I. Introduction

→ predictive maintenance, mobile health monitoring, drug discovery, recommandat° Σ, object recognit°

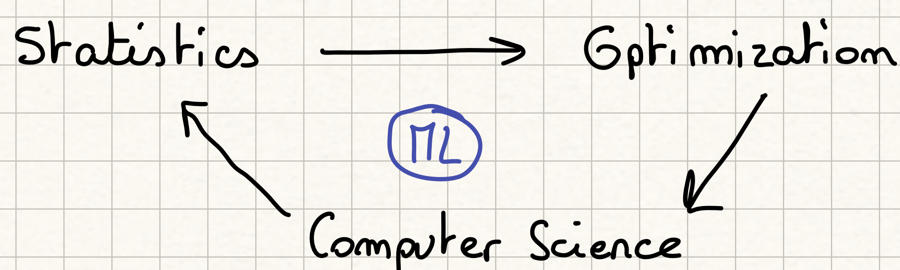
ChatGPT biased wtf qui s'y attendait ?!?

En gros ya du machine learning un peu partout

Def: type d'IA qui fait une tâche sans avoir été explicitement programmé pour.

Def by Tom Mitchell: learn from E with respect to class of task T and performance measure P

E = experience : data provided
T = task
P = accuracy on new data, ability to generalize



Supervised machine learning

- Predictive model: approximate a target function
- Conditional generative modeling approximate a target conditional distribution.

Unsupervised machine learning

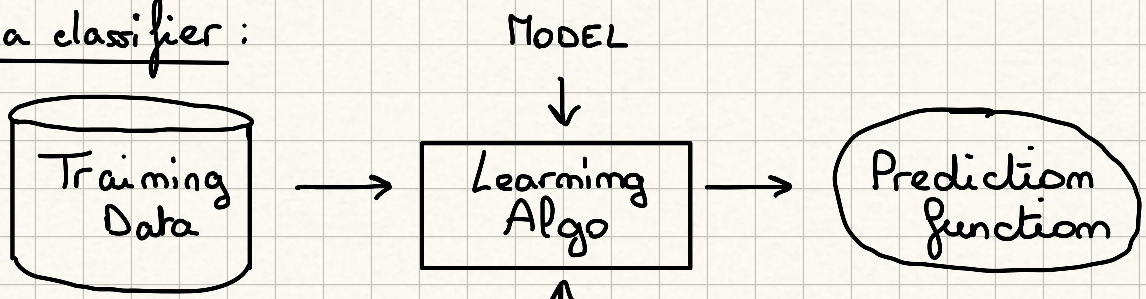
- Generative modeling (Gen AI): approximate a target dist.
- Clustering, Representation Learning, Dimension reduction

Learning paradigms

Customization learning VS Task-driven learning

II - Introduction to Supervised Learning with hands

Learning a classifier :



Loss function, penalties, hyper-param.

in scikitlearn
clf . fit (Xtrain, ytrain)

What do we need ?

→ Data representation
→ Output
→ Hypothesis space
→ Learning algorithm
→ evaluation

III - Probabilistic and statistical setting of Supervised Learning

Risk of a predictive model :

- local loss function : $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
- $R(g) = \mathbb{E}_{(X,Y)\sim P} [\ell(Y, g(X))]$

Min of true risk : $\underset{g : X \to Y}{\operatorname{argmin}} R(g)$

Binary classification rule

$$\ell(g(X), Y) = \mathbb{1}_{y \neq g(x)} \Rightarrow \arg\min_{g : X \to Y} P(Y \neq g(X))$$

Rappel : Bayes Rules

likelyhood

prior probability

$$\mathbb{P}(Y = k \mid x) = \frac{p(x \mid Y = k)\,\mathbb{P}(Y = k)}{p(x \mid Y = -1)\,\mathbb{P}(Y = -1) + p(x \mid Y = 1)\,\mathbb{P}(Y = 1)}$$

posterior proba

proba density

Bayes classifier :

$$g_{bayes}(x) = \mathbb{1}(\mathbb{P}[Y = 1 \mid X = x] > 0{,}5)$$

on dit que ça vaut 1 ou −1

bayes classifier achieves the minimal risk for the classification loss

/!\ bayes risk is characteristic of the "complexity" of the joint probability distribution P and the loss

Sum up : Target function in

- supervised classification : Bayes classifier for the 0-1 loss
- regression : $h(x) = \mathbb{E}[Y \mid x]$ for the square loss

en général, target funct° depend bcp de la loss

Statistical supervised learning problem

find a classifier (regressor) in $\mathcal{G}$ that minimizes

$$R(g) = \mathbb{E}_{(X,Y) \sim P}[\ell(g(X), Y)]$$

only on a finite training sample $S_n = \{(x_i, y_i)_{i=1}^{n}\}$

We look for $g^* \in \arg\min_{g : \mathcal{X} \to \mathcal{Y}} R(g)$ by providing an estimate $\hat{h} \in \mathcal{G}$ of g^* from S_n

IV - Minimization of the empirical risk

Empirical risk : $R_n(g) = \frac{1}{n} \sum_{1}^{n} \mathbb{1}(g(x_i) \neq y_i)$

Statistical learning by empirical risk minimization :

$$\hat{g} \in \arg\min_{g \in \mathcal{G}} R_n(g)$$

excess risk : $R(\hat{g}) - R^* = \underbrace{R(\hat{g}) - \inf R(g)}_{\text{estimation error}} + \underbrace{\inf R(g) - R^*}_{\text{approximation error}}$

$R^* = R(g^*)$

Risk convexification :

Pb : empirical risk hard to minimize
=> on prend $u \mapsto \ell(u,v)$ convexe, differentiable
↳ une autre loss function

Misclassification : $\ell(g(x), y) = \mathbb{1}(y g(x) < 0)$

Exponential : $\ell(g(x), y) = e^{-y g(x)}$

Binomial : $\ell(g(x), y) = -\log(1 + e^{-2y g(x)})$

Squared error : $\ell(g(x), y) = (1 - y g(x))^2$

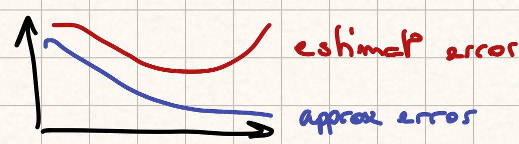
Summary : For (X, Y) et ℓ on veut un classifier proche de
$$g^* = \operatorname{argmin} E(\ell(g(X), Y))$$

Strategie : training sample of $(X,Y) \rightarrow$ on minimise le risque empirique

Method : Numerical optimizat° (ex: descente de grad)

V_ Relevance of Empirical risk minimization

Compromise biais / variance :



- if model too simple $\rightarrow$ large biais, no universality $\Rightarrow$ Underfitting
- if model too complex $\rightarrow$ large variance, no consistency $\Rightarrow$ Overfitting

Vapnik and Chervonenkis's result

$$\forall \mathbb{P},\ S_n \text{ drawn from } \mathbb{P},\ \forall h \in \mathcal{H},\ R(h) \leq R_n(h) + \overbrace{B(d, n)}^{\text{biais}}$$

measure of complexity of $\mathcal{H}$

Theorem : Soit $\mathcal{H}$ une famille de fonctions prenant des valeurs dans $\{-1, 1\}$ de VC-dim d_{VC}

Alors $\forall \delta > 0$, $\forall h \in \mathcal{H}$ avec proba $1-\delta$

$$R(h) \leq R_n(h) + \sqrt{\frac{8 d_{VC}\left(\ln \frac{2n}{d_{VC}} + 1\right) + 8 \log\left(\frac{4}{\delta}\right)}{n}}$$

Idée : on veut controller la complexité de $\mathcal{H}$ et réduire l'erreur empirique

⇒ on remplace la minimisat° du risk empirique par celle du structural risk

Shattering : $\mathcal{H}$ is said to shatter a set of data points if, $\forall$ 2^n possible assignments of binary labels to those points, $\exists\, h \in \mathcal{H}$ tq h ne fait $\emptyset$ erreur de prédiction

VC-dimension : size of the largest set that can be fully shattered by $\mathcal{H}$

$$d_{vc}(\mathcal{H}) = \max\{m : \exists (x_1, \dots x_m) \in \mathcal{X}^m \text{ shattered by } \mathcal{H}\}$$

nb: si $d_{vc} = d$ ça veut dire que ya un set de taille d

pas que tous les sets de taille d ou ⊖ sont shatterables.

VC-dimension of hyperplanes

$$d_{vc}(\mathcal{H}_d) = d + 1$$

hyperplanes in $\mathbb{R}^d$

Regularisation : $+ \lambda \Omega(h)$

role = control of the model complexity

→ imposition of some prior knowledge

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_i^n \ell(y_i, h(x_i)) + \lambda \Omega(h)$$

reg. parameter (hyperparameter)

complexity penalty