M2 CHPS

[Data & Apprentissage]

Introduction à la science des données et à l'apprentissage

Nicolas Vayatis

Introduction du cours

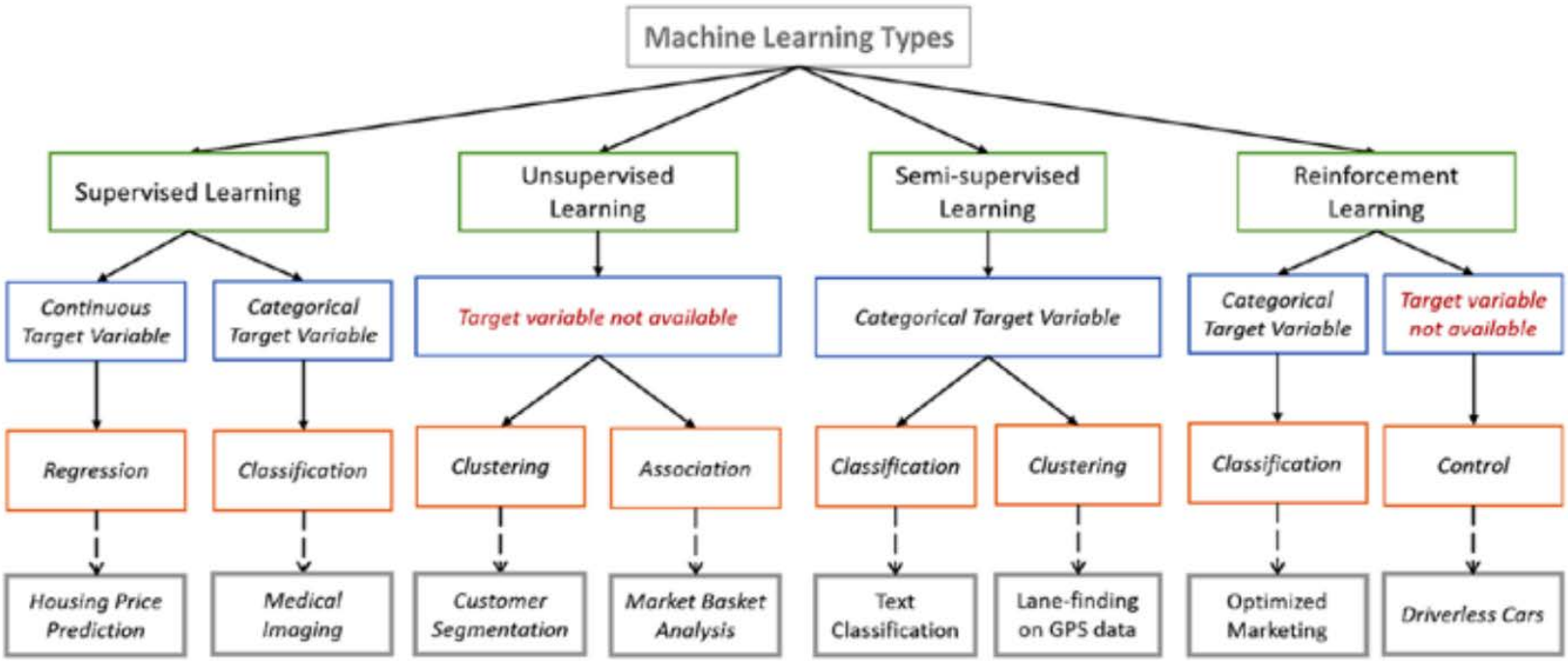
Data science : information, pipelines, decisions

- Why ? Information –> Data –> Predictions, decisions, knowledge...
- How ? Building pipelines from sensors to decisions
- For whom ? For humans ! with Humans in the loop (or not...)

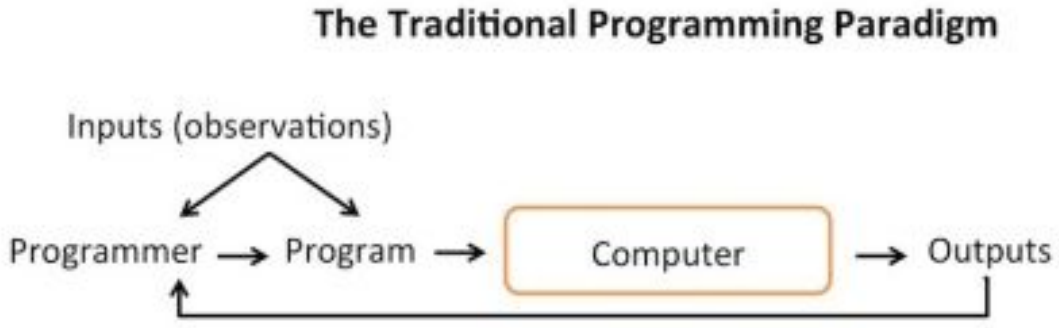
The three pillars of data science

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.

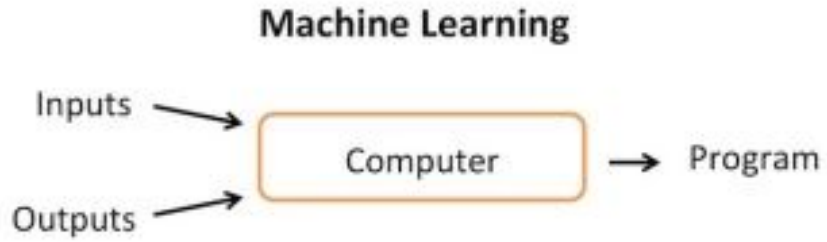
Types of Machine Learning problems



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)

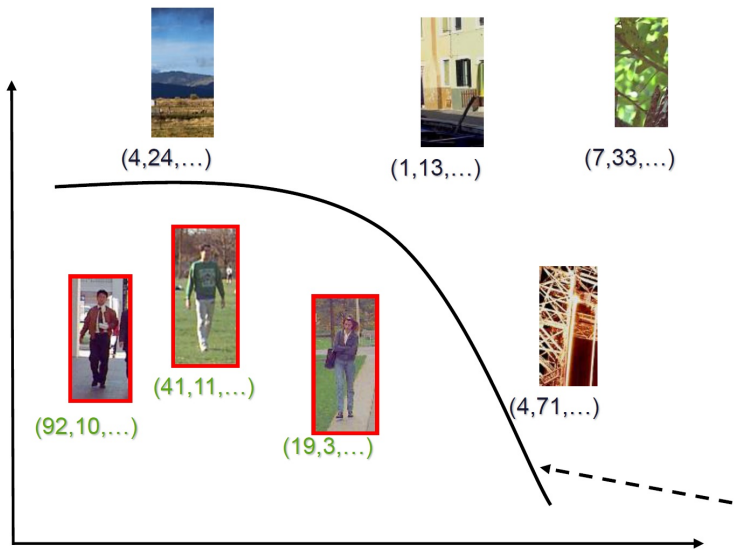


Sebastian Raschka, 2016

The goal of machine learning

Finding a function

- Example : Pedestrian detection from video cameras

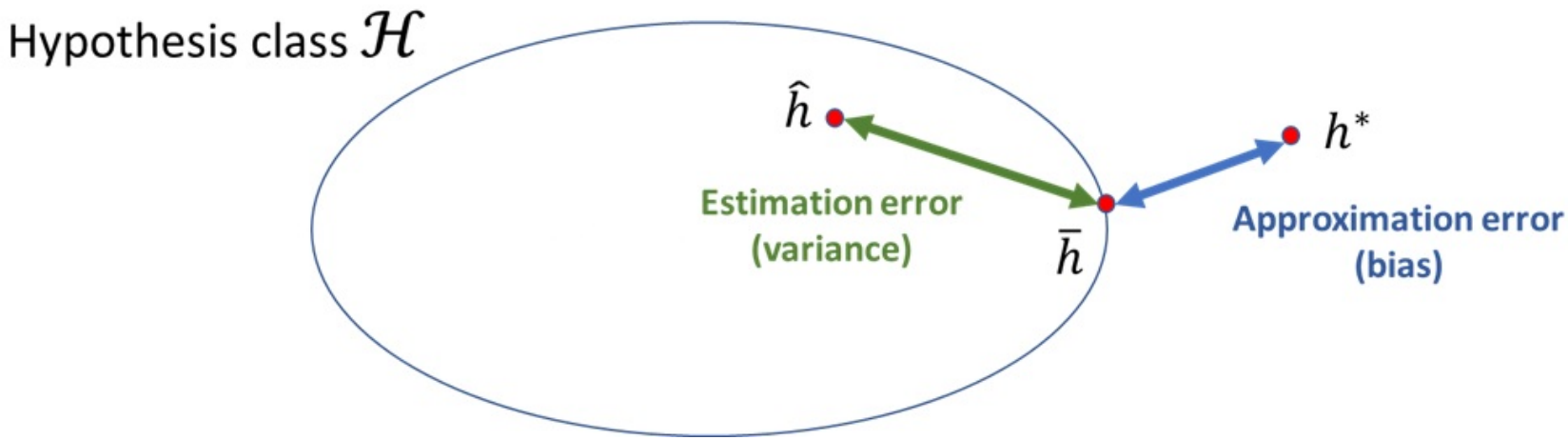


- What is the search space for such a function ?

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 families of ML algorithms

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

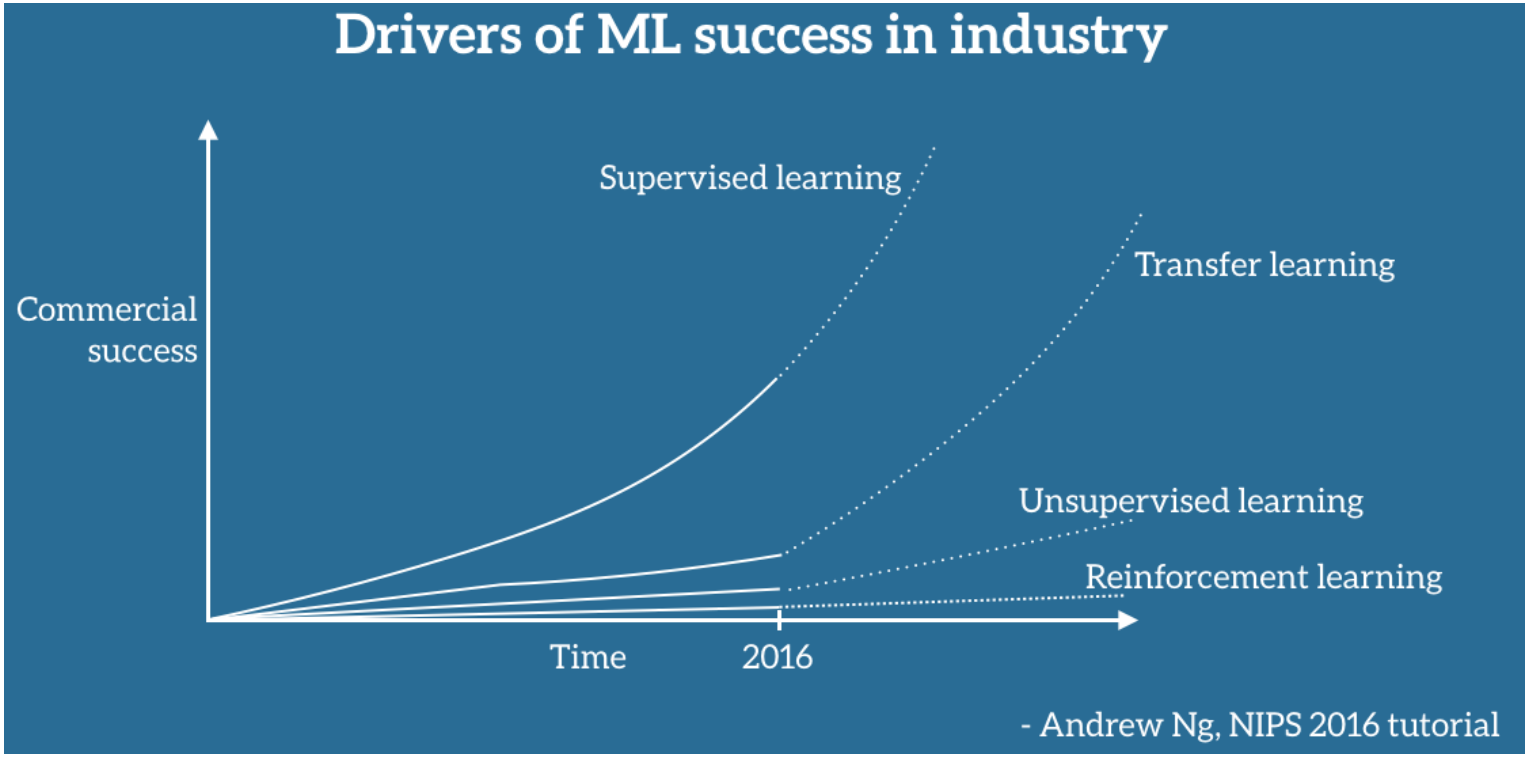
$$\min_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i) + \lambda_n \cdot \mathrm{pen}(h, n) \right)$$

  - 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

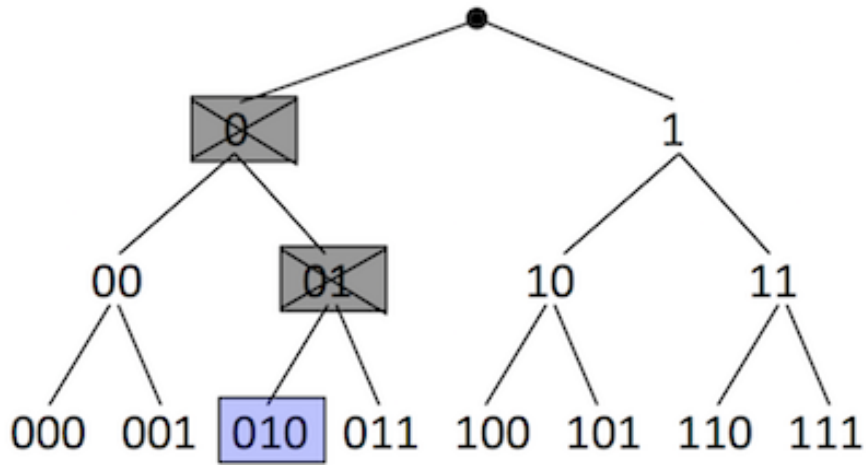
- Assume Nature has picked one function among K and we want to reveal this function
- Assume we have an oracle answering YES or NO when we ask a question about this function
- What is the optimal number n of questions to ask to find the unknown function ?

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 $n = \left\lceil \frac{\log K}{\log 2} \right\rceil = \lceil \log_2 K \rceil$
- 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 $\log K$

- Number of bits to encode a collection of K functions where each function can occur with probability $P(k)$



- Entropy of a distribution P in information theory :

$$H(P) = -\sum_{k=1}^{K} P(k) \log_2 P(k) \leq \log_2 K$$

From questions to data

a. Exhaustive search in the zero-error case
b. PAC learning in the zero-error case
c. PAC learning in the general case

PAC = Probably Approximately Correct

L. Valiant (1984). A theory of the learnable. Communications of the ACM.

From questions to data
a. Zero error case (1/2)

- Notations : Domain space $\mathcal{X}$ and label space $\mathcal{Y} = \{0, 1\}$
- Zero-error setup : Consider a finite set of indicator functions

$$f_k : \mathcal{X} \to \{0, 1\}, \quad k = 1, \ldots, K$$

and a collection of data points (x_i, y_i) such that there always exists some k for which $y_i = f_k(x_i)$, for any index i

- Worst case scenario : the collection of data points $x_i \in \mathcal{X}$ is such that the cardinality of the set of vectors $\{(f_1(x_i), \ldots, f_K(x_i)) : i \geq 1\}$ is maximal and equal to 2^K

From questions to data
a. Zero error case (2/2)

- ISSUE : How many examples $(x_i, y_i) \in \mathcal{X} \times \{0, 1\}$ to find the unknown indicator function among K possible indicator functions $f_k : \mathcal{X} \to \{0, 1\}$, $k = 1, \ldots, K$?
- SAME ANSWER : Number of examples $n = \left\lceil \frac{\log K}{\log 2} \right\rceil$
- 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.

From questions to data
b. PAC in the zero-error case

- REMARK : it may be hard to find such an x_i which splits every subset of functions into two equal parts.
- SAMPLING : Assume $X_1, \ldots, X_n$ is an IID sample
- QUESTION : In the zero-error setup, 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 whose error probability is ε-close to zero with probability $1 - \delta$?
- ANSWER : Number of examples

$$n = \left\lceil \frac{\log K + \log(1/\delta)}{\varepsilon} \right\rceil$$

(Proof left as an exercise)

From questions to data
c. PAC in the general case

- ASSUME : among K functions, NONE of them commits zero error on the sample $\{(X_i, Y_i) : i \geq 1\}$.
- SAME ISSUE AS BEFORE
- ANSWER : Number of examples on average

$$n = \left\lceil \frac{\log K + \log(1/\delta)}{2\varepsilon^2} \right\rceil$$

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

PAC bound - General case
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 :

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

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 random sampling ? the sample may not contain the optimal set of "questions"....

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

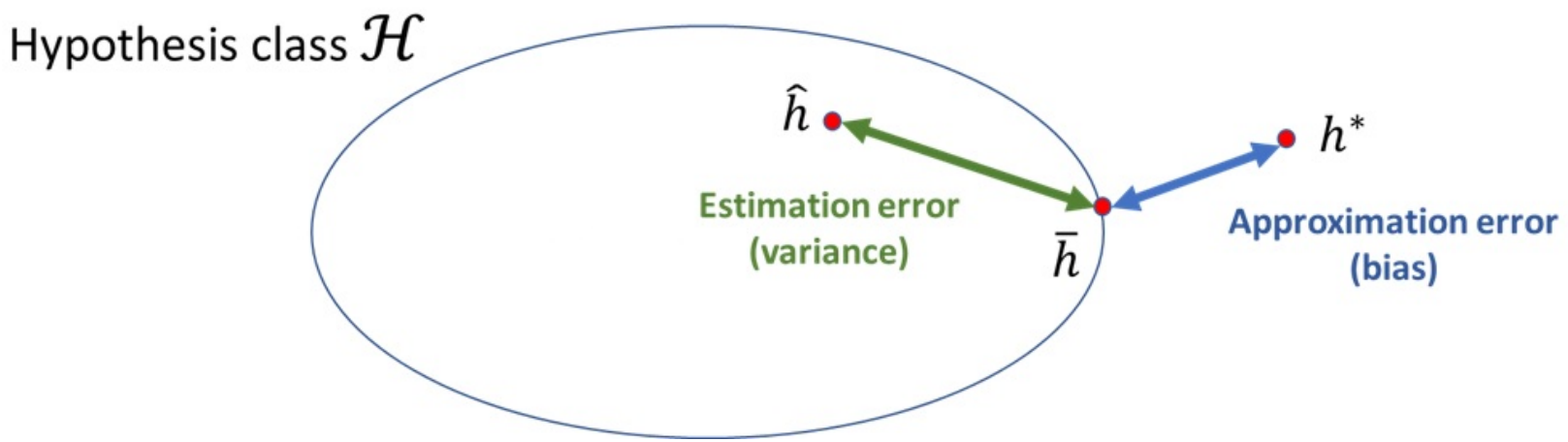
    $$D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

    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 students (should) know

The bias-variance trade-off in statistical inference

The case of parametric estimation

- Θ is a parameter set, subset of $\mathbb{R}^p$
- P_θ with $\theta \in \Theta$ is a parametric class of distributions
- P_{θ^*} is the true distribution of the data for some $\theta^* \in \Theta$ (assumption)
- $\widehat{\theta}_n$ is an estimator of θ^* based on a sample of size n
- Mean-squared error decomposition :

$$\mathbb{E}\left(\|\widehat{\theta}_n - \theta^*\|^2\right) = \mathbb{E}\left(\left((\widehat{\theta}_n - \mathbb{E}(\widehat{\theta}_n))(\widehat{\theta}_n - \mathbb{E}(\widehat{\theta}_n))^T\right)\right) + \|\mathbb{E}(\widehat{\theta}_n) - \theta^*\|^2$$

Exercise : compute MSE in the case of OLS in regression and ridge regression.

The case of prediction error in ML

- Observations : scalar Y, d-dimensional covariate vectors X
- Regression model : $Y = h^*(X) + \varepsilon$
- Random noise : ε independent of X
- Sample-based predictor $\widehat{h}_n$
- Mean-squared error at a fixed point (x, y) :

$$\mathbb{E}_n\left((y - \widehat{h}_n(x))^2\right) = \left(\mathbb{E}_n(\widehat{h}_n(x)) - h^*(x)\right)^2 + \mathbb{V}_n\left(\widehat{h}_n(x)\right) + \varepsilon^2$$

($\mathbb{E}_n, \mathbb{V}_n$: expectation and variance wrt training sample)

MSE = Squared-bias term + Variance of predictions + Bayes error

The case of linear models

- True model : $h^*(x) = x^T\theta^*$,for some $\theta^* \in \mathbb{R}^d$
- Linear models : $h(x) = x^T\theta$, where $\theta \in \mathbb{R}^d$
- Matrix/vector notations : $\mathsf{X} \in \mathbb{R}^{n\times d}$, $\mathsf{Y} \in \mathbb{R}^d$, $\boldsymbol{\varepsilon} \in \mathbb{R}^n$
- Assumption : X is of full rank
- MLE : $\widehat{\theta}_n^{\mathrm{MLE}} = (\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T\mathsf{Y}$
- Plugin predictor : $\widehat{h}_n(x) = x^T\widehat{\theta}_n^{\mathrm{MLE}}$
- Computations : for fixed (x, y)
  - Bias term : $x^T\mathbb{E}_n(\widehat{\theta}_n^{\mathrm{MLE}} - \theta^*) = 0$
  - Variance term : $\mathbb{E}_n\left((x^T(\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T\boldsymbol{\varepsilon})^2\right)$

What theory says : By Gauss-Markov theorem, MLE is the lowest variance unbiased estimator... but not necessarily the one with minimal MSE.

Variance computation
Gaussian model

- Assumption : $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$ n-dimensional multivariate gaussian
- Variance term : for any x

$$\begin{aligned}\mathbb{E}_{y|x}\mathbb{V}_n\left(x^T\widehat{\theta}_n\right) &= \mathbb{E}_{y|x}\mathbb{E}_n\left((x^T(\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T\varepsilon)^2\right)\\ &= x^T\mathbb{E}_n\left((\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T\mathbb{E}_{y|x}(\varepsilon\varepsilon^T)\mathsf{X}((\mathsf{X}^T\mathsf{X})^{-1})^T x\right)\\ &= \sigma^2 x^T\mathbb{E}_n\left((\mathsf{X}^T\mathsf{X})^{-1}\right)x\end{aligned}$$

- Assumption : random design $x, x_i \sim \mathcal{N}(0,1)$ IID
- $\mathbb{E}_x\mathbb{E}_{y|x}\mathbb{V}_n\left(x^T\widehat{\theta}_n\right) = \sigma^2 \cdot \dfrac{d}{n}$

General argument relies on Cochran's theorem (gaussian case).

Explanation of the d/n term

Property on the norm of projections of gaussian random vectors :

- Assume 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 $\Pi_{\mathcal{H}} Z$ has gaussian distribution $\mathcal{N}_n(0, \Pi)$ on $\mathbb{R}^n$ (linear transformation of a gaussian is a gaussian)
- Furthermore : $\|\Pi Z\|^2$ follows a chi-square distribution with

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

From gaussian linear regression to ML

- What if d larger than n ?
- What replaces the dimension d in nonlinear models ?
- Other tasks than regression ?

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

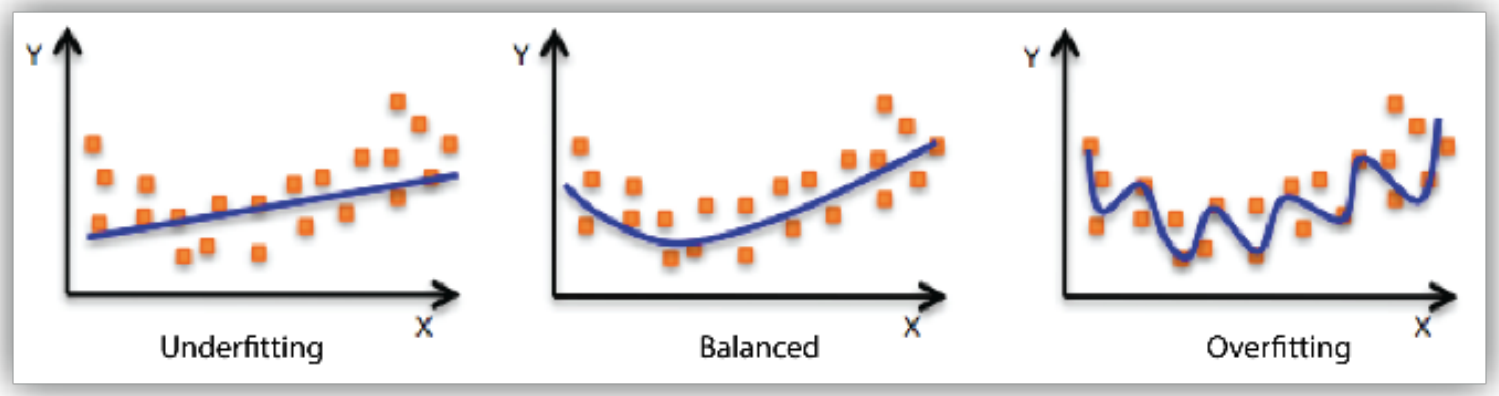
$$\widehat{h}_n = \arg\min_{h \in \mathcal{H}} \widehat{L}_n(h)$$

  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

- Supervised Machine Learning
  - Linear models for supervised learning (2 sessions)
  - From linear to for nonlinear models (2 sessions)
- Unupervised Machine Learning
  - Dimension reduction and clustering (1 session)