M2 CHPS - 2020/21

[Data & Apprentissage]

Introduction à la science des données et à l'apprentissage

Nicolas Vayatis

Séance 1 : Introduction du cours

Introduction

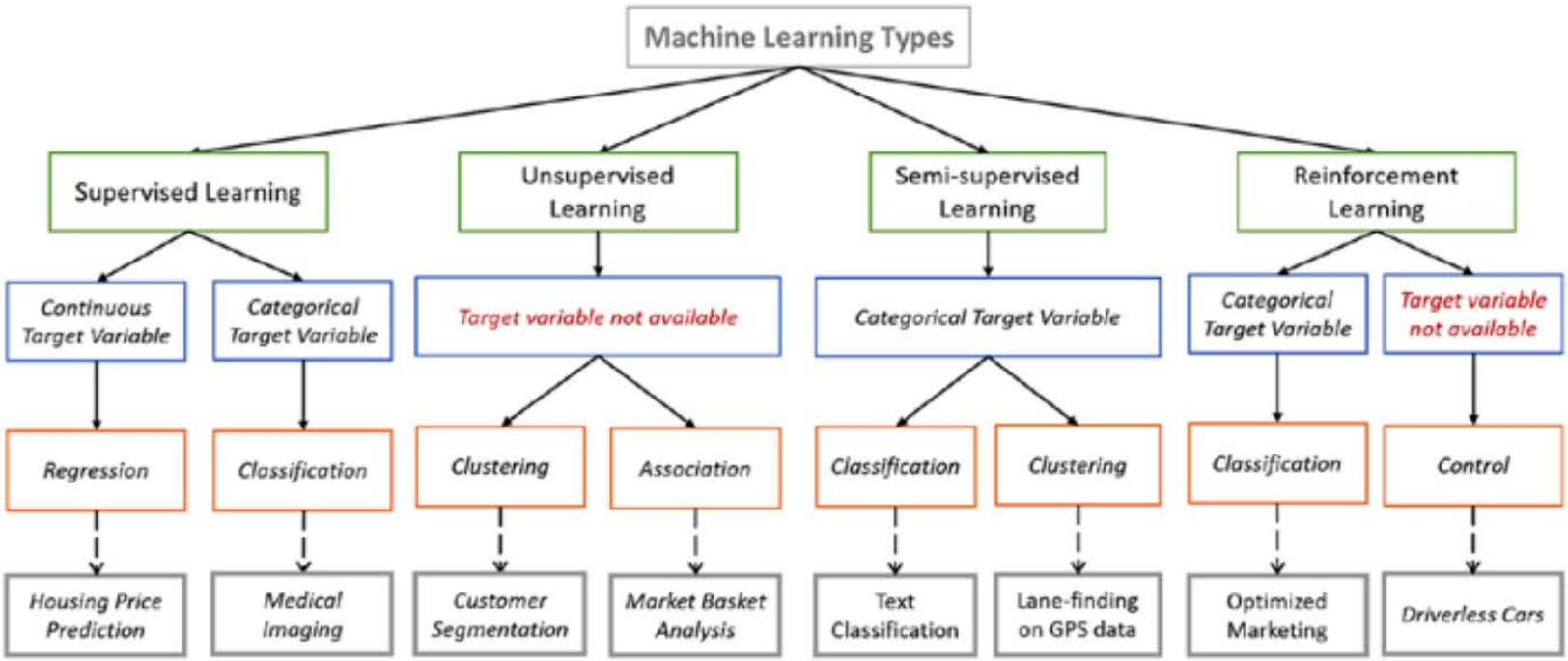
Data science : information, pipelines, decisions

- Why ? Information –> Data –> Predictions, decisions, knowledge...
- How ? Building pipelines from sensors to decisions (human in the loop ?)

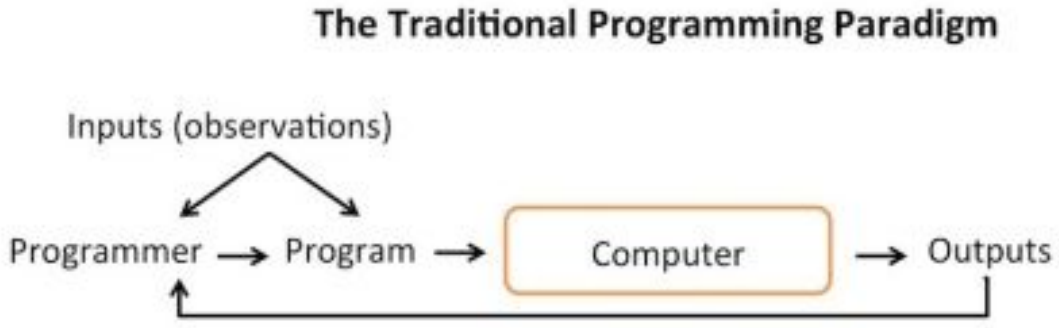
The three pillars of data science

1. Hardware : sensors, computing power, telecommunications, interfaces
2. Software environments : for design, testing and operations
3. Mathematical modeling and algorithms : high dimensional statistics, dimension reduction, **machine learning**, network science, time series, etc.

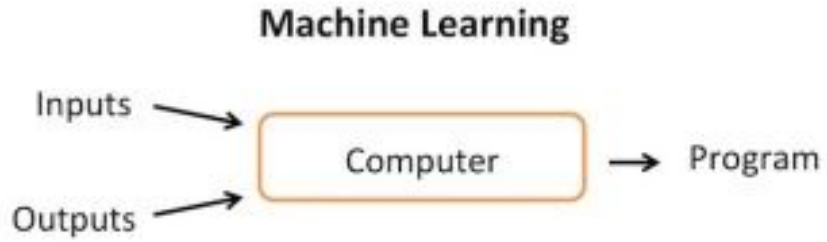
Types of Machine Learning problems



Symbolic AI vs. Machine Learning



Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
– Arthur Samuel (1959)

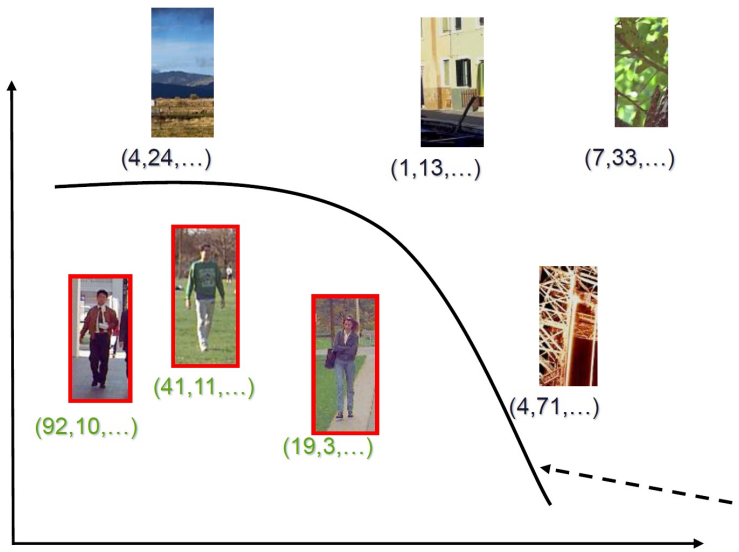


Sebastian Raschka, 2016

The goal of machine learning

Finding a function

- Example : Pedestrian detection from video cameras

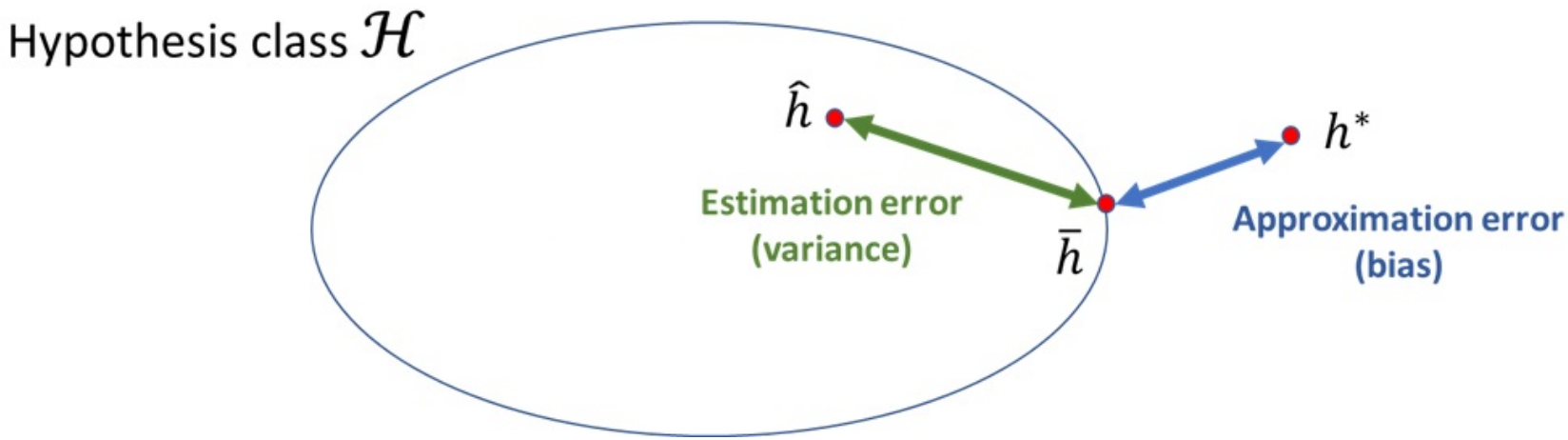


- What is the search space for such a function ?

The art of machine learning

Solving the "bias-variance" trade-off

- Distance between solution provided by a learning method and the optimal solution (function) : sum of *Approximation error* and *Estimation error*



- Learning a function amounts to :
  (a) chosing a search space (design process),
  (b) estimating the best function in this space (training process).

The three central paradigms of ML

1. Local methods : based on grouping and local voting (or averaging)
   - k-Nearest-Neighbors
   - Kernel rules
   - Decision trees
2. Global methods : based on functional optimization
   - Regularized regression (Ridge, LASSO...)
   - Support Vector Machines
   - Boosting
   - Feedforward neural networks
3. Ensemble methods : based on resampling and aggregation
   - Bagging
   - Boosting
   - Random forests

Shallow vs. Deep Learning

- Shallow learning : often relates to Tikhonov's regularization

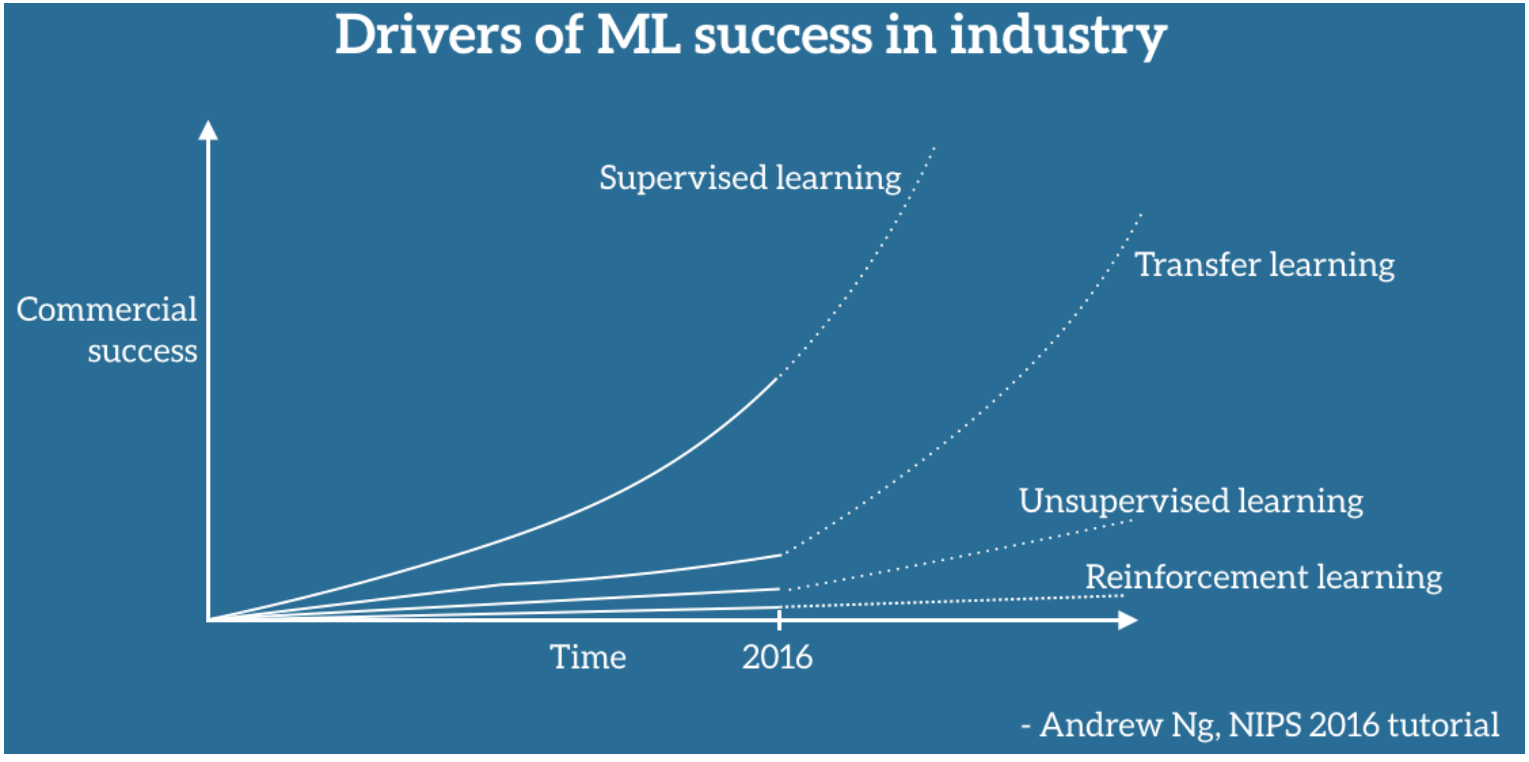
$$\min_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i) + \lambda_n \cdot \mathrm{pen}(h, n) \right)$$

  - The penalty controls the variance term (Occam's razzor)
  - It may also induce a desired structure of the function (e.g. sparsity).

- Deep Learning :

  - Universal approximators (zero bias)
  - No penalty term in the optimization but lots of tricks in the implementation which amount to *implicit regularization*

The various frameworks of Machine Learning



Machine Learning in the real world
A full pipeline for data processing

- data collection and indexing
- modeling,
- preprocessing (data quality, filtering and information compression)
- training,
- evaluation,
- monitoring,
- capitalization
- learning-to-learn

This course !

- Data points are **vectors** in $\mathbb{R}^d$
- Setup : supervised, some unsupervised
- Problems : classification, (scoring), regression, dimension reduction, clustering
- Focus on : a) problem setup, b) performance assessment, c) algorithms, d) principles and best-practice

Supervised Machine Learning

Learning and information

The bias-variance trade-off

Empirical Risk Minimization

Supervised Machine Learning

Learning and information

Learning like the twenty-question game

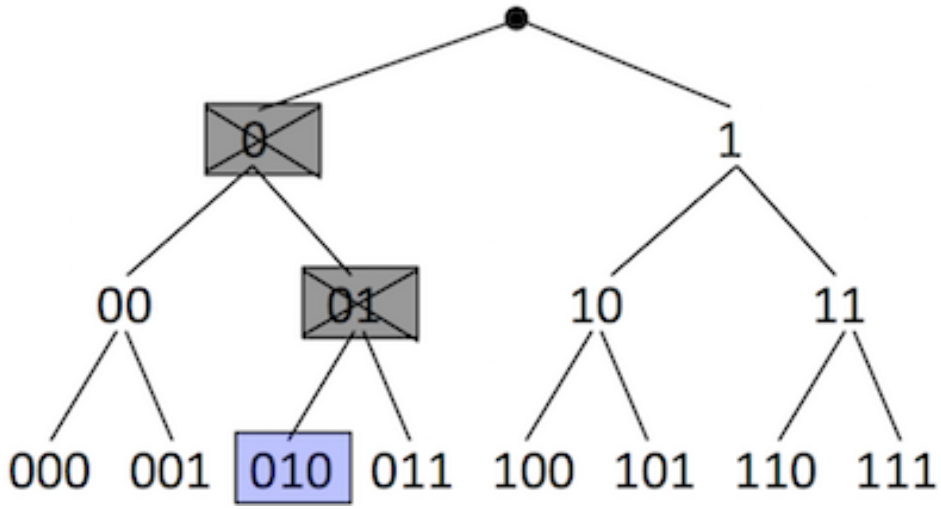
- Assume Nature has picked one function among K and we want to reveal this function
- Assume we have an oracle answering YES or NO when we ask a question about this function
- What is the optimal number n of questions to ask to find the unknown function ?

Brute force learning
Finite case

- ISSUE : How many questions with answers YES or NO one has to ask the oracle to find THE function among K functions ?
- STRATEGY : Proceed recursively by splitting the set of functions in two groups and asking whether THE function is the first group and removing the group which does not contain the function. This leads to the identification of the desired function with about $\log K$ questions.
- ANSWER : Number of questions $n = \dfrac{\log K}{\log 2}$
- NB : this quantity represents the number of bits of information characterizing the function in the set of K functions

Shannon's Information theory
The origin of the $\log K$

- Related to the entropy of a distribution P in information theory : $H(P) = -\sum_{k=1}^{K} P(k) \log P(k)$
- The entropy is the number of bits to encode a collection of K symbols (functions)



From questions to data
Zero error case

- Notations : Domain space $\mathcal{X}$ and label space $\mathcal{Y} = \{0, 1\}$
- ISSUE : How many examples $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ are required to find among a *finite* collection (size K) of indicator functions $f : \mathcal{X} \to \{0, 1\}$ the desired one ?
- SAME ANSWER : Number of examples $n = \dfrac{\log K}{\log 2}$
- STRATEGY : One has to find a vector x_i such that half of the functions take value 1 and the other half take value 0 and ask the oracle whether the desired function takes value 1 or 0 on this vector and discard those functions taking the opposite value. Apply this n times.

Probably approximately correct learning
Zero error case

- REMARK : it may be hard to find such an x_i which splits the collection of functions in two.
- NEW MODEL : Assume $X_1, \ldots, X_n$ is an IID sample
- QUESTION : How many examples (X_i, Y_i) are required to find among a finite collection of indicator functions $f : \mathcal{X} \to \{0, 1\}$ the one that with probability $1 - \delta$ is ε-close to the desired one ?
- ANSWER : Number of examples

$$n = \frac{\log K - \log \delta}{\varepsilon}$$

Probably approximately correct learning
General case

- ASSUME : among K functions, NONE of them commits zero error on the sample (X_i, Y_i).
- SAME ISSUE AS BEFORE
- ANSWER : Number of examples on average

$$n = \frac{\log K - \log \delta}{\varepsilon^2}$$

Same dependency on K, the only change is in the constant.

(Proof coming next)

Questions raised

- Proof arguments for PAC learnability (finite case)
- PAC : From finite to infinite collection of functions
- From "strategies" to "learning algorithms"
- What is lost through sampling ?

Supervised Machine Learning

The bias-variance decomposition in Machine Learning

General setup
Notations

- Goal of learning : an optimal decision function $h^* : \mathcal{X} \to \mathcal{Y}$
  $\mathcal{X}$: domain set, $\mathcal{Y}$: label set
- Input of learning :
  - **Training data :** a set of labeled data

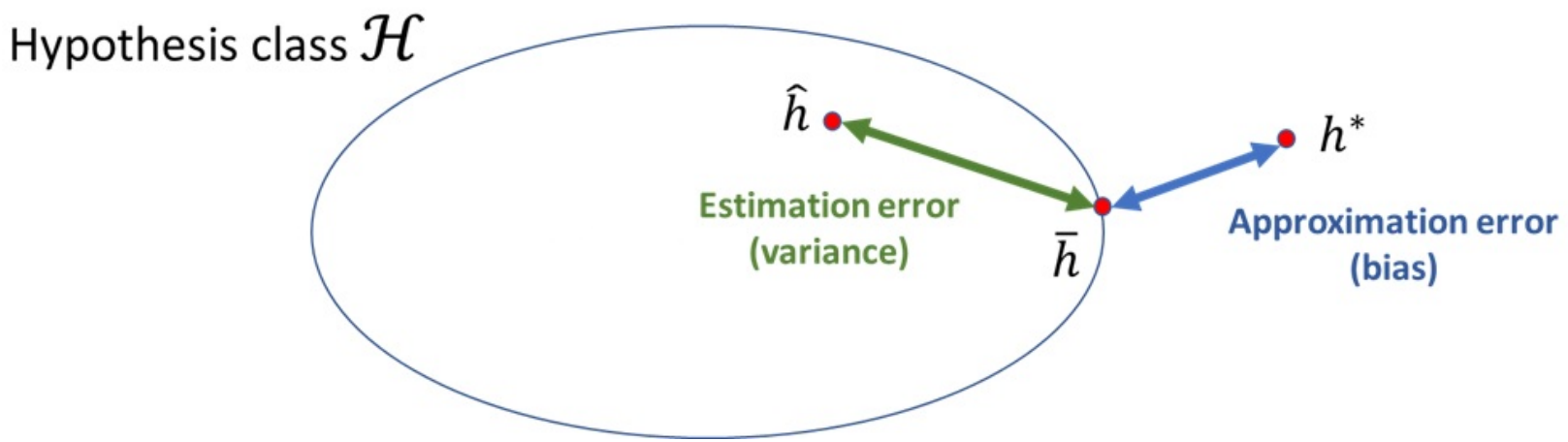
    $$D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

    of size n, where the (X, Y)'s are in $\mathcal{X} \times \mathcal{Y}$
  - **Hypothesis space :** a collection $\mathcal{H}$ of candidate decision functions $h : \mathcal{X} \to \mathcal{Y}$
- Output of learning : an empirical decision function $\widehat{h}$ in the hypothesis space $\mathcal{H}$ estimated from training data D_n
- Reference in $\mathcal{H}$: the best decision function $\bar{h}$ in the class (the more data, the closer $\widehat{h}$ to $\bar{h}$)

The key trade-off in Machine Learning

- Denote by $L(h)$ the error measure for any decision function h
- We have : $L(\bar{h}) = \inf_{\mathcal{H}} L$, and $L(h^*) = \inf L$
- Bias-Variance type decomposition of error for any output $\widehat{h}$:

$$L(\widehat{h}) - L(h^*) = \underbrace{L(\widehat{h}) - L(\bar{h})}_{\text{estimation (stochastic)}} + \underbrace{L(\bar{h}) - L(h^*)}_{\text{approximation (deterministic)}}$$



About approximation error

- Cybenko (1989) - Denseness result in the spirit of Stone-Weierstrass showing that any linear combination of compositions of sigmoid with linear functions is dense wrt the supremum norm in the space of continuous functions over the d-dimensional unit cube.
- Barron (1994) - Approximation error bound involves a parameter quantifying the smoothness of the target function.
- Status of this question in the regression setup :
  - For kernel machines : a full theory is available thanks to Smale (2003), Steinwart (2008).
  - For deep learning : recent work by Grohs, Perekrestenko, Elbrächter, and Bölcskei (2019) .
  - In the classification setup, tough problem, still open issue...

What all student know

The bias-variance trade-off in regression

The regression model

- Goal of learning : the *unknown* regression function $h^* : \mathbb{R}^d \to \mathbb{R}$
- Observations : IID random pairs $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$:

$$Y_i = h^*(X_i) + \varepsilon_i \ , \quad i = 1, \ldots, n$$

where ε_i is a random noise variable independent of X

- We shall use a vector notation as follows :

$$\mathbf{Y} = \mathbf{h}^* + \varepsilon$$

where the three terms are all in $\mathbb{R}^n$

Vector notations

- Elements in $\mathbb{R}^n$:

$$\mathbf{Y} = (Y_1, \ldots, Y_n)^T,$$
$$\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$$

- Image vectors : for any $h \in \mathcal{H}$, we use the bold characters as

$$\mathbf{h} = (h(X_1), \ldots, h(X_n))^T$$

- The image of data points through elements of $\mathcal{H}$ by

$$\mathcal{H}(X) = \{\mathbf{h} = (h(X_1), \ldots, h(X_n))^T \in \mathbb{R}^n : h \in \mathcal{H}\}$$

- Euclidean norm in $\mathbb{R}^n$:

$$\forall u = (u_1, \ldots, u_n)^T \in \mathbb{R}^n \ , \quad \|u\|^2 = \sum_{i=1}^{n} u_i^2$$

Least square estimator (LSE)

- Definition of the LSE :

$$\widehat{\mathbf{h}}_{\mathbf{n}} = \underset{\mathbf{h} \in \mathcal{H}(X)}{\arg\min} \frac{1}{n} \|\mathbf{Y} - \mathbf{h}\|^2$$

  where $\mathcal{H}(X) = \{\mathbf{h} = (h(X_1), \ldots, h(X_n))^T, : h \in \mathcal{H}\}$

- Error measure of the LSE :

$$L(\widehat{h}_n) = \frac{1}{n} \mathbb{E}\left(\|\mathbf{h}^* - \widehat{\mathbf{h}}_{\mathbf{n}}\|^2\right)$$

Gaussian linear model in $\mathbb{R}^d$
Two additional assumptions

- The hypothesis space $\mathcal{H}$ is the class of *linear* functions of rank d
- The noise vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$ is a *gaussian* random vector in $\mathbb{R}^n$ with distribution $\mathcal{N}_n(0, \sigma^2 I_n)$

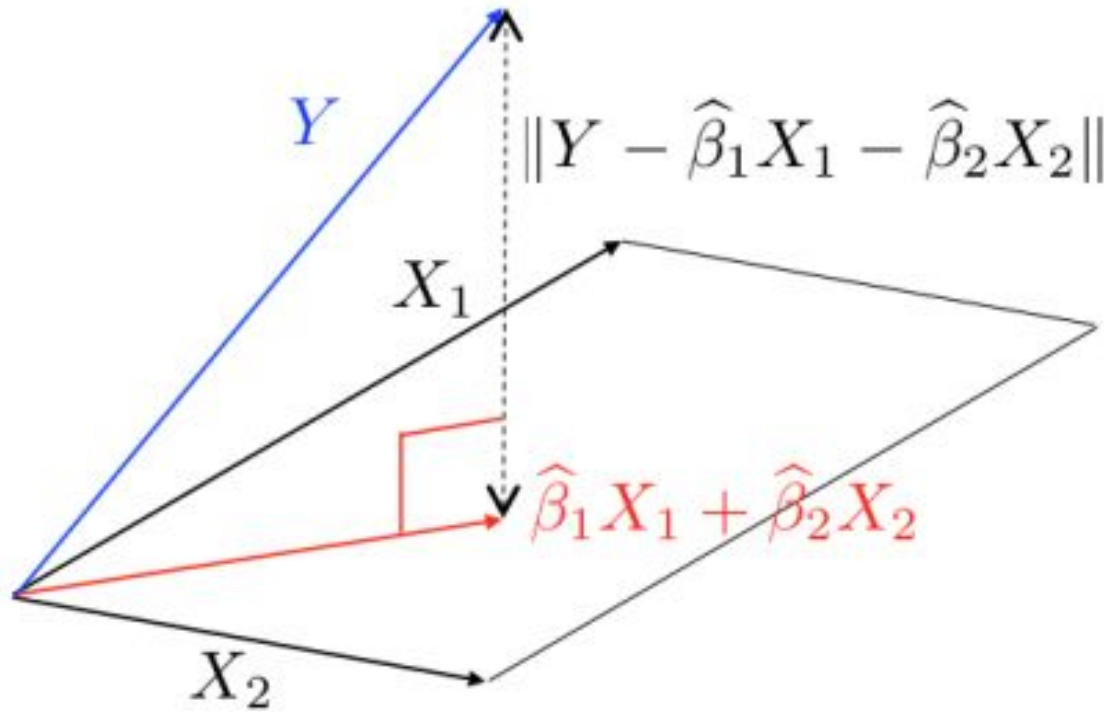
Linear models in $\mathbb{R}^d$
Examples

Notations : $x = (x^{(1)}, \ldots, x^{(d)})^T \in \mathbb{R}^d$

- Linear regression : $h(x) = \sum_{k=1}^{d} \beta_k x^{(k)}$
- Basis/frame expansion (Fourier, splines, wavelets, etc.)
- Additive models : $h(x) = \sum_{k=1}^{d} f_k(x^{(k)})$
- Piecewise constant regression (taking into account breakpoints)

LSE in linear regression

- Denote by $\mathbf{X}$ the data matrix $(n \times d)$
- Least square estimate : $\widehat{\mathbf{h}}_\mathbf{n} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \widehat{\Pi}\mathbf{Y}$



Bias-variance decomposition (1/2)
Derivation

- First note that : $\widehat{\mathbf{h_n}} = \widehat{\Pi}\mathbf{Y} = \widehat{\Pi}(\mathbf{h}^* + \varepsilon)$ and then

$$\mathbf{h}^* - \widehat{\mathbf{h_n}} = (I_n - \widehat{\Pi})\mathbf{h}^* - \widehat{\Pi}\varepsilon$$

- Note that $\widehat{\Pi} : \mathbb{R}^n \to \mathbb{R}^n$ the orthogonal projection onto $\mathcal{H}(X)$:

$$\widehat{\Pi} \circ \widehat{\Pi} = \widehat{\Pi}$$

- By orthogonality of the images of $I_n - \widehat{\Pi}$ and $\widehat{\Pi}$:

$$\begin{aligned} L(\widehat{h}_n) &= \frac{1}{n}\mathbb{E}\big(\|\mathbf{h}^* - \widehat{\mathbf{h_n}}\|^2\big) \\ &= \frac{1}{n}\mathbb{E}(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2 + \|\widehat{\Pi}\varepsilon\|^2) \end{aligned}$$

Bias-variance decomposition (2/2)
Result

- Using an additional technical result (next slide) :

$$
\begin{aligned}
L(\widehat{h}_n) &= \frac{1}{n}\mathbb{E}\big(\|\mathbf{h}^* - \widehat{\mathbf{h}}_\mathbf{n}\|^2\big) \\
&= \frac{1}{n}\mathbb{E}\big(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2 + \|\widehat{\Pi}\varepsilon\|^2\big) \\
&= \underbrace{\frac{1}{n}\mathbb{E}\big(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2\big)}_{\text{bias}} + \underbrace{\sigma^2\frac{d}{n}}_{\text{variance}}
\end{aligned}
$$

- Used in model selection $\rightarrow$ discussed in next class

Explanation of the d/n term

Property on the norm of projections of gaussian random vectors :

- Assume $\mathbf{Z}$ is a gaussian random vector $\mathcal{N}_n(0, I_n)$ in $\mathbb{R}^n$, $\mathcal{H}$ is a linear subspace of $\mathbb{R}^n$ and $\Pi : \mathbb{R}^n \to \mathbb{R}^n$ a linear projection onto $\mathcal{H}$
- Then : the random vector $\Pi_{\mathcal{H}}\mathbf{Z}$ has gaussian distribution $\mathcal{N}_n(0, \Pi)$ on $\mathbb{R}^n$ (linear transformation of a gaussian is a gaussian)
- Furthermore : $\|\Pi\mathbf{Z}\|^2$ follows a chi-square distribution with

$$\mathbb{E}(\|\Pi\mathbf{Z}\|^2) = \dim(\mathcal{H})$$

How Machine Learning takes over linear regression

1. What if non-additive noise ? Other tasks than regression ?
2. From linear to nonlinear models
3. What replaces the dimension d as a measure of complexity in nonlinear models ?
4. Is the d/n rate also typical for larger hypothesis classes ? What if d larger than n ?

Supervised Machine Learning

Empirical Risk Minimization (ERM)

The ERM principle
Definition

- Loss function : $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty]$
- Empirical risk of a decision rule h : this is a data-dependent functional

$$\widehat{L}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i)$$

- ERM = Empirical Risk Minimization

  Learning from training data amounts to solving the following optimization problem

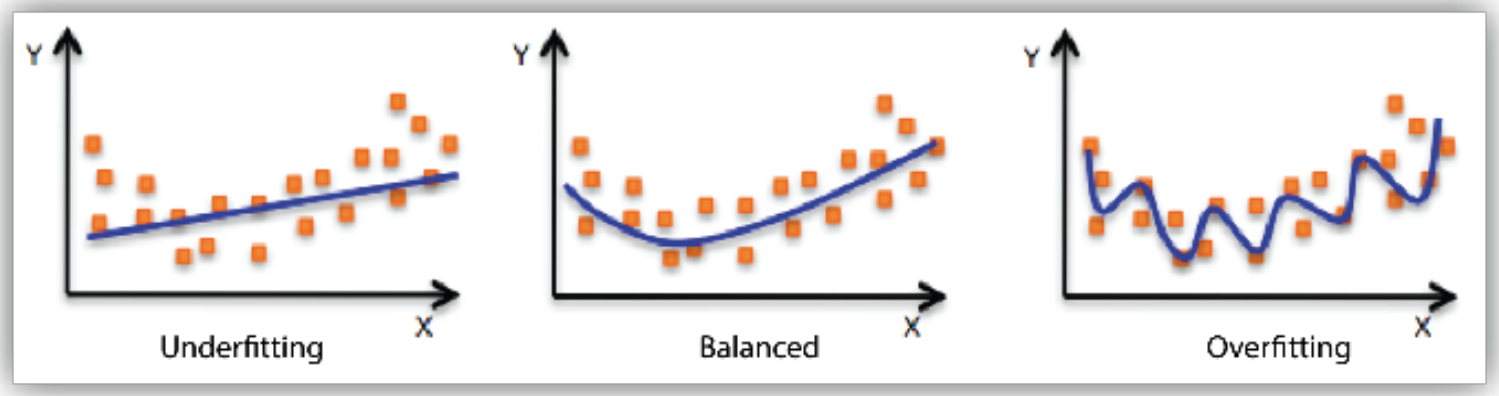
$$\widehat{h}_n = \arg\min_{h \in \mathcal{H}} \widehat{L}_n(h)$$

  where the minimization is restricted to the hypothesis space.

The ERM principle
Main questions

1. The question of consistency : convergence of $\widehat{h}_n$ wrt the sample size n ?
2. What is the cost incurred for using training data instead of the actual data ?
3. What is the nature of the trade-off to calibrate the complexity of the hypothesis space $\mathcal{H}$?

Overfitting vs. underfitting



Less is more :

- It turns out that considering all measurable functions leads to overfitting $\Rightarrow \mathcal{H}$ has to be a restricted class !

But greed is good :

- Algorithms which have the capacity to overfit means they have high representation power (arbitrary small approximation error)

The notion of *true* error

- Assumption :

  (X, Y) is a pair of random variables with joint distribution P

- True error of a decision rule h : this is a distribution-dependent functional

$$L(h) = \mathbb{E}(\ell(h(X), Y)) = \int \ell(h(x), y) dP(x, y)$$

Examples of tasks/problems

- Binary classification problem : Y takes 0-1 values

$$\ell(y, y') = \mathbb{I}\{y \neq y'\} \quad \text{and} \quad L(h) = \mathbb{P}\{h(X) \neq Y\}$$

- Regression : Y takes values in $\mathbb{R}$

$$\ell(y, y') = (y - y')^2 \quad \text{and} \quad L(h) = \mathbb{E}\big((Y - h(X))^2\big)$$

Optimal elements, consistency and bounds

- Bayes rule h^* and Bayes error L^*

$$h^* = \arg\min_h L(h) \quad \text{and} \quad L^* = L(h^*)$$

- (Strong) Consistency of an inference principle $\widehat{h}_n$

$$L(\widehat{h}_n) \to L^* \ , \quad \text{almost surely}$$

- The nonasymptotic bounds Eldorado :

$$L(\widehat{h}_n) - L^* \leq U(n, \mathcal{H}) \quad \text{whp}$$

Exercise

Find optimal elements h^* and L^* in these two cases :

- Binary classification problem : Y takes 0-1 values

$$\ell(y, y') = \mathbb{I}\{y \neq y'\} \quad \text{and} \quad L(h) = \mathbb{P}\{h(X) \neq Y\}$$

- Regression : Y takes values in $\mathbb{R}$

$$\ell(y, y') = (y - y')^2 \quad \text{and} \quad L(h) = \mathbb{E}\big((Y - h(X))^2\big)$$

We shall use the notations $\eta(x) = \mathbb{E}(Y \mid X = x)$ and use the fact that $\mathbb{E}(Y) = \mathbb{E}\big(\mathbb{E}(Y \mid X)\big)$

Estimation vs. approximation error
Extension of bias-variance decomposition

- Proof idea : Add and retrieve $\widehat{L}(\widehat{h}_n)$, $\widehat{L}(\overline{h})$, $L(\overline{h})$, then use the definition of ERM to upper bound the sum. Difference between L and $\widehat{L}$ appear twice.
- We have :

$$L(\widehat{h}_n) - L^* \leq \underbrace{2 \sup_{h \in \mathcal{H}} |L(h) - \widehat{L}_n(h)|}_{\text{estimation (stochastic)}} + \underbrace{L(\overline{h}) - L^*}_{\text{approximation (deterministic)}}$$

Finite hypothesis class
Generalization error bound for ERM

- Assume that the hypothesis class $\mathcal{H}$ of decision functions is finite and $h^* \notin \mathcal{H}$
- Then, we have, for any δ, with probability at least $1 - \delta$:

$$\forall h \in \mathcal{H} \ , \quad L(h) \leq \widehat{L}_n(h) + \sqrt{\frac{\log |\mathcal{H}| + \log\left(\frac{1}{\delta}\right)}{2n}}$$

- $\log |\mathcal{H}| = \log K \rightarrow$ Statement in the introduction, see !

Finite hypothesis class
Sketch of proof

- Hoeffding's inequality :
  - Consider $Z_1, \ldots, Z_n$ IID over $[0,1]$ and $\overline{Z}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$
  - We have, for any $t > 0$

$$\mathbb{P}\{\overline{Z}_n - \mathbb{E}(Z_1) > \varepsilon\} \leq \exp(-2n\varepsilon^2)$$

- Union bound : For any two measurable sets A, B, we have :

$$\mathbb{P}\{A \cup B\} \leq \mathbb{P}\{A\} + \mathbb{P}\{B\}$$

Next sessions

- Regularization and sparsity assumptions for linear models
- From linear to for nonlinear models
- Mainstream ML algorithms and their evaluation