RezelIA

An Introduction to AI

Vincent Bardusco, MVA (ENS Paris Saclay)
Denis Fouchard, Inria Saclay (MIND team)

Machine Learning
Deep Learning
Artificial Intelligence
…
First : data science !

Data
• Image (PNG, JPG) • Text • Sound • Coordinate • Arrays • …

Task
• Classification • Regression • Clustering • Generation • …

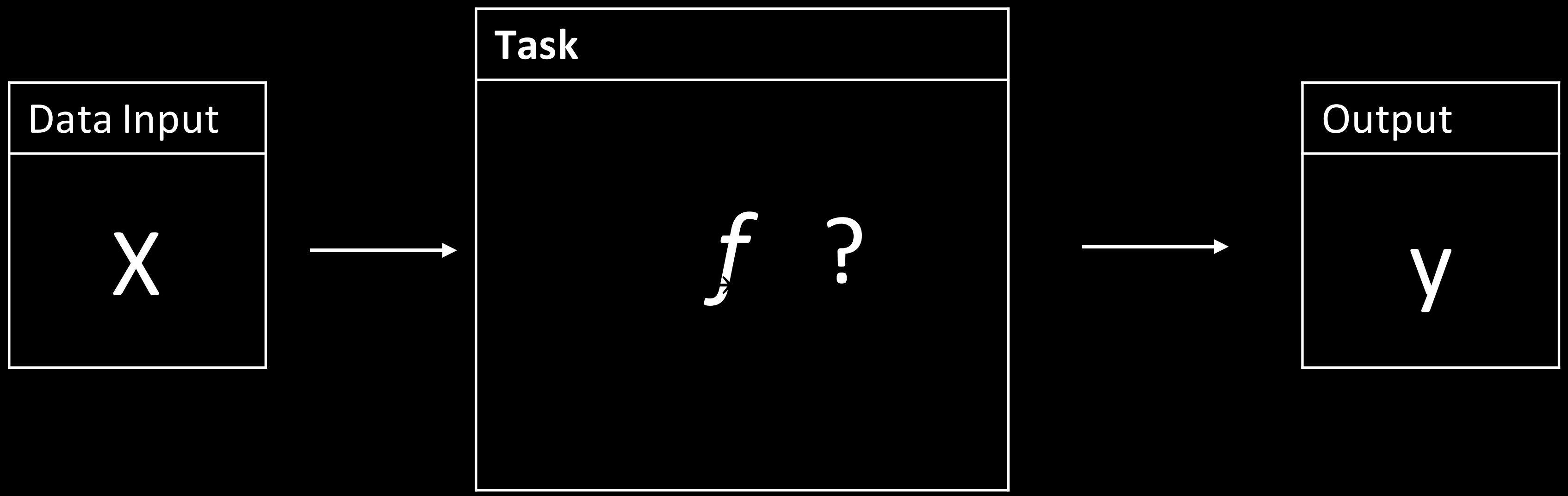
Data

- Image (PNG, JPG)
- Text
- Sound
- Coordinate
- Arrays
- …

Task

f ?

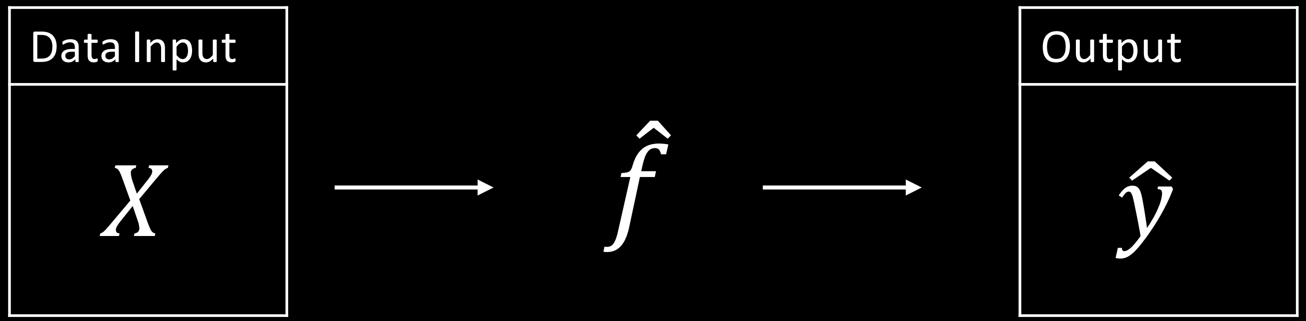
Objective


Data Input
X
Task
f ?
Output
y

How to find f

Start with $\hat{f}$

$\hat{f}$ is an **estimator** of f


Data Input
X
$\hat{f}$
Output
$\hat{y}$

Output

$\hat{y}$

Score : 0.145…

Scoring

- F1 score
- Avg accuracy
- Mean-Square Error
- …

Define a loss function L

Loss

- MSE
- Cross-Entropy
- Log Loss
- [Custom Loss]

Task	Error type	Loss function	Note
Regression	Mean-squared error	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Easy to learn but sensitive to outliers (MSE, L2 loss)
	Mean absolute error	$\frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$	Robust to outliers but not differentiable (MAE, L1 loss)
Classification	Cross entropy = Log loss	$-\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] =$	Quantify the difference between two probability

Define a loss function L

Loss
• MSE • Cross-Entropy • Log Loss • [Custom Loss]

$$L(\hat{y}, y) \rightarrow 0$$

Data Input
X

$$X \longrightarrow \hat{f} \longrightarrow \hat{y} \longrightarrow L(\hat{y}, y)$$

Output
$\hat{y}$

Learn
• Iteratively (gradient descent, SRM…) • Analyticly (Ridge Regression)

Data Input

X

Task

f

Output

y

Estimator

$\hat{f}$

Loss function

L

Scoring

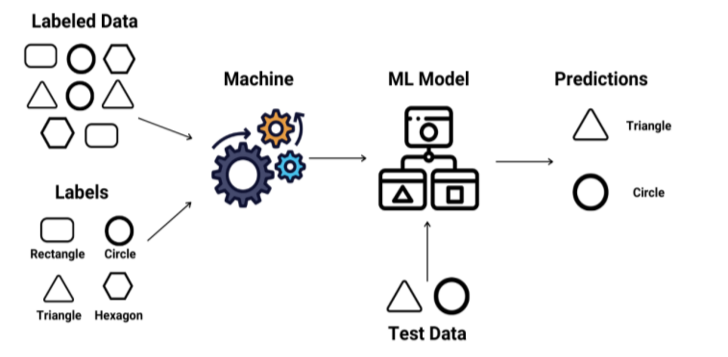
Learning procedure

- Iteratively (gradient descent, SRM…)
- Analyticly (Ridge Regression)

Supervised learning

- Clear objective (labels, …)
- Prediction
- Regression
- The loss is defined by the data

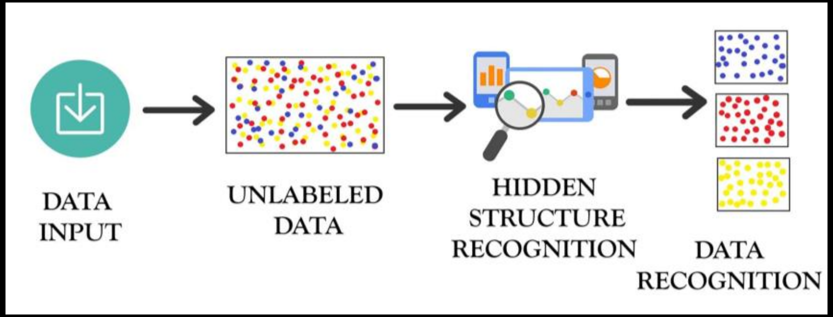
Classification
Regression
Optical Character Recognition (OCR)



Unsupervised learning

- Clustering
- Generation
- The model learns hidden patterns in the data

Clustering
Image segmentation
Text/image generation



x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		x_n

Preprocessing

- Labelisation
- Tokenisation
- Parsing
- Formating
- Shuffling

We do it **on all the data !**

x_4	x_2	x_1	x_n	x_7	x_5	x_3	x_8		x_3

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		x_n

Shuffling

x_4	x_2	x_1	x_n	x_7	x_5	x_3	x_8		x_3

Training data set

x_7	x_5	x_3	x_8		x_3

Testing data set

x_4	x_2	x_1	x_n

x_7	x_5	x_3	x_8		x_3

x_4	x_2	x_1	x_n

Preprocessing
• Regularisation • Normalisation

We do it **separately** on training and testing data !

x_7	x_5	x_3	x_8		x_3

x_4	x_2	x_1	x_n

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		x_n

Preprocessing
• Labelisation • Tokenisation • Parsing • Formating • Shuffling

We do it **on all the data !**

x_4	x_2	x_1	x_n	x_7	x_5	x_3	x_8		x_3

Shuffling

x_4	x_2	x_1	x_n	x_7	x_5	x_3	x_8		x_3

x_7	x_5	x_3	x_8		x_3

x_4	x_2	x_1	x_n

x_7	x_5	x_3	x_8		x_3

x_4	x_2	x_1	x_n

Preprocessing
• Regularisation • Normalisation

We do it **separately** on training and testing data !

x_7	x_5	x_3	x_8		x_3

x_4	x_2	x_1	x_n

Common models

Classification	Regression	Clustering
• Decision trees • Support Vector Machines (SVM) • K-Nearest-neighbours • Neural Networks	• Ordinary Linear Regression(OLS) • Regularised Regression (Ridge, LASSO) • Neural Networks	• K-Means, K-Means++

For anything more complex -> Neural Networks

A specific model: Regression

- What we got: data

(x,y)

- What we want to do: the task

Find the relation between x and y:
$f(x) \approx y$

A specific model: Regression

- What we got: data

(x,y)

The features: what characterizes the phenomenon Ex: physical measures (distances, volumes, …), discrete characteristics (number of occurences of smthing, …), other (name, age, color, …)
$\in R^p$

- What we want to do: the task

Find the relation between x and y:
$f(x) \approx y$

A specific model: Regression

- What we got: data

The target: a number we want to predict Ex: a price, a temperature, ...
$\in R$

(x,y)

The features: what characterizes the phenomenon Ex: physical measures (distances, volumes, …), discrete characteristics (number of occurences of smthing, …), other (name, age, color, …)
$\in R^p$

- What we want to do: the task

Find the relation between x and y:
$f(x) \approx y$

A specific model: Regression

- What we got: data

(x,y)

The target: a number we want to predict Ex: a price, a temperature, ...
$\in R$

Supervised learning

The features: what characterizes the phenomenon Ex: physical measures (distances, volumes, …), discrete characteristics (number of occurences of smthing, …), other (name, age, color, …)
$\in R^p$

- What we want to do: the task

Find the relation between x and y:
$f(x) \approx y$

A specific model: Regression

- What we got: data

(x,y)

The target: a number we want to predict Ex: a price, a temperature, ...
$\in R$

Supervised learning

The features: what characterizes the phenomenon Ex: physical measures (distances, volumes, …), discrete characteristics (number of occurences of smthing, …), other (name, age, color, …)
$\in R^p$

- What we want to do: the task

Find the relation between x and y:
$f(x) \approx y$

In what space?

A specific model: Regression

- What we got: data

The target: a number we want to predict Ex: a price, a temperature, ... → Supervised learning

$\in R$

(x,y)

The features: what characterizes the phenomenon Ex: physical measures (distances, volumes, …), discrete characteristics (number of occurences of smthing, …), other (name, age, color, …)

$\in R^p$

- What we want to do: the task

Find the relation between x and y:

$f(x) \approx y$

What does it mean?

In what space?

1. Model choice: Linear Regression

- Too many possibilities for f, we suppose that there is a linear relation between x and y

$x = (x_1,x_2,\ldots,x_p)$ p features

$y = \theta^*_1 x_1 + \theta^*_2 x_2 + \ldots + \theta^*_p x_p$

$y = \sum \theta^*_j x_j$

- We then restrain ourselves to functions of the type:

$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_p x_p$

1. Model choice: Linear Regression

- Too many possibilities for f, we suppose that there is a linear relation between x and y

$x = (x_1, x_2, \ldots, x_p)$ p features

$y = \theta^*_1 x_1 + \theta^*_2 x_2 + \ldots + \theta^*_p x_p + \varepsilon$

$y = \sum \theta^*_j x_j + \varepsilon$

- We then restrain ourselves to functions of the type:
- $f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_p x_p$

(f is parametric)

2. Vector/Matrix notation

We have a dataset of n samples (n patients, n occurrences, n realisations, ...), each sample has p features

Data samples

x_1	x_{11}	x_{12}	x_{13}	x_{14}		x_{1d}	y_1
x_2	x_{21}	x_{22}	x_{23}	x_{24}		x_{2d}	y_1
			…				…
x_n	x_{n1}	x_{n2}	x_{n3}	x_{n4}		x_{nd}	y_n

Data samples

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

2. Vector/Matrix notation

- We have a dataset of n samples (n patients, n occurrences, n realisations, ...), each sample has p features

$$Y = X\theta^* + \varepsilon$$

- We want to find f of the form:

$f_\theta(X) = X\theta$

- Still an infinite number of possibilities! (an infinite number of parameters theta possible)… How to find the best one?

3. Loss function

- Objective: measure how far our prediction is from the reality on the set of known Y

$$L(Y, f_\theta(X)) = ||Y - f_\theta(X)||^2 = ||Y - X\theta||^2$$

This is the sum of squared differences!

Why this function?

3. Loss function

- Objective: measure how far our prediction is from the reality on the set of known Y

$$L(Y,f_\theta(X)) = ||Y - f_\theta(X)||^2 = ||Y - X\theta||^2$$

This is the sum of squared differences!

Why this function?

- Well measures how bad our predictions are (penalizes more the very bad predictions)
- Easy to optimize (remember we want to minimize it!)
- Theoretical statistical reasons

4. Resolution: find the best estimation

- As f is entirely determined by the parameter θ, finding the best f is the same as finding the best θ

- We're lucky, there is an exact formula to compute the estimator θ that minimizes our loss!

$$\hat{\theta} = argmin_{\{\theta\}}||Y - X\theta||^2$$

$$\hat{\theta} = (X^t X)^{-1} X^t Y$$

- This estimator is called the OLS estimator

5. Predictions: use the model

- We have learned a prediction function on our dataset, we can now use it to predict Y for any X!

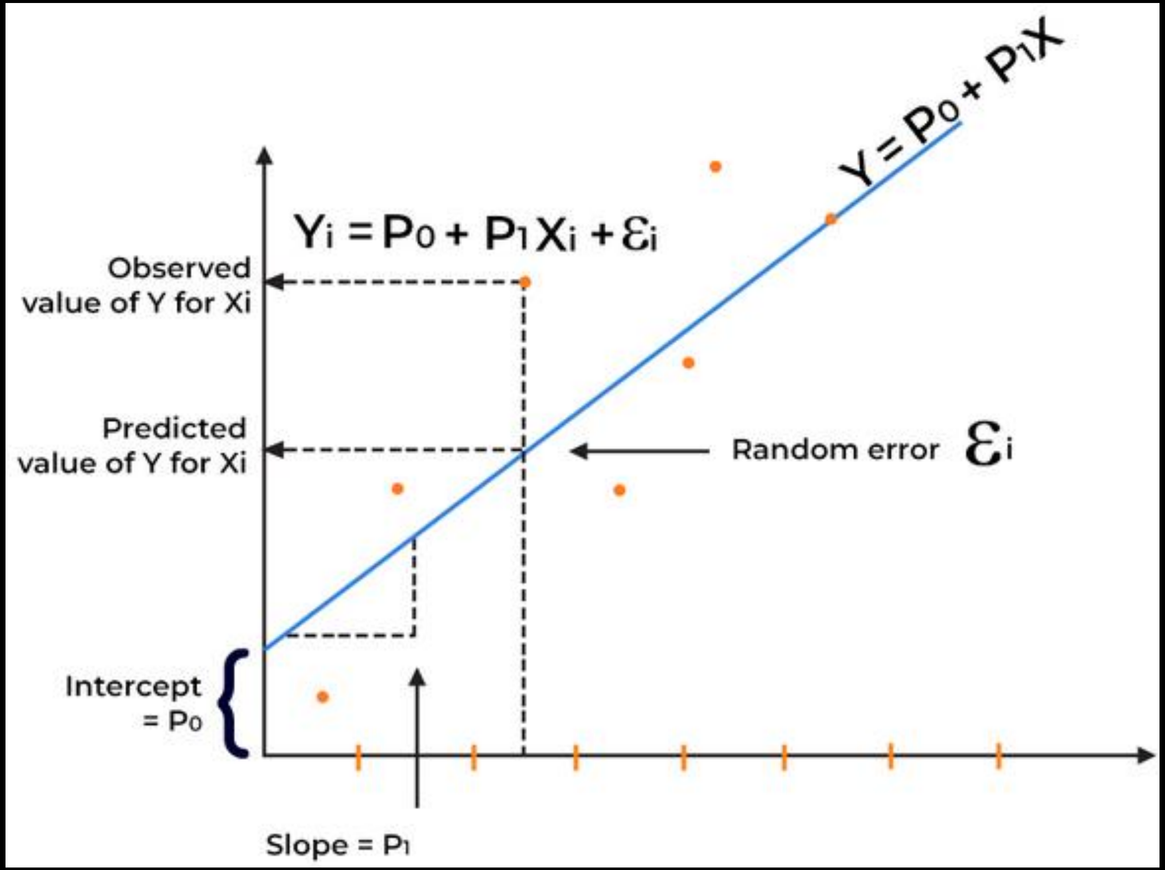
- How?

5. Predictions: use the model

- We have learned a prediction function on our dataset, we can now use it to predict Y for any X!

- How?

Recap on Linear Regression on the simple visual case



A bit more complex…

Task : Ridge Regression
Minimize $\lvert\lvert Y - X\theta \rvert\rvert^2 + \lambda \left\lvert \lvert \theta \rvert \right\rvert^2$

$$\hat{\theta} = argmin_{\{\theta\}} \lvert\lvert Y - X\theta \rvert\rvert^2 + \lambda \left\lvert \lvert \theta \rvert \right\rvert^2$$

$$\hat{\theta} = (X^t X + \lambda I)^{-1} X^t Y$$

Now it is your turn !

Be sure to have installed…

Mandatory

- Python >= 3.9
- Jupyter-Notebook
- Numpy
- Scipy
- Sci-kit learn
- Pandas

Very helpful

- Anacoda/pyenv
- VSCode

Must-have VSCode extensions

- Jupyter
- Github Copilot (but not for this lab…)

Useful ressources

- kaggle.com: datasets, example notebooks
- scikit-learn.org: models, documentation
- microsoft.github.io/AI-For-Beginners/: courses and labs to overview AI techniques
- youtube.com/c/3blue1brown: introduction to statistics and ML concepts