mathbb{P}(Y \neq f^*(X))$$

$$= \int_{\mathcal{X}} \{\mathbb{P}(Y \neq f(x) \mid X=x) - \mathbb{P}(Y \neq f^*(x) \mid X=x)\}\, \mathbb{P}_X(dx)$$

It is enough to show that $\forall x$, the integrand is non-negative.
First, note that

$$\mathbb{P}(Y \neq f(x) \mid X=x)$$
$$= 1 - \mathbb{P}(Y = f(x) \mid X=x)$$
$$= 1 - \{\mathbb{P}(Y=1, f(x)=1 \mid X=x) + \mathbb{P}(Y=0, f(x)=0 \mid X=x)\}$$

$$
\begin{aligned}
&= 1 - \left\{ \mathbb{E}\left[\mathbb{1}(Y=1)\mathbb{1}(f(x)=1) \mid X=x \right] + \mathbb{E}\left[\mathbb{1}(Y=0)\mathbb{1}(f(x)=0) \mid X=x \right] \right\} \\
&= 1 - \left\{ \mathbb{1}(f(x)=1)\, \mathbb{P}(Y=1 \mid X=x) + \mathbb{1}(f(x)=0)\, \mathbb{P}(Y=0 \mid X=x) \right\} \\
&= 1 - \left\{ \mathbb{1}(f(x)=1)\, r(x) + \mathbb{1}(f(x)=0)(1-r(x)) \right\}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\mathbb{P}(Y \neq f(X) \mid X=x) - \mathbb{P}(Y \neq f^*(X) \mid X=x) \\
&= r(x) \left\{ \mathbb{1}(f^*(x)=1) - \mathbb{1}(f(x)=1) \right\} \\
&\quad + (1-r(x)) \left\{ \mathbb{1}(f^*(x)=0) - \mathbb{1}(f(x)=0) \right\} \\
&= r(x) \left\{ \mathbb{1}(f^*(x)=1) - \mathbb{1}(f(x)=1) \right\} \\
&\quad - (1-r(x)) \left\{ \mathbb{1}(f^*(x)=1) - \mathbb{1}(f(x)=1) \right\} \\
&= \underset{\uparrow \atop \in [0,1]}{(2r(x)-1)} \left\{ \mathbb{1}(f^*(x)=1) - \mathbb{1}(f(x)=1) \right\}
\end{aligned}
$$

If x is such that $r(x) \geqslant 1/2$, then $2r(x)-1 \geqslant 0$, and

$$\underbrace{\mathbb{1}(f^*(x)=1)}_{=1} - \underbrace{\mathbb{1}(f(x)=1)}_{0 \text{ or } 1} \geqslant 0$$

If x is such that $r(x) < 1/2$, then $2r(x)-1 < 0$, and

$$\underbrace{\mathbb{1}(f^*(x)=1)}_{=0} - \underbrace{\mathbb{1}(f(x)=1)}_{0 \text{ or } 1} \leqslant 0$$

In both cases, the product $(2r(x)-1)\left\{ \mathbb{1}(f^*(x)=1) - \mathbb{1}(f(x)=1) \right\}$ is non-negative, which concludes the proof. ▨

Put $\mathcal{E}_{0\text{-}1}(f) := R(f) - R^*$ = excess risk under 0-1 loss

x **Remark**: The previous result establishes a tight relationship between the regression problem and binary classification: the value of $r(x) = \mathbb{E}(Y \mid X=x)$ (larger or smaller than $1/2$) provides an optimal predictor in the context of binary classification.

$\Rightarrow$ if $f(x)$ is a good predictor in a regression context (i.e. if f is close to r in quadratic mean), it is natural to ask whether its value compared to $1/2$ provides a good predictor as well in a binary classification task. The following theorem shows that the answer to this question is positive.

Theorem: Let $f : \mathcal{X} \to \mathbb{R}$ be a measurable function,

Put
$$g(x) = \begin{cases} 1 & \text{if } f(x) \geqslant 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$
\begin{aligned}
\mathcal{E}_{0\text{-}1}(g) &= \mathbb{P}(Y \neq g(X)) - \mathbb{P}(Y \neq f^*(X)) \\
&\leqslant 2 \int_{\mathcal{X}} | f(x) - r(x) | \, \mathbb{P}_X(dx) \\
&\leqslant 2 \left(\int_{\mathcal{X}} (f(x) - r(x))^2 \, \mathbb{P}_X(dx) \right)^{1/2} \\
&= 2 \sqrt{\mathcal{E}(f)}
\end{aligned}
$$

Baye's classifier (page 10)

$r(x) = \mathbb{E}(Y \mid X=x) = \mathbb{P}(Y=1 \mid X=x)$

A small excess risk $\mathcal{E}(f)$ ensures a small excess risk $\mathcal{E}_{01}(g)$ for the associated binary classifier.

x **Consequence**: You may wish to model the joint distribution of (X, Y) (GENERATIVE MODELING) or simply the conditional distribution of $Y \mid X=x$ (DISCRIMINATIVE MODELING), in a parametric context.

proof =

$$\mathcal{E}_{0\text{-}1}(g) = \mathbb{P}(Y \neq g(X)) - \mathbb{P}(Y \neq f^*(X))$$

page 11

$$= \int (2r(x)-1)\{\mathbb{1}(f^*(x)=1) - \mathbb{1}(g(x)=1)\}\,\mathbb{P}_X(dx)$$

$$= \int |2r(x)-1| \times |\mathbb{1}(f^*(x)=1) - \mathbb{1}(g(x)=1)|\,\mathbb{P}_X(dx)$$

sign of $2r(x)-1$ and $\{\dots\}$ is the same ; see page 11

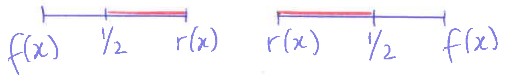
$$= 2\int \left|r(x)-\tfrac{1}{2}\right| \left|\mathbb{1}(f^*(x)=1) - \mathbb{1}(g(x)=1)\right| \mathbb{P}_X(dx)$$

$$= 2\int \left|r(x)-\tfrac{1}{2}\right| \left|\mathbb{1}(f^*(x) \neq g(x))\right| \mathbb{P}_X(dx)$$

If $f^*(x) \neq g(x)$, then

either $r(x) \geqslant \frac{1}{2}$, $f(x) < 1/2$

or $r(x) < 1/2$, $f(x) \geqslant 1/2$



$$\Rightarrow \left|r(x)-\tfrac{1}{2}\right| \leqslant |r(x) - f(x)|$$

distance in red

We get

$$\mathcal{E}_{0\text{-}1}(g) \leqslant 2\int |r(x)-f(x)| \overbrace{\mathbb{1}(f^*(x) \neq g(x))}^{\leqslant 1} \mathbb{P}_X(dx)$$

$$\leqslant 2\int |r(x)-f(x)|\,\mathbb{P}_X(dx)$$

$$= 2\,\mathbb{E}_X |r(X) - f(X)|$$

$$\leqslant 2\left(\mathbb{E}_X\{r(X)-f(X)\}^2\right)^{1/2}$$

Lyapunov inequality

∎

The bound is not tight : for g to be a good approximator of f^*, we only need f and r to be on the same side of the decision boundary : f and r do not need to be close together. This is often interpreted by saying that 'binary classification is easier than regression'.

II- LEARNING WITH DATA

The learning sample $\mathcal{L}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ is used to construct a predictor $f_n : \mathcal{X} \to \mathbb{R}$. More precisely, f_n is obtained as the output of a PREDICTION ALGORITHM $\mathcal{A}$:

$$\boxed{\mathcal{A} : \bigcup_{i \geqslant 1} (\mathcal{X} \times \mathcal{Y})^i \to \mathcal{M} := \text{set of all measurable functions } \mathcal{X} \to \mathbb{R}},$$

so that $\mathcal{A}(\mathcal{L}_n)$ is a function $\mathcal{X} \to \mathbb{R}$, and $\mathcal{A}(\mathcal{L}_n)(x)$ represents the predicted label for a new input point x, computed based on the algorithm $\mathcal{A}$, and observed values $(x_1, y_1), \dots, (x_n, y_n)$. When there is no confusion, we write f_n for $\mathcal{A}(\mathcal{L}_n)$

$f_n(x)$ is random, since it inherits the randomness of the random sample (even if x is fixed)

The risk of f_n is

$$\boxed{R(f_n) = \mathbb{E}\{\ell(Y, f_n(X)) \mid \mathcal{L}_n\}} = \mathcal{L}_n\text{- measurable RV}.$$

We aim at a small risk (with high probability), or small expected risk $\mathbb{E}\{R(f_n)\}$.

Alternatively, we look at the excess risk $\underset{\mathcal{E}(f_n)}{\underset{\parallel}{R(f_n) - R^*}}$, or at the expected excess risk $\mathbb{E}\{\mathcal{E}(f_n)\}$

$\mathbb{E}(.)$ taken over the distribution of $\mathcal{L}_n$.

II.1 Consistency.

A natural property of $\overset{\mathcal{A}(\mathcal{L}_n)}{f_n}$, is that its performance, as measured by the excess risk, improves as the size n of $\mathcal{L}_n$ increases. This leads to the notion of consistency:

Definition. A sequence of estimates $\{f_n\}$ is called

(i) WEAKLY CONSISTENT for a distribution $\mathbb{P}_{X,Y}$ if

$$\lim_{n\to+\infty} \mathbb{E}\{\mathcal{E}(f_n)\} = 0,$$

(ii) STRONGLY CONSISTENT for a distribution $\mathbb{P}_{X,Y}$ if

$$\lim_{n\to+\infty} \mathcal{E}(f_n) = 0, \quad \text{almost surely.}$$

Provided the excess risk remains bounded, as in binary classification, an application of the dominated convergence theorem shows that strong consistency implies weak consistency; hence the terminology. (see p.20 in PT: CONVERGENCE).

f_n may be consistent for some $\mathbb{P}_{X,Y}$, but not for another. A more interesting property would be consistency for any $\mathbb{P}_{X,Y}$:

Definition: A sequence of estimates $\{f_n\}$ is called

(i) WEAKLY UNIVERSALLY CONSISTENT if it is weakly consistent for all distributions $\mathbb{P}_{X,Y}$ of (X,Y) with $\mathbb{E}\,Y^2<\infty$.

(ii) STRONGLY UNIVERSALLY CONSISTENT if it is strongly consistent for all distributions $\mathbb{P}_{X,Y}$ of (X,Y) with $\mathbb{E}\,Y^2<\infty$.

Strongly universally consistent predictors f_n are of high interest since they guarantee good asymptotic performance in any situation. Such predictors do exist, see for example the K-nearest neighbour estimator, in the context of classification.

II.2. Minimax optimality.

Consistency is arguably a desired property of an estimator f_n. However, in practice, the sample size may be small, or convergence of the (expected) excess risk may be slow. Therefore, we are more interested in finite sample guarantees. One way is to identify a class $\mathcal{P}$ of distributions for (X,Y), and a sequence $\{e_n\}$ of positive numbers $e_n \downarrow 0$ as $n\to+\infty$, such that

$$\sup_{\mathbb{P}_{X,Y}\in\mathcal{P}} \mathbb{E}\{\mathcal{E}(f_n)\} \le e_n \quad \text{———} (*)$$

'sup' ≡ worse case scenario: the worse you can achieve in $\mathcal{P}$.

Alternatively, you may consider rates of convergence with high probability.

A possible way to define optimality of a predictor is provided by the MINIMAX point of view: a predictor f_n is called OPTIMAL for the class $\mathcal{P}$ if $(*)$ holds, and if, for some $c\in(0,1]$, we have in addition that

$$\inf_{\bar{f}_n} \sup_{\mathbb{P}_{X,Y}\in\mathcal{P}} \mathbb{E}\{\mathcal{E}(\bar{f}_n)\} \geq c\,e_n \quad \text{———} (**)$$

The sequence $\{e_n\}$ is called the MINIMAX rate of convergence for the class $\mathcal{P}$.

inf → 'MINI'
sup → 'MAX'

II.3. No free lunch results / slow rates

When no reasonable class of distributions $\mathcal{P}$ is available, a natural question is to ask what happens when we consider $\mathcal{P}$ to be the class of all probability distributions on (X, Y): can we find a predictor f_n whose (expected) excess risk converges to zero at a given rate, for all distributions of (X, Y)? The answer to this question is unfortunately negative. Such results are known as 'No free lunch' theorems in the litterature.

NO FREE LUNCH THEOREM #1

Let
- $(X, Y) \in \mathcal{X} \times \{0,1\}$ (binary classification)
- $\ell(y,f) = \mathbb{1}(y \neq f)$ (0-1 loss)
- $\mathcal{A}$ = prediction algorithm returning a function $\mathcal{X} \to \{0,1\}$
- $n \leq |\mathcal{X}|/2$; n = sample size

Then, there exists a distribution $\mathbb{P}$ over $\mathcal{X} \times \{0,1\}$ s.t

(i) There exists a function $f : \mathcal{X} \to \{0,1\}$ with $R(f) = 0$ $(R^* = 0)$

(ii) With probability $\geq 1/7$ over the choice of $\underset{\mathbb{P}^n}{\overset{S}{\mathcal{L}_n}}$, we have that $R(\mathcal{A}(\mathcal{L}_n)) \geq 1/8$

Bound in probability

Even if there exists a perfect predictor achieving zero risk, there is a relatively high chance that you construct a predictor based on $\mathcal{L}_n$ whose risk is bounded away from zero.

For every predictor, there is a task on which it fails, even though that task can be perfectly learned by another predictor.

proof: Since $|\mathcal{X}| \geq 2n$, take $\mathcal{C} \subset \mathcal{X}$ of size $2n$.

There are $N := 2^{2n}$ possible functions $\mathcal{C} \to \{0,1\}$.

Denote them $f_1, \dots, f_N$.

For each f_i, $i=1,\dots,N$, let $\mathcal{P}_i$ = distribution on $\mathcal{C} \times \{0,1\}$ defined by

$$\mathcal{P}_i(\{\underset{\in \mathcal{C}\times\{0,1\}}{\underbrace{x,y}}\}) = \begin{cases} 1/|\mathcal{C}| & \text{if } y = f_i(x) \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathbb{E}_{\mathcal{P}_i}\{R(f_i)\} = \mathcal{P}_i(Y \neq f_i(X)) = 0$, $i=1,\dots,N$

$(X,Y) \in \mathcal{C} \times \{0,1\}$

We want to show that

$$\max_{1 \leq i \leq N} \mathbb{E}_{\mathcal{L}_n \sim \mathcal{P}_i^n}\{R(\mathcal{A}(\mathcal{L}_n))\} \geq \frac{1}{4}$$

This implies that for any prediction algorithm $\mathcal{A}'$ receiving n observations from $\mathcal{X} \times \{0,1\}$, there exists a function $f : \mathcal{X} \to \{0,1\}$ and a probability distribution $\mathbb{P}$ on $\mathcal{X} \times \{0,1\}$ such that $R(f) = 0$, and $\mathbb{E}_{\mathcal{L}_n \sim \mathbb{P}^n}\{R(\mathcal{A}'(\mathcal{L}_n))\} \geq 1/4$, which in turn implies that $R(\mathcal{A}'(\mathcal{L}_n)) \geq \frac{1}{8}$ w.p. $\geq \frac{1}{7}$

Indeed, for a random variable Z taking values in $[0,1]$, we have that $\forall z \in [0,1]$,

$$\mathbb{P}(Z \leq 1-z) = \mathbb{P}(1-Z \geq z) \leq \frac{1-\mathbb{E}Z}{z} \Rightarrow \mathbb{P}(Z > 1-z) \geq \frac{\mathbb{E}Z-(1-z)}{z}$$

Replacing z by $1-z$ yields

$$\mathbb{P}(Z > z) \geq \frac{\mathbb{E}Z - z}{1-z}$$; & plugging in $z = 1/8$ with $\mathbb{E}Z \geq \frac{1}{4}$

yields $\mathbb{P}(Z > 1/8) \geq 1/7$.

The set $\mathcal{C}$ is of size $(2n)$. There are $M := (2n)^n$ possible sequences of length n from $\mathcal{C}$ (with replacement). We denote these sequences $\mathcal{L}_n^{(1)}, \dots, \mathcal{L}_n^{(M)}$.

In addition, if $\mathcal{L}_n^{(j)} = \{x_1, \dots, x_n\}$, we denote by $\mathcal{L}_n^{(j),i} := \{(x_1, f_i(x_1)), \dots, (x_n, f_i(x_n))\}$

label using the function f_i.

Therefore, $\mathbb{E}_{\mathcal{L}_n \sim P_i^n}\{R(A(\mathcal{L}_n))\} = \frac{1}{M} \sum_{m=1}^{M} R(A(\mathcal{L}_n^{(m),i}))$

Under P_i, the possible learning samples are $\mathcal{L}_n^{(1),i}, \dots, \mathcal{L}_n^{(M),i}$, and each have equal probability of being sampled.

$$\Rightarrow \max_{1 \le i \le N} \frac{1}{M} \sum_{m=1}^{M} R(A(\mathcal{L}_n^{(m),i})) \geqslant \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{m=1}^{M} R(A(\mathcal{L}_n^{(m),i}))$$

"max" is larger than "average"

$$= \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{i=1}^{N} R(A(\mathcal{L}_n^{(m),i}))$$

"min" is smaller than "average"

$$\geqslant \min_{1 \le m \le M} \frac{1}{N} \sum_{i=1}^{N} R(A(\mathcal{L}_n^{(m),i}))$$

Fix $m \in \{1, \dots, M\}$, consider $\mathcal{L}_n^{(m)} = \{x_1, \dots, x_n\}$, and let $v_1, \dots, v_p$ be the observations in $\mathcal{C}$ that do not appear in $\mathcal{L}_n^{(m)}$. We have that $p \geqslant n$.

$\Rightarrow \forall$ function $h : \mathcal{C} \to \{0,1\}$, $\forall i \in \{1, \dots, N\}$,

$$\frac{1}{2n} \sum_{x \in \mathcal{C}} \mathbb{1}(h(x) \neq f_i(x)) \geqslant \frac{1}{2n} \sum_{r=1}^{p} \mathbb{1}(h(v_r) \neq f_i(v_r))$$

$$\geqslant \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}(h(v_r) \neq f_i(v_r)).$$

Hence,

$$\frac{1}{N} \sum_{i=1}^{N} R(A(\mathcal{L}_n^{(m),i})) \geqslant \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}(A(\mathcal{L}_n^{(m),i})(v_r) \neq f_i(v_r))$$

$$= \frac{1}{2p} \sum_{r=1}^{p} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(A(\mathcal{L}_n^{(m),i})(v_r) \neq f_i(v_r))}$$

Fix some $r \in \{1, \dots, p\}$ & partition functions $f_1, \dots, f_N$ into $N/2$ distinct pairs $(f_i, f_{i'})$, such that $\forall x \in \mathcal{C}$, $f_i(x) \neq f_{i'}(x) \Leftrightarrow x = v_r$.

they agree everywhere, except on one point, given by v_r

For such pairs we must have $\mathcal{L}_n^{(m),i} = \mathcal{L}_n^{(m),i'}$, and

$$\mathbb{1}(A(\mathcal{L}_n^{(m),i})(v_r) \neq f_i(v_r)) + \mathbb{1}(A(\mathcal{L}_n^{(m),i'})(v_r) \neq f_{i'}(v_r)) = 1$$

and thus $\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(A(\mathcal{L}_n^{(m),i})(v_r) \neq f_i(v_r)) = 1/2$,

from which we get that $\frac{1}{N} \sum_{i=1}^{n} R(A(\mathcal{L}_n^{(m),i})) \geqslant 1/4$,

&

$$\max_{1 \le i \le N} \mathbb{E}_{\mathcal{L}_n \sim P_i}\{R(A(\mathcal{L}_n))\} \geqslant \min_{1 \le m \le M} \frac{1}{N} \sum_{i=1}^{N} R(A(\mathcal{L}_n^{(m),i}))$$

$$\geqslant 1/4,$$

which concludes the proof. ∎

[REF]: Theorem 5.1 in Understanding Machine Learning, Shai Shalev-Shwartz & Shai Ben-David.

NO FREE LUNCH THEOREM #2

Let
- $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ & ℓ = square loss function
- $\{a_n\}$ be a sequence of positive number converging to zero, such that $1/64 \geq a_1 \geq a_2 \geq \dots$
- $\mathcal{A}$ = prediction algorithm that returns a function $\mathcal{X} \to \mathbb{R}$

Then for any sequence of regression estimates $\{\mathcal{A}(\mathcal{L}_n)\}$, there exists a distribution $\mathbb{P}$ on (X, Y), such that $\mathcal{L}_n \sim \mathbb{P}^n$, $X \sim \mathcal{U}(0,1)$, $Y = \mathbb{E}(Y|X)$, and

$$\mathbb{E}\{\mathcal{E}(\mathcal{A}(\mathcal{L}_n))\} = \mathbb{E}\int_0^1 \left(\mathcal{A}(\mathcal{L}_n)(x) - r(x)\right)^2 dx \geq a_n \quad , \forall n.$$

over the distribution of $\mathcal{L}_n \sim \mathbb{P}^n$ $\qquad r(x) := \mathbb{E}(Y|X=x)$

(No free lunch theorem for the regression problem)

proof: see Chapter 3 in A distribution-free theory of non-parametric regression by L. Györfi et al.

# Take Away Message: There are no guaranteed rates of convergence.

We need to impose restrictions on the distribution of (X, Y).

Ex: Smoothness conditions on $\mathbb{E}(Y|X)$;
- differentiability, etc...
- Lipschitz conditions ...

III. LOCAL AVERAGING.

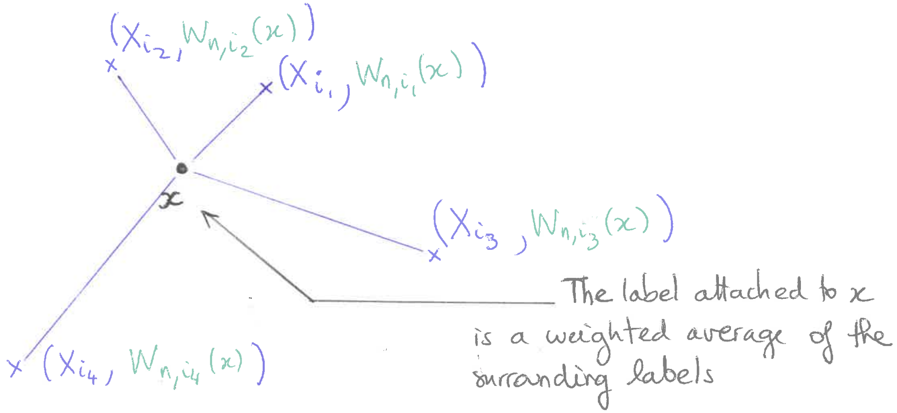
The regression function $r(x) = \mathbb{E}(Y|X=x)$ provides an optimal predictor both in a regression context (page 8) and in the context of binary classification (page 10). A natural approach is to construct an approximation of r based on local averaging: values of Y are averaged in a local neighborhood of x. More precisely, we construct

$$\forall x \in \mathcal{X} \qquad f_n(x) = \sum_{i=1}^{n} W_{n,i}(x)\, Y_i$$

based on $\mathcal{L}_n$, where the WEIGHT FUNCTIONS $W_{n,i}$ are usually taken such that $W_{n,i}(x) \geq 0$, and $\sum_{i=1}^{n} W_{n,i}(x) = 1$

$\forall x \in \mathcal{X}$

A natural condition to ensure local averaging is that more weight should be given to observations closer to x: for a given metric d, we should have (although not necessary)
$W_{n,i}(x) \geq W_{n,j}(x)$ iff $d(x, X_i) \leq d(x, X_j)$

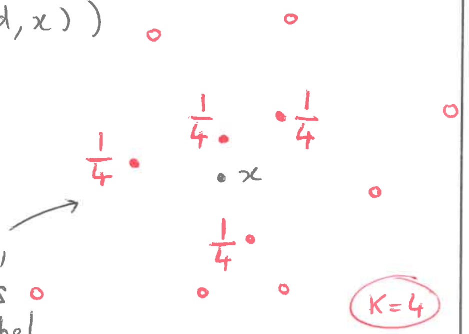

$(X_{i_2}, W_{n,i_2}(x))$
$(X_{i_1}, W_{n,i_1}(x))$
x
$(X_{i_3}, W_{n,i_3}(x))$
$(X_{i_4}, W_{n,i_4}(x))$

The label attached to x is a weighted average of the surrounding labels

We discuss three main choices for the weights:

(1) Nearest Neighbours. Fix an integer K in $\{1, \dots, n\}$, where n denotes the sample size. For a metric d on $\mathcal{X}$, and a point $x \in \mathcal{X}$, we define $\mathcal{N}(K, d, x)$ to be the random subset of $\{X_1, \dots, X_n\}$ composed of the K closest points to x. The associated weights are

$$W_{n,i}(x) = \frac{1}{K} \mathbb{1}(X_i \in \mathcal{N}(K, d, x))$$

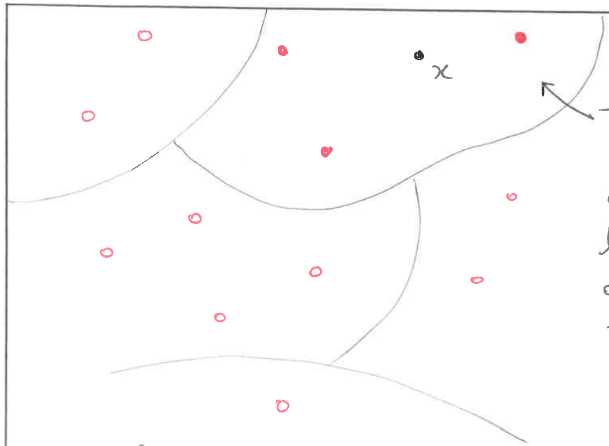


Points ● have weight 1/4, while remaining observations ○ have zero weight. The label attached to x is the average value of the four neighbours.

This method is known as K-nearest neighbours (**K-NN**) in the litterature.

(2) Partitioning. Let $\mathcal{P} = \{A_1, A_2, \dots\}$ be a partition of $\mathcal{X}$ ($A_i \cap A_j = \emptyset$, $i \neq j$, and $\bigcup A_j = \mathcal{X}$), and let $A(x)$ denotes the cell in $\mathcal{P}$ containing x. Then partitioning estimate (aka histogram) is defined as

$$f_n(x) = \frac{\sum_{i=1}^{n} Y_i \mathbb{1}(X_i \in A(x))}{\sum_{i=1}^{n} \mathbb{1}(X_i \in A(x))}, \text{ so that } W_{n,i}(x) = \frac{\mathbb{1}(X_i \in A(x))}{\sum_{j=1}^{n} \mathbb{1}(X_j \in A(x))}.$$

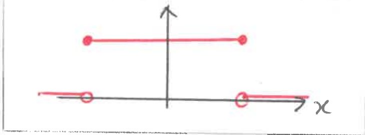
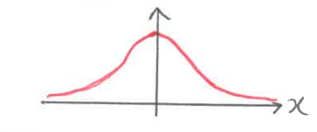


The label associated with x is the average value of the labels attached to observations belonging to the same cell as x.

Typically, when $\mathcal{X} = \mathbb{R}^d$ with $d > 1$, a cubic partition is used, so that the A_j are of volume h^d, where h denotes the side length of the cube A_j.

(3) Kernel estimates. Another popular local averaging procedure is based on kernels, which are functions $K : \mathbb{R} \to \mathbb{R}_+$, which are typically unimodal, and decrease to zero as $x \to \pm\infty$.

For example,

$K(x) = \mathbb{1}(\lvert x \rvert \leq 1)$	$K(x) = e^{-x^2}$
	

The kernel estimate is defined as

$$f_n(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}, \text{ so that } W_{n,i}(x) = \frac{K\left(\frac{x - X_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x - X_j}{h}\right)}$$

↖ $h > 0$ is usually refered to as the BANDWITH.

With $K(x) = \mathbb{1}(\lvert x \rvert \leq 1)$, one estimates the regression function by averaging the Y_i s such that the distance between X_i and x is less than h.

IV - LEARNING WITH HYPOTHESIS CLASSES

We often restrict the search of the prediction function to a class $\mathcal{F}$ of candidates, which is a subset of the set of measurable functions $X \to \mathbb{R}$.

Examples: (i) Class of linear functions.

$$\mathcal{F} = \left\{ f : \mathbb{R}^d \to \mathbb{R} \mid f(x) = \beta_0 + \beta^t x, \ \begin{matrix} \beta_0 \in \mathbb{R} \\ \beta \in \mathbb{R}^d \end{matrix} \right\}$$

(ii) Class of polynomials.

$$\mathcal{F} = \left\{ f : \mathbb{R} \to \mathbb{R} \mid f(x) = \beta_0 + \beta_1 x + \dots + \beta_d x^d, \ x \in \mathbb{R}, \ \beta_i \in \mathbb{R} \ i = 0, -, d \right\}$$

(iii) Convex combination of basis functions:

$$\mathcal{F} = \left\{ \sum_{i=1}^{M} \beta_i f_i \mid \|\beta\|_1 \leqslant 1, \ f_i : X \to \mathbb{R} \right\}$$

(iv) Neural network.

$$\mathcal{F} = \left\{ \sum_{i=1}^{M} \beta_i \sigma(a_i + b_i^t x) + \beta_0 \mid \begin{matrix} a_i, \beta_i \in \mathbb{R} \\ b_i, x \in \mathbb{R}^d \end{matrix} \right\}$$

$$\sigma(x) = \frac{e^x}{1+e^x}$$

- The prediction algorithm $\mathcal{A}$ returns a function from the class $\mathcal{F}$: $\mathcal{A} : \bigcup_{i \geqslant 1} (X \times Y)^i \longrightarrow \mathcal{F}$. We denote this function $\mathcal{A}(\mathcal{L}_n) = f_n \in \mathcal{F}$. For a loss function ℓ, the risk of f_n is then $R(f_n) = \mathbb{E}\{ \ell(Y, f_n(X)) \mid \mathcal{L}_n \}$. Recall that R^* denotes the smallest achievable risk, $R^* = R(f^*)$, where $f^* \in \operatorname{argmin}_{\substack{\text{meas functions} \\ f : X \to \mathbb{R}}} R(f)$, and $\mathcal{E}(f_n) := R(f_n) - R^*$ denotes the excess risk.

IV.1. Estimation vs Approximation.

We have the following decomposition:

$$\mathcal{E}(f_n) = R(f_n) - R^*$$
$$= \left\{ R(f_n) - \inf_{f \in \mathcal{F}} R(f) \right\} + \left\{ \inf_{f \in \mathcal{F}} R(f) - R^* \right\}$$

And so for a fixed class $\mathcal{F}$, consistency defined page 15 will not hold in general.

ESTIMATION ERROR

(random)

measures the performance of f_n compared to the best possible predictor in $\mathcal{F}$.

→ The estimation error gets larger as the complexity of $\mathcal{F}$ increases, since selecting f_n in $\mathcal{F}$ from $\mathcal{L}_n$ is more challenging

↓

controlling the estimation error: → PAC learning

APPROXIMATION ERROR.

(deterministic)

non-negative; may be zero even if $\mathcal{F}$ is smaller than the class of measurable functions $X \to \mathbb{R}$.

→ The approximation error gets smaller as the complexity of $\mathcal{F}$ increases.

trade-off.

↓

controlling the approximation error: → method of sieves

IV.2. Method of sieves.

The basic idea is that as n increases, we can afford fitting more complex models, so that instead of keeping the class of candidates fixed, we may consider a class $\mathcal{F}_n$ whose complexity increases with n, & such that $\mathcal{F}_n \subset \mathcal{F}_{n+1} \subset \dots$. If in addition the sequence $\{\mathcal{F}_n\}$ is such that

$$\inf_{f \in \bigcup_n \mathcal{F}_n} R(f) = R^*,$$

then the approximation error vanishes as $n \to +\infty$. Note that in the context of regression with square loss, the optimal predictor is $\mathbb{E}(Y \mid X = x)$, so that one can take for instance the class of polynomials of increasing order for the sequence $\{\mathcal{H}_n\}$ (indeed, the polynomials are dense in the space of continuous functions, which is dense in the space $\mathcal{L}^2$).

IV.3. PAC learning.

The concept of PAC learnability addresses the problem of quantifying the size of the estimation error.

↳ In other words, for a given class of candidates $\mathcal{H}$, is it possible, for any distribution of (X, Y), to select in $\mathcal{H}$ a predictor whose risk is arbitrarily close to that of the best predictor in that class.

PAC learnability. A hypothesis class $\mathcal{H}$ of functions is called PAC learnable if there exists an at most polynomial function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a predictor $f_n \in \mathcal{H}$ such that, $\forall (\varepsilon, \delta) \in (0,1)^2$, all $n \geq n_{\mathcal{H}}(\varepsilon, \delta)$, $\forall$ distribution of (X, X), the inequality

$$R(f_n) \leq \inf_{f \in \mathcal{H}} R(f) + \varepsilon$$

holds with probability $\geq 1 - \delta$.

Probably Approximately Correct ⇒ PAC

PAC learnability is thus the property of a class of candidates $\mathcal{H}$. As we will see later in this course, the so-called VC dimension of $\mathcal{H}$ plays a key role in establishing whether a class $\mathcal{H}$ is PAC learnable or not.

V _ EMPIRICAL RISK MINIMIZATION

In addition to local averaging techniques, another common algorithmic procedure is known as Empirical Risk Minimization (ERM): the learning sample $\mathcal{L}_n$ is used to approximate the theoretical risk $R(f) = \mathbb{E}\ell(Y, f(X))$, and then use this approximation to pick the function in the class of candidates $\mathcal{H}$ which minimizes it.

Definition. The EMPIRICAL RISK is defined as

$$\hat{R}_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i)).$$

for a fixed function $f \in \mathcal{H}$, the Law of Large Numbers ensures that $\hat{R}_n(f) \xrightarrow{a.s.} R(f)$ as $n \to +\infty$

Empirical Risk Minimization is the process of choosing a predictor $f \in \mathcal{H}$ which minimizes the empirical risk:

$$f_n \in \operatorname*{argmin}_{f \in \mathcal{H}} \hat{R}_n(f)$$

The minimizer is not necessarily unique.

ERM is a general technique for solving the learning problem, but provides no guidance for constructing f_n explicitely.

In some cases, the expression of the empirical risk minimizer can be computed exactly (Ex: square loss, $\mathcal{F}$ = space of linear functions), but most of the time, we must do this approximately. The minimization task is made easier when the loss is convex, and when the class $\mathcal{F}$ is convex as well ⇒ standard algorithmic procedures apply.

× Example: ERM for binary classification under 0-1 loss:

$$\boxed{f_n \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(Y_i \neq f(X_i))}$$

$\{$functions $\mathcal{X} \to \{0,1\}\}$

Note that neither the loss, nor the class $\mathcal{F}$ is convex.

⇒ In practice, finding f_n is intractable, except in a few particular cases. For instance, considering the class of linear predictors $\mathcal{F} = \left\{ x \mapsto \text{sign}(\beta_0 + \beta^t x) \mid \begin{smallmatrix} \beta_0 \in \mathbb{R} \\ \beta \in \mathbb{R}^d \end{smallmatrix} \right\}$, and provided the learning sample $\mathcal{L}_n$ is linearly separable, the task is relatively easy, and leads to the perceptron algorithm. In the non-separable case, the task is more challenging. A common approach is to consider a convex relaxation of the 0-1 loss: the original ERM task can be rewritten

$$(*) \quad \boxed{f_n \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \varphi_{\mathbb{1}}(-Y_i f(X_i)), \text{ where } \varphi_{\mathbb{1}}(z) = \mathbb{1}(z > 0)},$$

and the response variable Y is assumed to take values in $\{-1,1\}$. This expression follows from the fact that $Y_i \neq f(X_i) \Leftrightarrow Y_i f(X_i) < 0$

To make the optimization task $(*)$ a convex optimization problem, we need two ingredients:

↘ A convex function $\varphi_{\mathbb{1}}$

↘ A convex class $\mathcal{F}$

Introduce a function $\varphi : \mathbb{R} \to \mathbb{R}_+$ such that $\varphi(0) = 1$ and $\varphi(z) \geq \varphi_{\mathbb{1}}(z)$ $\forall z$. Such a function is called a CONVEX SURROGATE

The collection of functions taking values in $\{-1, 1\}$ is a non-convex set as soon as it contains two elements ⇒ instead, consider the class of functions taking values in the interval $[-1, 1]$ (these functions are referred to as SOFT CLASSIFIERS), and then perform binary classification according to the sign of the soft classifier evaluated at a particular x (the associated HARD CLASSIFIER)

ERM consists in solving the convex optimization problem

$$(**) \quad \boxed{f_n \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \varphi(-Y_i f(X_i))}$$

class of soft classifiers

a convex surrogate

taking values in $\{-1,1\}$.

A natural question: is this a meaningful thing to do? Do we have theoretical guarantees on the performance of the empirical risk minimizer of $(**)$? The answer to this question is positive. To show this, we introduce the φ-risk of $f \in \mathcal{F}$ = class of soft classifers: $\boxed{R_\varphi(f) := \mathbb{E}\, \varphi(-Y f(X))}$, and we write $\boxed{f_\varphi^* = \underset{f}{\operatorname{argmin}}\ R_\varphi(f)}$ & $\boxed{R_\varphi^* = R_\varphi(f_\varphi^*)}$

Then it is possible to show that

(1) $\boxed{P(Y=1 \mid X=x) \geqslant 1/2 \iff \operatorname{sign} f_\varphi^*(x) \geqslant 0}$

And therefore, considering ERM of the surrogate loss over the space of soft classifiers is a meaningful thing to do.

Moreover, it is possible to establish under general assumptions the existence of a constant $C<\infty$, and $\gamma \in [0,1]$, such that

Bayes Risk

(2) $$\boxed{\underbrace{R_{0\text{-}1}(\operatorname{sign} f) - R^*}_{\text{Excess risk of the associated hard classifier.}} \leqslant c\,\Big(\underbrace{R_\varphi(f) - R_\varphi^*}_{\text{Excess risk of the soft classifier}}\Big)^\gamma}$$

If the excess risk of a soft classifier is small, then the 0-1 excess risk of the associated hard classifier is also small.

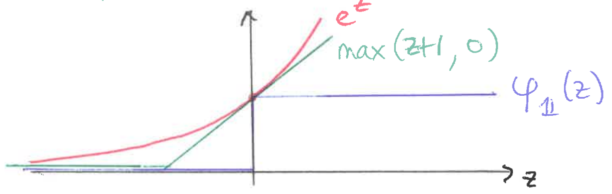
→ For proofs of (1) and (2), see chapter SL: CONVEX RELAXATION

× Examples of convex surrogates include the following:

(i) $\varphi(z) = \log(1+e^z)$ → logistic regression

(ii) $\varphi(z) = \max(z+1, 0)$ "hinge loss" → SVM

(iii) $\varphi(z) = e^z$ "exponential loss" → AdaBoost



× Remark: In practice, the performance of the hard classifier $\operatorname{sign}(f_n)$ constructed from $\mathcal{L}_n$ is evaluated by means of a CONFUSION MATRIX defined as

$$\begin{array}{ccc} & 1 & 0 \\ 1 & \mathrm{TP} & \mathrm{FP} \\ 0 & \mathrm{FN} & \mathrm{TN} \end{array} \leftarrow \text{truth}\ ,$$

↑ your prediction

where
- TP = number of True Positives
- TN = number of True Negatives
- FP = number of False Positives
- FN = number of False Negatives

Note that $TP + TN + FP + FN = n$ = sample size.

Moreover, one usually consider:

- TPR = True Positive Rate $= \dfrac{TP}{TP+FN}$ aka Sensitivity
- TNR = True Negative Rate $= \dfrac{TN}{TN+FP}$ aka Specificity

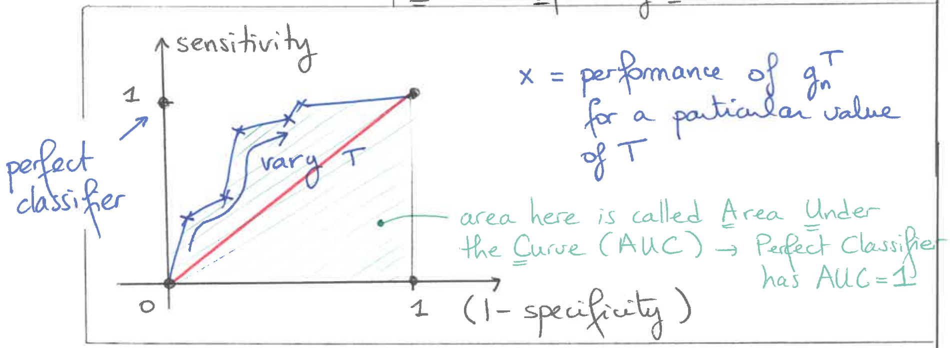
In theory, there is support for considering $\operatorname{sign}(f_n)$ as the final predictor. In practice however, one may consider more generally the following hard classifiers:

$$g_n^T(x) = \begin{cases} 1 & \text{if } f_n(x) \geqslant T \\ -1 & \text{otherwise} \end{cases},$$

for a range of values of the threshold T. The performance of each g_n^T is summarized in a confusion matrix, and it is customary to consider a graphical plot known as a ROC CURVE to compare the performance of g_n^T as T is varied.

↳ A ROC curve plots the sensitivity of g_n^T as a function of (1- specificity).

Receiver Operating Characteristic curve



x = performance of g_n^T for a particular value of T

area here is called Area Under the Curve (AUC) → Perfect Classifier has AUC = 1

On the ROC curve, we distinguish four particular points:

- (0, 0) = (hard) classifier always classify as 0
- (1, 1) = classifier always classify as 1
- (0, 1) = perfect classifier (since there are no false negatives & no false positives)
- (1, 0) = perfectly innacurate classifier (since there are no true positive & no true negative).

In addition, a random guess corresponds to a point on the diagonal. Since a good prediction corresponds to a high sensitivity and high specificity, points lying above the diagonal correspond to estimators performing better than a random guess, while point below the diagonal correspond to hard classifiers performing worse than a random guess.

x Remark: Probabilistic interpretation of AUC.

For a learning sample $\mathcal{L}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$, $Y_i \in \{0, 1\}$ the TPR and FPR are theoretically defined as

$$TPR = \mathbb{P}(f_n(X) \geqslant t \mid Y=1, \mathcal{L}_n) = \bar{F}_1(t) = \int_t^1 f_1(u)\, du$$

$$TNR = \mathbb{P}(f_n(X) \geqslant t \mid Y=0, \mathcal{L}_n) = \bar{F}_0(t) = \int_t^1 f_0(u)\, du$$

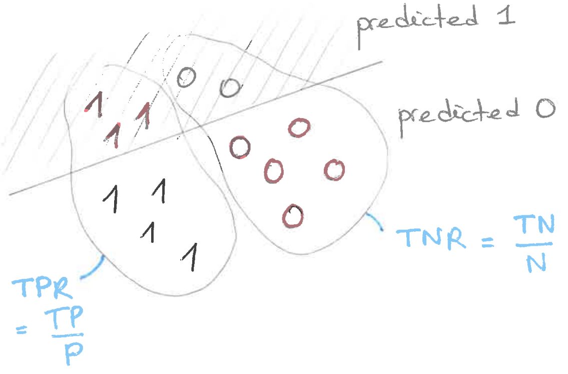
hard classifier

$$AUC = \int_1^0 \bar{F}_1(t)\, d\bar{F}_0(t) = \int_0^1 \int_t^1 f_1(u)\, du\, f_0(t)\, dt$$

$$= \mathbb{P}(f_n(X_1) \geqslant f_n(X_0) \mid Y_1 = 1, Y_0 = 0, \mathcal{L}_n),$$

where $(X_1, Y_1 = 1)$ and $(X_0, Y_0 = 0)$ are two randomly & independently chosen pairs.

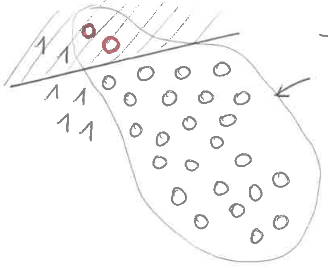
⇒ AUC corresponds to the probability that the score $f_n(X_1)$ is larger than $f_n(X_0)$, for a randomly generated sample $(X_1 \mid Y_1 = 1)$ from category 1, and for a randomly generated sample $(X_0 \mid Y_0 = 0)$ from category 0.



$TNR = \frac{TN}{N}$

$TPR = \frac{TP}{P}$

- **Imbalanced datasets.** In many interesting real-life applications, the positive class is under-represented (e.g. in medical application, default rates, ...)



The TNR is artificially inflated and the ROC curve is artificially shifted to the left, thus artificially increasing the AUC. But this is just a consequence of dealing with an imbalanced dataset, and has nothing to do with better predictions.

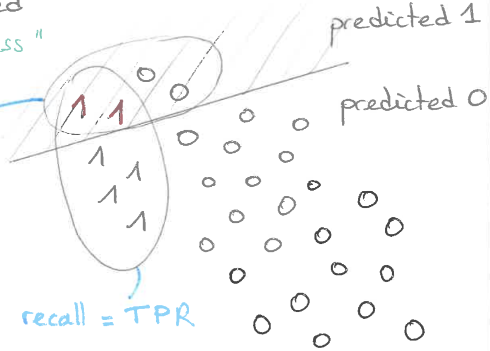
↳ Use Precision-Recall curves instead. These do not take into account the TNs.

$$\text{Precision} = \frac{TP}{TP+FP}$$ = proportion of retrieved items that are relevant

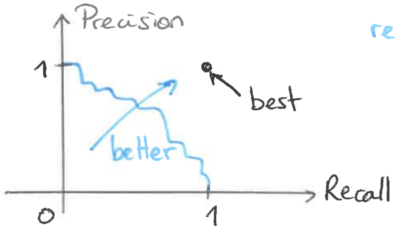
"Classifier exactness"

$$\text{Recall} = \frac{TP}{TP+FN}$$ = proportion of relevant items that are retrieved

"Classifier completeness"



⇒ We want a high precision and a high recall.



Since we want a classifier with a high precision and a high recall, we may define a new metric that measures the balance between them.

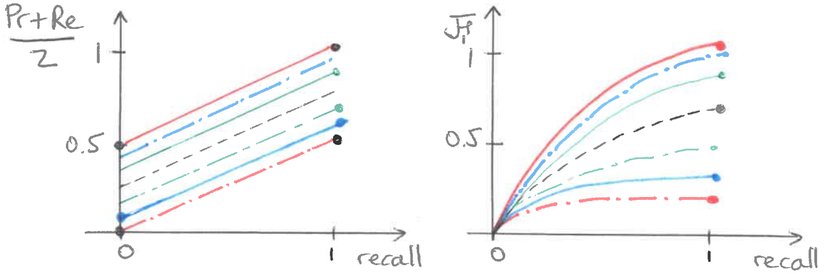
- Attempt #1: Arithmetic mean of Pr & Re : $\frac{1}{2}(Pr + Re)$.

Unfortunately, this is not a good measure, as it is easy for a classifier to obtain a good arithmetic mean of precision and recall.

- Attempt #2 : Harmonic mean $F_1 = \dfrac{1}{\frac{1}{2}\left(\frac{1}{Pr} + \frac{1}{Re}\right)} = \dfrac{2 \times Pr \times Re}{Pr + Re}$

Called the F_1 score

F_1 is better as the harmonic mean penalizes small values.



Arithmetic & Harmonic means of precision and recall as a function of precision (—·—, — —, —·—, - - - -, ——, —·—, ——) and recall. A good value of F_1 requires a high precision and recall; which is not true for the arithmetic mean.

- Remark: K-class classification, Precision and Recall

Inspired by the binary case,

truth

	1	0
1	TP	FP
0	FN	TN

→ precision

↓ recall

predicted

we may define precision and recall for each of the predicted classes :

← true class

	0	1	2	...	K
0	TP_0				
1		TP_1			
2			TP_2		
...					
K					TP_K

predicted class

precision : exactness
how much items predicted as class k are correct ?

confusion matrix .

recall : completeness
how much of items from class k have we retrieved ?

↳ Similarly, we may define the accuracy of a binary classifier as $\frac{TP+TN}{TP+FP+TN+FN}$, and the accuracy of a K-class classifier as $(TP_0 + \cdots + TP_K) / \#$ of observations .

An unfair metric for imbalanced datasets, as a high accuracy is not an indication of a good classifier (artificially inflated by the majority class) .

Better: Balanced Accuracy

$$\frac{1}{K}\left\{\frac{\#\text{recovered samples of class } 1}{\#\text{samples in class } 1} + \cdots + \frac{\#\text{recovered samples of class } K}{\#\text{samples in class } K}\right\}$$

= Average of recall obtained on each class .

Remark: Statistical Modeling: The Two Cultures
(Leo Breiman '01)

Leo Breiman's 2001 paper had a huge impact on the way statistical modeling is used to draw conclusions from the data. There are two main approaches :

#1 ↗ Assume a parametric model for the input-output relationship, or

#2 ↘ Consider a purely algorithmic procedure.

Depending on which approach you choose, you may reach different conclusions (and interpret the data in a different way). We explain the differences through the eye of linear regression, and logistic regression .

- Linear regression.

Approach #1: Assume a stochastic model for the relationship between X and Y of the form: $Y = \beta_0 + \beta_1 X + \varepsilon$, where
→ $X \in \mathbb{R}$, whose distribution is left unspecified
→ X and ε are independent
→ $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

Goal: estimation of the model parameters β_0, β_1 and σ^2.

[More generally, assume that the joint distribution of (X, Y) is $\mathbb{P}_\theta$, know up to a parameter $\theta \in \Theta$ = parameter space, and estimate θ using Maximum Likelihood for example]

Conditionally on $x_1, \ldots, x_n$, the MLE of β_0 and β_1 is obtained by maximizing the log-likelihood function $\ell(\beta_0, \beta_1, \sigma^2)$, given by:

$$\ell(\beta_0, \beta_1, \sigma^2) = \log\left\{ \frac{1}{(2\pi\sigma^2)^{n/2}} \prod_{i=1}^{n} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right)\right\}$$

↑ Since $(X_1, Y_1), \dots, (X_n, Y_n)$ are independent, and $Y_i \mid X_i = x_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$

$$= -\frac{n}{2}\log\sigma^2 - \frac{n}{2}\log 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

↗ Maximizing the log-likelihood with respect to β_0, β_1, is equivalent to minimizing $\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$ with respect to β_0 and β_1.

The MLE $\hat{\beta}_0$ and $\hat{\beta}_1$ of β_0 and β_1 satisfy:

$$\text{(I)}\quad (\hat{\beta}_0, \hat{\beta}_1) = \underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

↑ And we have theoretical guarantees for the quality of the estimates $(\hat{\beta}_0, \hat{\beta}_1)$, deduced from the general properties of the MLE: we have consistency (provided the model is correct, $\hat{\beta}_0, \hat{\beta}_1 \xrightarrow{a.s} \beta_0, \beta_1$), and asymptotic normality.

Approach #2: ERM for the regression problem, under a square loss, and over the class $\mathcal{F}$ of linear functions:

$$f_n = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$$

$$\mathcal{F} = \{x \mapsto \beta_0 + \beta_1 x\}$$

$$= \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

$$\Rightarrow f_n(x) = \tilde{\beta}_0 + \tilde{\beta}_1 x \quad \text{, where}$$

$$\text{(II)}\quad (\tilde{\beta}_0, \tilde{\beta}_1) = \underset{(\beta_0, \beta_1)}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

↗ Compare with minimization task (I): they are the same! However, the properties of the estimates $(\tilde{\beta}_0, \tilde{\beta}_1)$ are rather different: meaningless to talk about consistency and asymptotic normality of $(\tilde{\beta}_0, \tilde{\beta}_1)$ since we do not assume a stochastic model for (X, Y): no assumption on $P_{X,Y}$ here. In particular, there are no guarantees that the optimal predictor $E(Y \mid X = x)$ is linear in x.

Meaningful questions in this context: what is the estimation error $\{R(f_n) - \inf_{f \in \mathcal{F}} R(f)\}$ and the approximation error $\{\inf_{f \in \mathcal{F}} R(f) - R^*\}$?

x In general, Approach #1 useful when the task is more explanatory: you have compelling evidence that a linear relationship holds between the input and the output, and you are interested in interpreting the coefficients.

Approach #2 useful in a predictive task. Emphasis is on predicting the label of a new x (since by construction, we are minimizing a loss)

↑ In this example, the two approaches yield the same optimization problem. However, this is not true in general, and in particular, this is not true for the logistic regression problem.

- Logistic regression

Approach # 1 : The task is binary classification of the response variable $Y \in \{0, 1\}$ based on the predictor $X \in \mathbb{R}$. We assume that the relationship between X and Y is captured by the stochastic model $Y = \mathbb{1}(\beta_0 + \beta_1 X + \varepsilon > 0)$, where

- $X \in \mathbb{R}$, whose distribution is left unspecified
- X and ε are independent
- $\varepsilon \sim$ sigmoid distribution: $\mathbb{P}(\varepsilon \leq u) = (1 + e^{-u})^{-1} =: \sigma(u)$

Compare this model with the linear regression model page 38.

Goal: estimation of the model parameters β_0, β_1

Tool : maximum likelihood estimation (conditionally on $x_1, \dots, x_n$).

$$\ell(\beta_0, \beta_1) = \log\left\{\prod_{i=1}^{n} \mathbb{P}(Y_i = y_i \mid X = x_i \,;\, \beta_0, \beta_1)\right\}$$

$$= \sum_{i=1}^{n} \log\left\{ \sigma_i^{y_i} (1 - \sigma_i)^{1-y_i} \right\},$$

where $\sigma_i := \sigma(\beta_0 + \beta^t x_i)$
$= \mathbb{P}(Y_i = 1 \mid X_i = x_i)$

Maximize this quantity with respect to β_0, β_1

Note that $\sigma(u) = 1 - \sigma(-u)$

Moreover, we recover the usual expression for logistic regression, namely

$$\log\left\{\frac{\mathbb{P}(Y=1 \mid X=x)}{\mathbb{P}(Y=0 \mid X=x)}\right\} = \beta_0 + \beta_1 x$$

Analytical expression of the MLE is not available, but one may use Newton's procedure to obtain estimates numerically. As before, provided the model is correct, consistency of the MLE follows.

Approach #2 : As a starting point, we make use of the expression of the log-likelihood previously derived, and we show that the maximization task can be reexpressed as an ERM problem, with an appropriate loss function and class $\mathcal{H}$.

Maximizing the log-likelihood $\ell(\beta_0, \beta_1)$ with respect to β_0, β_1 is equivalent to minimizing

$$\sum_{i=1}^{n} - \log\left\{\left(\frac{1}{1 + \exp[-(\beta_0 + \beta_1 x_i)]}\right)^{y_i} \left(\frac{\exp[-(\beta_0 + \beta_1 x_i)]}{1 + \exp[-(\beta_0 + \beta_1 x_i)]}\right)^{1-y_i}\right\}$$

$$\Updownarrow$$

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} \log\left\{\left(1 + e^{-(\beta_0 + \beta_1 x_i)}\right) e^{(1-y_i)(\beta_0 + \beta_1 x_i)}\right\}$$

Put $y_i' := 2y_i - 1 \in \{-1, 1\}$

↘ If $y_i = 0$, $y_i' = -1$ and

$$\log\{\dots\} = \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) = \log\left(1 + e^{-y_i'(\beta_0 + \beta_1 x_i)}\right).$$

↘ If $y_i = 1$, $y_i' = 1$ and

$$\log\{\dots\} = \log\left(1 + e^{-(\beta_0 + \beta_1 x_i)}\right) = \log\left(1 + e^{-y_i'(\beta_0 + \beta_1 x_i)}\right)$$

The optimization problem is equivalent to

$$\min_{\beta_0, \beta_1} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i'(\beta_0 + \beta_1 x_i)}\right)$$

introduce the convex surrogate $\varphi(z) = \log(1 + e^{z})$ (page 31)

introduce the class of linear functions $\mathcal{H}: \{x \mapsto \beta_0 + \beta_1 x\}$

We obtain :

$$\hat{f}_n = \underset{f\in\mathcal{F}}{\operatorname{argmin}} \ \frac{1}{n}\sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

$f =$ soft classifier

$y_i \in \{-1, 1\}$

- An ERM task ! (see page 30)
- A convex optimization problem !

Take this ERM task as a starting point : no need of a stochastic model for (X, Y) : The modern approach

VI_ BIAS-VARIANCE TRADE-OFF.

The goal of the prediction problem is to select/construct a function $f_n \in \mathcal{F}$ based on a learning sample $\mathcal{L}_n$, whose excess risk $\mathcal{E}(f_n) = R(f_n) - R^*$ remains small (with high probability). As mentioned previously (see page 26), there is a tradeoff between two competing terms : the estimation error, and the approximation error, and one needs to select a class of predictors $\mathcal{F}$ carefully.

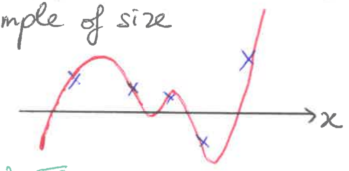
↗ A rich class $\mathcal{F}$ yields a small approximation error, but a large estimation error ← " small bias high variance "

↘ A small class $\mathcal{F}$ yields a large approximation error, but a small estimation error ← " high bias small variance "

What do you understand by a small / rich class $\mathcal{F}$?

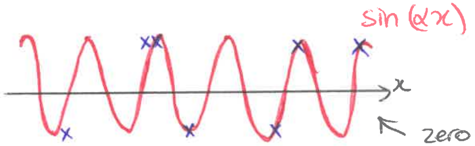
(Using terminology of classical estimation theory)

x Remark : . A polynomial of degree d, with $(d+1)$ parameters, can fit a sample of size d with zero error :



number of parameters ↔ complexity of $\mathcal{F}$

· However, in binary classification, it is possible to show that for any n, there exists a configuration of points $(x_1, \dots, x_n)$, such that irrespectively of the labels $(y_1, \dots, y_n)$, $y_i \in \{-1, 1\}$, there is an element of the class $\mathcal{F} := \{ x \mapsto \text{sign}(\sin(\alpha x)) \ , \alpha > 0 \}$ that achieves zero error on $\{(x_1, y_1), \dots, (x_n, y_n)\}$



↖ zero error, even if the class $\mathcal{F}$ contains only one parameter.

Conclusion: the number of parameters is not a good indication of the complexity of the class $\mathcal{F}$. As we will see later, a better notion of complexity is that of VC dimension.

→ We make the analogy between (approximation error & bias
estimation error & variance) more precise in the context of regression under a square loss.

Bias-Variance Tradeoff under a Square loss.

We have the following decomposition:

$$\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\} = \mathbb{E}_X\{\mathrm{Var}(Y|X)\} + \mathbb{E}_X\{\underbrace{(\mathbb{E}_{\mathcal{L}_n}(f_n(X)|X) - \mathbb{E}(Y|X))^2}_{\text{I}}\} + \underbrace{\mathbb{E}_{X,\mathcal{L}_n}\{(f_n(X) - \mathbb{E}_{\mathcal{L}_n}(f_n(X)|X))^2\}}_{\text{II}}$$

under a square loss

Remark: • $f_n(x)$ is a random function, even for a fixed x, as it is constructed from $\mathcal{L}_n$, and inherits the randomness of the learning sample

↳ Interpretation: $\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\}$ is decomposed into three terms

- $\mathbb{E}_X\{\mathrm{Var}(Y|X)\}$ = irreducible error (what is left unexplained)
- (I) = difference between the optimal predictor and the average prediction → SQUARE BIAS.

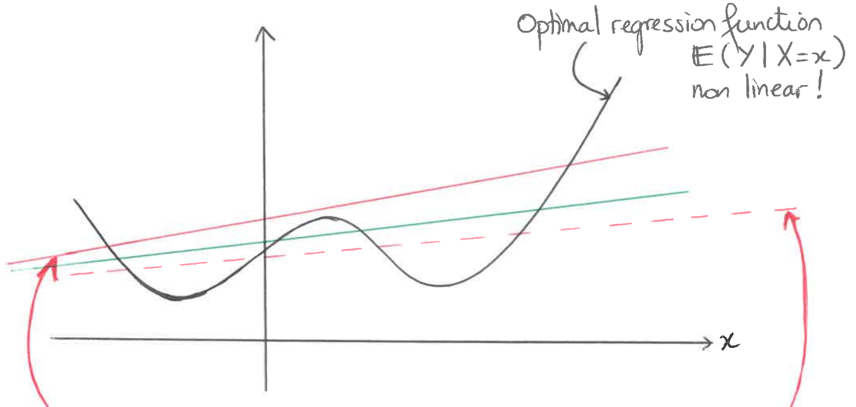
  The richer the class $\mathcal{F}$, the closer on average you will be to the optimal regression function $\mathbb{E}(Y|X)$ → small approximation error & small square bias.

- (II) = difference between the average prediction and a particular prediction for one learning sample → VARIANCE.

  The richer the class $\mathcal{F}$, the further away a particular predictor will be from the average prediction, since it will tune itself to the particular dataset → large estimation error & high variance.

[High bias & Small variance]

Consider the class of linear predictors.



Linear regression fit, based on the learning sample $\mathcal{L}_n^{(1)}$

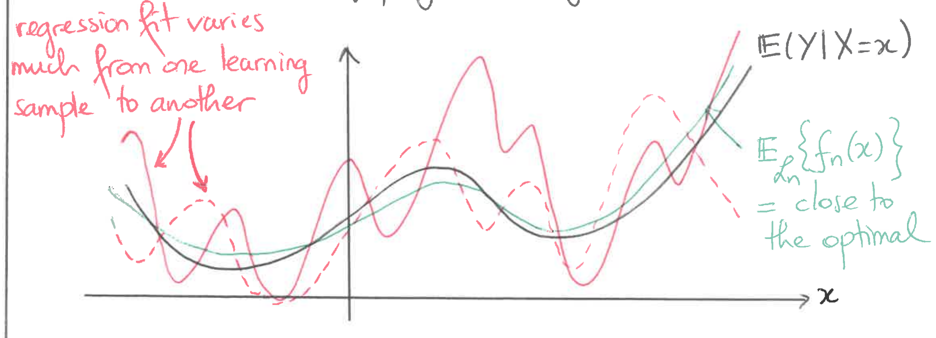
Linear regression fit, based on a second sample $\mathcal{L}_n^{(2)}$

In green: average prediction $\mathbb{E}_{\mathcal{L}_n}\{f_n(x)\}$

Conclusion: The linear fits do not change much from one sample to another (small variance), but remain on average far away from the optimal predictor (high bias)

[Small bias & High variance]

Consider the class of polynomials of order 100.



Proof of the bias-variance decomposition.

- First, we show that for a fixed predictor f (not constructed from the data), we have

$$R(f) = \mathbb{E}_X\{\mathrm{Var}(Y|X)\} + \mathbb{E}_X\{(f(X)-\mathbb{E}(Y|X))^2\}$$

Indeed,

$$R(f) = \mathbb{E}(Y-f(X))^2$$

$$= \mathbb{E}(Y-r(X)+r(X)-f(X))^2$$

with $r(X) = \mathbb{E}(Y|X)$

$$= \mathbb{E}(Y-r(X))^2 + \mathbb{E}(f(X)-r(X))^2$$

$$+\,2\,\underbrace{\mathbb{E}\{(Y-r(X))(f(X)-r(X))\}}$$

this term vanishes since:

$$= \mathbb{E}_X\Big\{(f(X)-r(X))\,\underbrace{\mathbb{E}[(Y-r(X))|X]}_{=0}\Big\}$$

- We apply the previous expression on f_n constructed from the learning sample, conditionning on $\mathcal{L}_n$ where appropriate:

$$R(f_n) = \mathbb{E}_X\{\mathrm{Var}(Y|X)\} + \mathbb{E}_X\big[(f_n(X)-r(X))^2\,|\,\mathcal{L}_n\big]$$

↳ Taking expectation with respect to $\mathcal{L}_n$ on both sides yields:

$$\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\} = \mathbb{E}_X\{\mathrm{Var}(Y|X)\} + \underbrace{\mathbb{E}_{X,\mathcal{L}_n}\big[(f_n(X)-r(X))^2\big]}$$

We turn our attention to this term, and introduce artificially the function $\mathbb{E}_{\mathcal{L}_n}\{f_n(X)\,|\,X\} =: f_m(X)$
"mean" predictor

$$\Rightarrow \mathbb{E}_{X,\mathcal{L}_n}\{(f_n(X)-r(X))^2\}$$

$$= \mathbb{E}_{X,\mathcal{L}_n}\{(f_n(X)-f_m(X)+f_m(X)-r(X))^2\}$$

$$= \mathbb{E}_{X,\mathcal{L}_n}\{(f_n(X)-f_m(X))^2\}$$

$$+\ \mathbb{E}_{X,\mathcal{L}_n}\{(f_m(X)-r(X))^2\}$$

$$+\ 2\times\underset{\underset{0}{\shortparallel}}{\text{Cross product}}.$$

, same reason as before.

Drop averaging with respect to $\mathcal{L}_n$ since both f_m and r are constant with respect to it.

Remark: Making use of the notation $f_m(X) := \mathbb{E}_{\mathcal{L}_n}\{f_n(X)\,|\,X\}$ introduced in the proof, and recalling that $\ell(y,f) = (y-f)^2$, the bias-variance decomposition may be rewritten:

$$\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\} = \underset{\text{irreducible error}}{\mathbb{E}_X\{\mathrm{Var}(Y|X)\}} + \underset{\text{bias term}}{\mathbb{E}_X\{\ell(f_m(X),r(X))\}}$$

$$+\ \underset{\text{variance term}}{\mathbb{E}_{X,\mathcal{L}_n}\{\ell(f_n(X),f_m(X))\}}$$

↑

Based on this observation, the bias-variance decomposition under a square loss can be generalized in the context of binary classification under a 0-1 loss. What we need is:

↘ expression of the optimal classifier; call it r^*

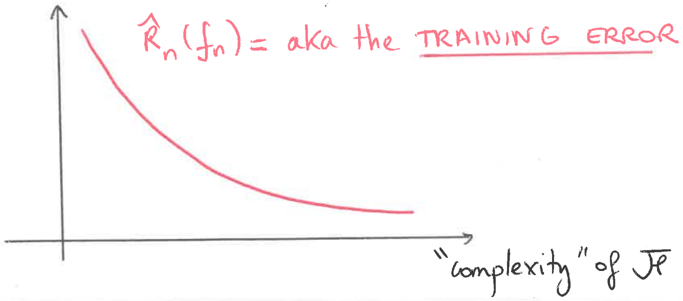
↘ definition of the main predictor $f_m(X)$,

and hope for a decomposition Noise + Bias + Variance, where the Bias term is related to $\ell_{0\text{-}1}(f_m, r^*)$, where $\ell_{0\text{-}1}$

is the 0-1 loss function, and the Variance term is expressed in terms of $\ell_{0\text{-}1}(f_n, f_m)$. It turns out that such a decomposition exists, and we refer the reader to [P. Domingos (2000). A Unified Bias-Variance Decomposition for Zero-One and Square Loss] for further details.

Remark: Estimating $\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\}$ in practice in the context of ERM.

- The empirical minimizer f_n is such that $\hat{R}_n(f_n) \leq \hat{R}_n(f)$, for any $f \in \mathcal{F}$, where $\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i))$ denotes the empirical risk (page 28).
- In practical situations, we typically observe one sample $\mathcal{L}_n$, so we can only hope for an estimate of $R(f_n)$, instead of $\mathbb{E}_{\mathcal{L}_n}\{R(f_n)\}$
- A natural candidate is $\hat{R}_n(f_n)$, since the SLLN ensures that for a fixed predictor $f \in \mathcal{F}$, $\hat{R}_n(f) \xrightarrow{a.s.} R(f)$ as $n \to +\infty$.
- However, as the complexity of the class $\mathcal{F}$ increases, you expect $\min_{f \in \mathcal{F}} \hat{R}_n(f)$ to decrease, as there are more and more elements in $\mathcal{F}$ that can reproduce the data.



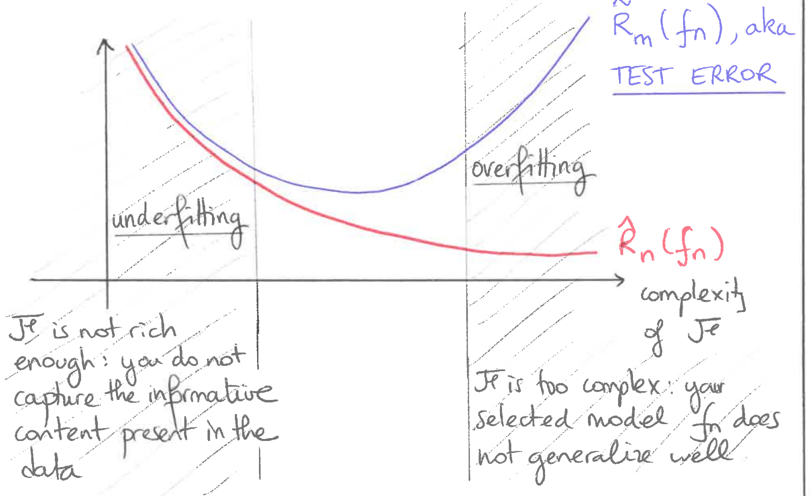
What goes wrong is that as the complexity of $\mathcal{F}$ increases, $\hat{R}_n(f_n)$ is not representative anymore of $R(f)$, since f_n is trained on $\mathcal{L}_n$; thus expecting good performance on $\mathcal{L}_n$, which does not guarantee a good performance on a new sample: we are OVERFITTING. The predictor f_n tunes itself to the noise present in the data.

→ The learning sample is used twice: for estimation and evaluation of the performance.

If you are given an additional independent TEST SAMPLE $\{(x_{n+1}, y_{n+1}), \dots, (x_{n+m}, y_{n+m})\}$, then

$$\tilde{R}_m(f_n) := \frac{1}{m}\sum_{i=1}^{m} \ell(y_{n+i}, f_n(x_{n+i}))$$

is a good estimator of $R(f_n)$, since $\tilde{R}_m(f_n) \to R(f_n)$, as $m \to +\infty$.



If a test sample is not available, then split $\mathcal{L}_n$ in two (fit the model on the first and evaluate its performance on the second), or use CROSS-VALIDATION.

↳ The optimism of the training error $\hat{R}_n(f_n)$ is due to the fact that we are using the sample twice: for estimation & evaluation. In other words, the response variables $Y_1, \dots, Y_n$ are used to train the model, and to test the performance accuracy. This optimism can be further understood if we compare $\hat{R}_n(f_n)$ with the theoretical performance of f_n when generating new samples $Y'_1, \dots, Y'_n$ associated with the <u>same</u> $X_1, \dots, X_n$. Formally, we introduce

$$\mathcal{L}_n = \{(x_1, Y_1), \dots, (x_n, Y_n)\} = \text{learning sample.}$$

The input variables $x_1, \dots, x_n$ are kept fixed ($\equiv$ conditioned on).

The only source of variability in $\mathcal{L}_n$ comes from the response variables $Y_1, \dots, Y_n$

In this notation, the training error is

$$\hat{R}_n(f_n) := \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_n(x_i)) \qquad (1)$$

↑ f_n constructed from $\mathcal{L}_n$.

We compare the training error with the "in-sample" error :

$$\bar{R}_n(f_n) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{Y'_i}\{\ell(Y'_i, f_n(x_i)) \mid \mathcal{L}_n\}, \qquad (2)$$

where $Y'_i \overset{d}{=} Y_i$ are iid RVs, and $\mathbb{E}_{Y'_i}\{\cdots\}$ denotes the expectation under the distribution of Y'_i. $\bar{R}_n(f_n)$ thus evaluates the true performance of f_n for the input points $x_1, \dots, x_n$. We define the optimism in the training error to be the difference $\bar{R}_n(f_n) - \hat{R}_n(f_n)$ between

these two quantities. The average optimism is the average value of $\bar{R}_n(f_n) - \hat{R}_n(f_n)$ computed over the distribution of the training sample (keeping $x_1, \dots, x_n$ fixed) :

$$\boxed{\text{Opt} := \mathbb{E}_{\mathcal{L}_n}\{\bar{R}_n(f_n) - \hat{R}_n(f_n)\}}.$$

↖ We derive another expression for Opt in the case of a square loss. To do so, we artificially introduce $\mathbb{E}Y_i$ / $\mathbb{E}Y'_i$ and $\mathbb{E}\{f_n(x_i)\}$ inside the squares in (1) and (2), and expand the terms :

- $$\hat{R}_n(f_n) = \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{Y_i - \mathbb{E}Y_i} + \underbrace{\mathbb{E}Y_i - \mathbb{E}f_n(x_i)} + \underbrace{\mathbb{E}(f_n(x_i)) - f_n(x_i)}\Big)^2$$

$$\begin{aligned}
= \frac{1}{n}\Big\{ &\sum_{i=1}^{n}(Y_i - \mathbb{E}Y_i)^2 \\
&+ \sum_{i=1}^{n}(\mathbb{E}Y_i - \mathbb{E}f_n(x_i))^2 \\
&+ \sum_{i=1}^{n}(\mathbb{E}f_n(x_i) - f_n(x_i))^2 \\
&+ \boxed{2\sum_{i=1}^{n}(Y_i - \mathbb{E}Y_i)(\mathbb{E}Y_i - \mathbb{E}f_n(x_i))} \\
&+ \boxed{2\sum_{i=1}^{n}(Y_i - \mathbb{E}Y_i)(\mathbb{E}f_n(x_i) - f_n(x_i))} \\
&+ \boxed{2\sum_{i=1}^{n}(\mathbb{E}Y_i - \mathbb{E}f_n(x_i))(\mathbb{E}f_n(x_i) - f_n(x_i))}\Big\}
\end{aligned}$$

Taking $\mathbb{E}_{\mathcal{L}_n}\{\cdots\}$ ↓

$$\begin{aligned}
\mathbb{E}_{\mathcal{L}_n}\{\hat{R}_n(f_n)\} = \frac{1}{n}\Big\{ &\sum_{i=1}^{n}\text{Var}\,Y_i + \sum_{i=1}^{n}(\mathbb{E}Y_i - \mathbb{E}f_n(x_i))^2 \\
&+ \sum_{i=1}^{n}\text{Var}\,f_n(x_i) \\
&- \boxed{2\sum_{i=1}^{n}\text{Cov}(Y_i, f_n(x_i))}\Big\}
\end{aligned}$$

(Terms $\boxed{\cdots}$ vanish)

$$\cdot\ \bar{R}_n(f_n) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{Y_i'}\Big\{\big(\underbrace{Y_i' - \mathbb{E}Y_i'}_{} + \underbrace{\mathbb{E}Y_i' - \mathbb{E}f_n(x_i)}_{} + \underbrace{\mathbb{E}f_n(x_i) - f_n(x_i)}_{}\big)^2 \,\Big|\, \mathcal{L}_n\Big\}$$

$$= \frac{1}{n}\Big\{\sum_{i=1}^{n} \mathbb{E}_{Y_i'}(Y_i' - \mathbb{E}Y_i')^2$$

$$+ \sum_{i=1}^{n} \cancel{\mathbb{E}_{Y_i'}}(\mathbb{E}Y_i' - \mathbb{E}f_n(x_i))^2$$

$$+ \sum_{i=1}^{n} \cancel{\mathbb{E}_{Y_i'}}\{(\mathbb{E}f_n(x_i) - f_n(x_i))^2 \mid \mathcal{L}_n\}$$

$$+ 2\sum_{i=1}^{n} \mathbb{E}_{Y_i'}(Y_i' - \mathbb{E}Y_i')(\mathbb{E}Y_i' - \mathbb{E}f_n(x_i))$$

$$+ 2\sum_{i=1}^{n} \mathbb{E}_{Y_i'}\{(Y_i' - \mathbb{E}Y_i')(\mathbb{E}f_n(x_i) - f_n(x_i) \mid \mathcal{L}_n\}$$

$$+ 2\sum_{i=1}^{n} \mathbb{E}_{Y_i'}\{(\mathbb{E}Y_i' - \mathbb{E}f_n(x_i))(\mathbb{E}f_n(x_i) - f_n(x_i)) \mid \mathcal{L}_n\}\Big\}$$

Taking $\mathbb{E}_{\mathcal{L}_n}\{\dots\}$ ↓

$$\mathbb{E}_{\mathcal{L}_n}\{\bar{R}_n(f_n)\} = \frac{1}{n}\Big\{\sum_{i=1}^{n} \mathrm{Var}\, Y_i + \sum_{i=1}^{n}(\mathbb{E}Y_i - \mathbb{E}f_n(x_i))^2$$

$$+ \sum_{i=1}^{n} \mathrm{Var}\, f_n(x_i) + 0\Big\}$$

(all cross product terms vanish)

$\Rightarrow$ We obtain

$$\boxed{\mathrm{Opt} = \frac{2}{n}\sum_{i=1}^{n} \mathrm{Cov}(Y_i, f_n(x_i))}\,.$$

The harder we fit the data, the larger $f_n(x_i)$ correlates with Y_i, thus increasing our optimism.

$\Rightarrow$ You may correct the training error by adding your optimism; and consider the adjusted training error:

$\hat{R}_n(f_n) + \frac{2}{n}\sum_{i=1}^{n} \mathrm{Cov}(Y_i, f_n(x_i))$. We discuss this further in the chapter SL: MODEL SELECTION

Apart from cross-validation, or adjusting the training error, there are several ways to avoid overfitting. These include:

(i) Restriction on the size / complexity of $\mathcal{F}$.

Consequence: it places an upper bound on the estimation error, but places a lower bound on the approximation error.

(ii) Modify the empirical risk to include a penalty / cost associated with more complex models.

$$\boxed{f_n \in \operatorname*{argmin}_{f\in\mathcal{F}} \{\underset{\substack{\uparrow\\ \text{goodness of fit}\\ \text{term}}}{\hat{R}_n(f)} + \underset{\substack{\uparrow\\ \text{penalty term}\\ \text{(depends on } f\text{)}}}{C(f)}\}}$$

Ex: in linear regression, you may want to limit the size of the coefficients, in order to decrease the overall variance of your prediction: for $f(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d$, take $C(f) = \sum_{i=1}^{d} \beta_i^2$ (ridge regression)

or $C(f) = \sum_{i=1}^{d} |\beta_i|$ (lasso)

or something else ...

(iii) If the class $\mathcal{F}$ is itself a (at most countable) union of classes $\mathcal{F}_1, \mathcal{F}_2, \dots,$ of increasing complexity; $\mathcal{F} = \bigcup_{k\geq 1}\mathcal{F}_k$, select $f_n \in \mathcal{F}$ such that

$$\boxed{f_n \in \operatorname*{argmin}_{f\in \bigcup \mathcal{F}_k} \{\hat{R}_n(f) + C(f)\}},$$

where the penalty $C(f)$ depends only on which $\mathcal{F}_k$ the predictor f belongs to (and not on f itself). Techniques for selecting $C(f)$ include STRUCTURAL RISK MINIMIZATION (SRM). SRM addresses the problem of model selection.