Fast supervised learning with guarantees

Ulysse Marteau-Ferey – Inria, ENS Paris, Sierra Project-Team

February 16, 2021

Overview

Introduction

Construction of a criterion of choice

Note on model selection

Finding the parameters

My phd

Supervised learning

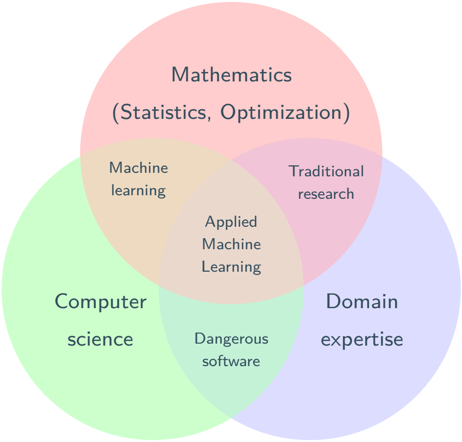
- Empirical risk minimization: OLS, Logistic regression, Ridge, Lasso, Quantile regression
- Deep learning

Introduction

What is ML?

Machine Learning : artificial intelligence which can learn and model some phenomena without being explicitly programmed

$$\text{Machine Learning} \subset \text{Statistics} + \text{Computer Sciences}$$

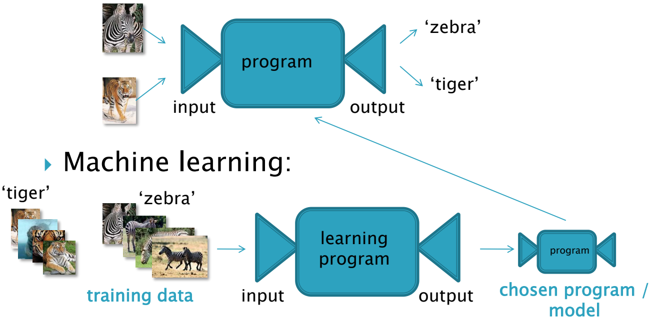


What is ML?

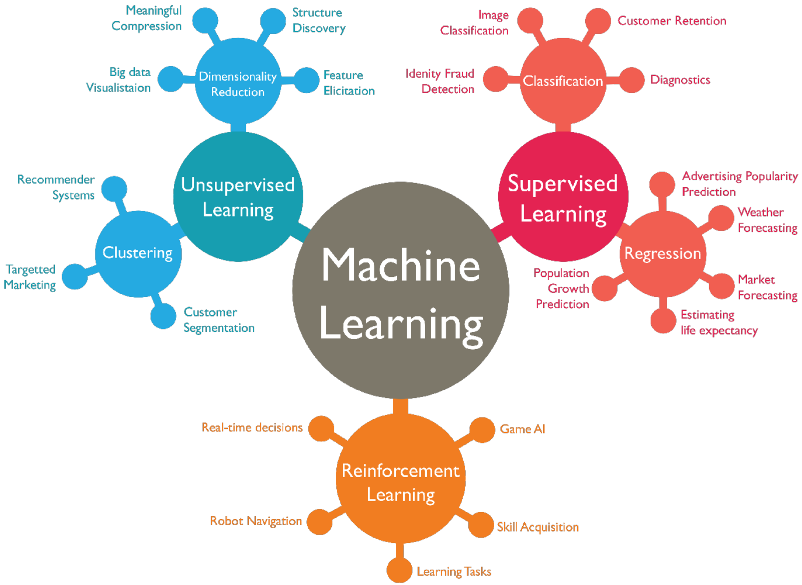
Machine Learning : artificial intelligence which can learn and model some phenomena without being explicitly programmed

$$\text{Machine Learning} \subset \text{Statistics} + \text{Computer Sciences}$$

- Traditional programming:
- Machine learning:



Overview of Machine Learning

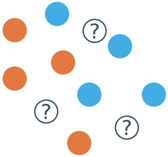


Supervised learning

Predict output Y from some input data X. The training data has a known label Y.

Examples:

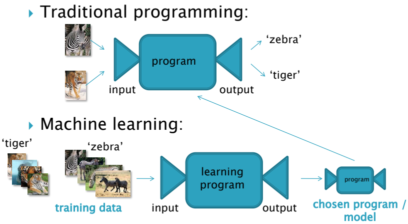
- X is a picture, and Y is a cat or a dog
- X is a picture, and $Y \in \{0, \ldots, 9\}$ is a digit
- X is are videos captured by a robot playing table tennis, and Y are the parameters of the robots to return the ball correctly
- X is a music track and Y are the audio signals of each instrument



Dog or cookie ?

More formal formulation

Goal : Predict output $Y \in \mathcal{X}$ from some input data $X \in \mathcal{X}$.
Predictor (program) : $f : \mathcal{X} \to \mathcal{Y}$: find f such that $f(X) = Y$



Machine learning model : $\{f_\theta \ : \ \theta \in \Theta\}$: set of predictors
Machine learning algorithm (learning program) : Chooses the best θ (i.e. best f_θ)

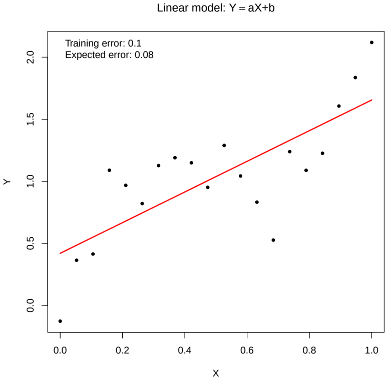
Training data : Examples $(x_1, y_1), \ldots, (x_n, y_n)$.

Example : linear model

Goal : Predict output $Y \in \mathbb{R}$ from some input data $X \in \mathbb{R}$.
Predictor (program) : $f_{a,b}(x) = ax + b$

Machine learning model : $\{f_{a,b} \ : \ (a,b) \in \Theta = \mathbb{R}^2\}$: set of predictors
Machine learning algorithm (learning program) : Chooses the best a, b such that the line $aX + b$ fits Y

Training data : Examples $(x_1, y_1), ..., (x_n, y_n)$.



Rest of the talk

- Which θ do we want the algorithm to choose ? What are the criteria (Statistics) ?
- How do we choose the best model ? (line, parabola ?)
- How do we find the θ effectively (Optimization) ?

Construction of a criterion of choice

Two points of views

- Idealistic point of view : best performance for **every possible example** (x, y)
- The computer point of view : only sees examples $(x_1, y_1), ..., (x_n, y_n)$.

Need for both criterion : one to find the "best" θ during the algorithm, and one to evaluate it afterwards.

Using a loss function

Goal: from training data, we want to predict an output Y (or the best action) from the observation of some input X using our model f_θ

Difficulties: Y is not a deterministic function of X. There can be some **noise**.

Loss function: ℓ to measure the difference between prediction $f_\theta(X)$ and the truth Y :

$$\text{Loss given an example } (x, y) : \ell_{x,y}(\theta) = \ell(f_\theta(X), Y)$$

	Least square regression	**Classification**
$\mathcal{A} = \mathcal{Y}$	$\mathbb{R}$	$\{0, 1, \dots, K-1\}$
$\ell(a, y)$	$(a-y)^2$	$\mathbb{1}_{a \neq y}$
$R(f)$	$\mathbb{E}\left[(f(X) - Y)^2\right]$	$\mathbb{P}(f(X) \neq Y)$
f^*	$\mathbb{E}[Y \mid X]$	$\arg\max_k \mathbb{P}(Y = k \mid X)$

Ideal problem vs algorithmic problem

- **Ideal problem :** as if we had seen everything.
  We define the risk

$$R(\theta) := \mathbb{E}\left[\ell_{X,Y}(\theta)\right] \qquad = \quad \text{expected loss at } \theta$$

  Ideal goal : minimize R
- **Algorithmic problem :** using only the training data.
  Idea: estimate $R(\theta)$ thanks to the training data with the empirical risk

$$\underbrace{\hat{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell_{X_i,Y_i}(\theta)}_{\text{average error on training data}} \quad \approx \quad \underbrace{R(\theta) = \mathbb{E}\left[\ell_{X,Y}(\theta)\right]}_{\text{expected error}}$$

  We tell our algorithm to find $\hat{\theta}_n$ by minimizing the empirical risk

$$\hat{\theta}_n \in \operatorname*{arg\,min}_{\theta \in \Theta} \hat{R}_n(\theta)\,.$$

Note on model selection

Model selection : decomposition of the error

$$\underbrace{\hat{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell_{X_i,Y_i}(\theta)}_{\text{average error on training data}} \approx \underbrace{R(\theta) = \mathbb{E}\left[\ell_{X,Y}(\theta)\right]}_{\text{expected error}}$$

We tell our algorithm to find $\hat{\theta}_n$ by minimizing the empirical risk

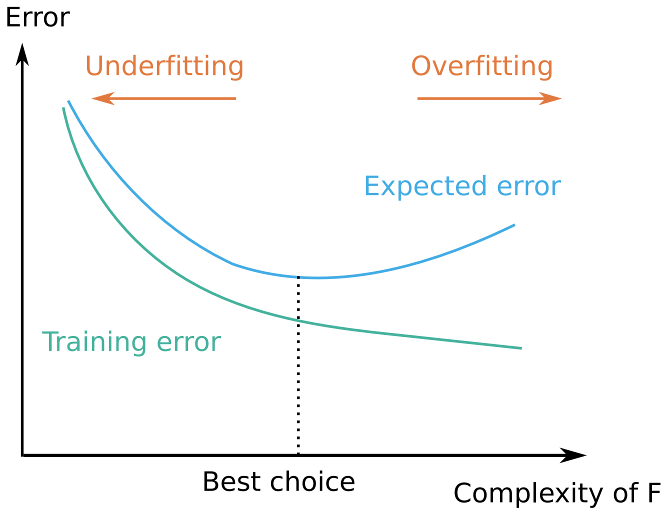
$$\hat{\theta}_n \in \arg\min_{\theta \in \Theta} \hat{R}_n(\theta) \,.$$

Decomposition of the error

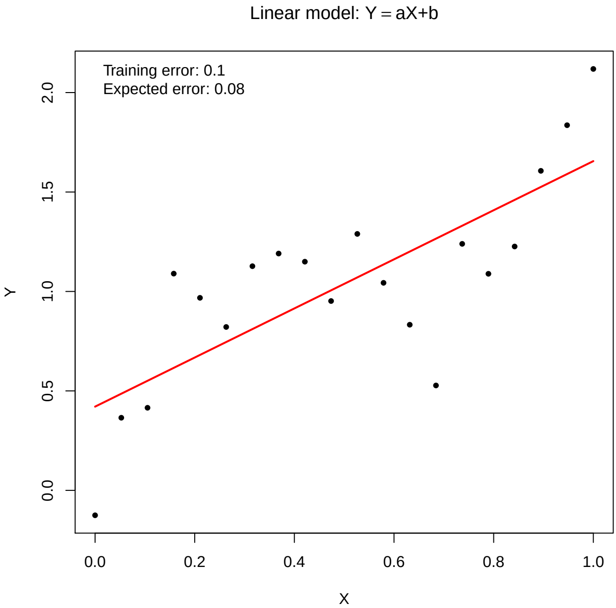
- **Approximation error** : model
- **Estimation error** : number of examples

$$R(\hat{\theta}_n) = \underbrace{\min_{\theta \in \Theta} R(\theta)}_{\text{Approximation error}} + \underbrace{R(\hat{\theta}_n) - \min_{\theta \in \Theta} R(\theta)}_{\text{Estimation error}}$$

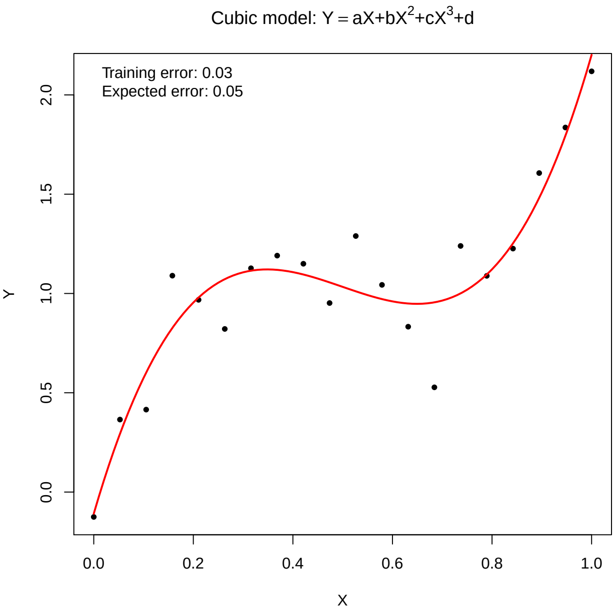
Overfitting



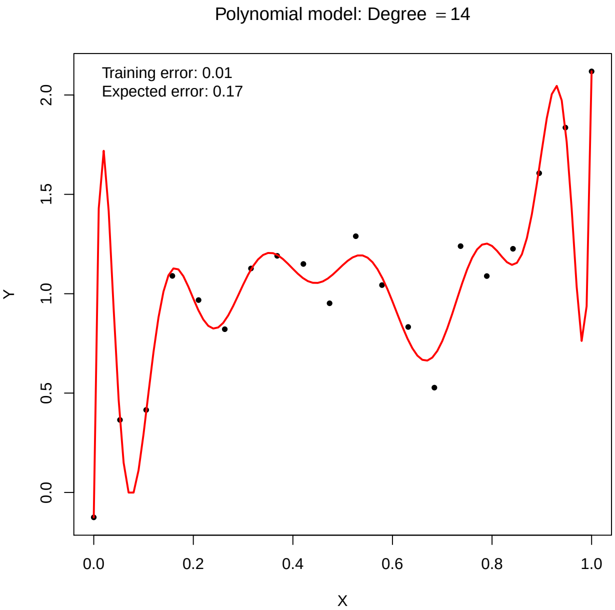
Overfitting: example in regression



Overfitting: example in regression



Overfitting: example in regression



Finding the parameters

Optimization

$$\underbrace{\hat{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell_{X_i,Y_i}(\theta)}_{\text{average error on training data}} \approx \underbrace{R(\theta) = \mathbb{E}[\ell_{X,Y}(\theta)]}_{\text{expected error}}$$

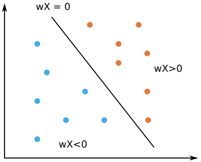
We tell our algorithm to find $\hat{\theta}_n$ by minimizing the empirical risk

$$\hat{\theta}_n \in \arg\min_{\theta \in \Theta} \hat{R}_n(\theta) .$$

How do we solve this problem ? Optimization

Linear classification

Predict binary label $Y \in \{0,1\}$ from X

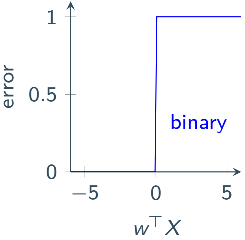


Best linear classifier such that

$$f_{a,b}(X) = aX + b \begin{cases} \geqslant 0 & \Rightarrow \quad Y = +1 \\ < 0 & \Rightarrow \quad Y = 0 \end{cases}$$

Binary loss : ℓ is 0 if right, and 1 if wrong.

$$\hat{a}, \hat{b} = \arg\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_{a,b}(X_i)) \,.$$

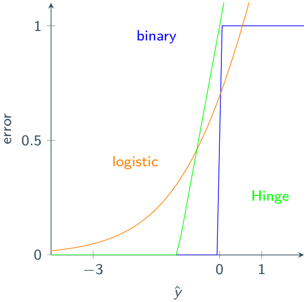


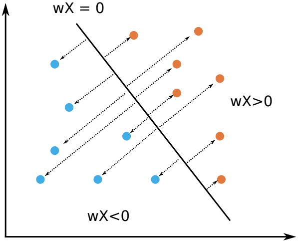
This is not convex in a, b. Very hard to compute!

Optimization interacts with model selection... through the loss !!

Idea: replace the loss with a convex loss

$$\ell(w^\top X, y) = y \log\left(1 + e^{-w^\top X}\right) + (1 - y)\log\left(1 + e^{w^\top X}\right)$$





Convex losses : fast algorithms with guarantees

My phd

My phd

- Provide theoretical bounds in n (guarantees !)

$$R(\hat{\theta}_n) = \underbrace{\min_{\theta \in \Theta} R(\theta)}_{\text{Approximation error}} + \underbrace{R(\hat{\theta}_n) - \min_{\theta \in \Theta} R(\theta)}_{\text{Estimation error}}$$

- Provide a fast convex optimization algorithm to compute $\hat{\theta}_n$.

Thank you for you attention !!