Data Science

Marc Tommasi

September 9, 2025

Outline

1. Introduction
2. Introduction to ML
3. Formal approach to ML

Outline

1. Introduction
2. Introduction to ML
3. Formal approach to ML

Preamble

- Evaluation: CCI
  - Note UE = (Note1 + Note2) avec Note1 = (Exam 1 + Project) /2 et Note2 = DS Final et Second chance : min(10, Note UE + Note Rendus/10), where Note Rendus on the last TPs rendus and homeworks
  - Exam 1: Week 7 (20/10/25)
- Cours moodle : `https://moodle.univ-lille.fr/course/view.php?id=17020`
  - Group 2: wx9j5q
  - Group 1: d3dsuw
  - Group MISO: zk8jhv

Objectives

- Presentation of some well-known methods and algorithms in Data Science
- Introduction to Machine Learning
- Explanation of some theoretical foundations of ML
- Training on practical and technical aspects mainly with Python notebooks, sklearn, Pandas
- You won't be ML/DS experts with only one course!

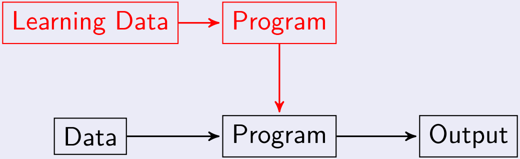
BI vs ML

- BI (business intelligence, informatique décisionnelle)
  - (Big) data analysis,
  - Data availability, data management, data modeling for fast access mainly in read-only mode (Snowflake, stars,. . .), data storage (datawarehouse, datamart,. . .)
  - Reports, Visualization
  - Browsers (drill up, down,. . .)
  - and some advanced techniques of pattern matching, association rules, etc. . .
- ML
  - Prediction models! We build functions (programs) from data that solve tasks for unseen data.

Programmation, IA et ML

Programming

AI and Machine Learning

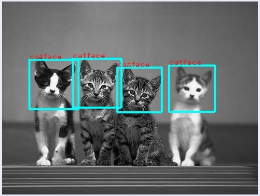


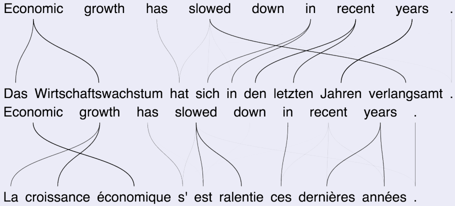
- Classic AI expert systems, rule based systems, knowledge representation, logical reasoning,. . .
- Machine Learning: data driven approaches.

Machine Learning

- ML provides a computer with the ability to do certain tasks without being explicitly programmed for it (Arthur Samuel, 1959)
- This is done by learning from data
- Multidisciplinary field: computer science, statistics, optimization
- ML is fueling the current progress in AI
- The full process can be automated!

Examples of applications





- Requires large amounts of data (potentially *sensitive, personal data*)

Data types

- Images: astronomy, agriculture, weather forecast, archeology, facial recognition, medicine and health (MRI, FRMI, . . .), OCR, autonomous cars,
- Text: Natural Language Understanding (NLU), Text Generation, translation, text classification (Spam,. . .), automatic summarization, . . .
- Sound: Music Information Retrieval (MIR), Speech to text, text to speech (STT, TTS), Chatbots, translation,
- *-Omics data
- Sensor data: transportation, robotics, predictive maintenance, . . .
- Games
- Web data: recommendation, etc. . .

When ML approaches are competitive. . .

- You can't formalize easily the task (you need to automate a task and you can't write a classic program)
- The combinatorics are too large
- You need frequent adaptation or personalization.

Limits and Research themes

- Confidence: Attacks on ML models, Robustness
- Ethics: Privacy, fairness
- Interpretability: understand a model, give an explanation of a result
- Energy consumption
- Society: Respect legal rules, acceptability, understand the difference between automatization and IA/ML.

Conclusion

- Impressive results (mostly with text, images and sounds or videos)
- Transforms all economical sectors,
- Often, immature technologies in constant evolution
- You will need to be highly qualified and follow this evolution.
- Numerous social effects

Contents of this lecture

- Data Exploration : numpy, pandas, matplotlib,. . .
- Some theory, methodology and algorithms for supervised learning
- Theory: Empirical Risk Minimization for supervised classification
- Methodology: Error estimation, model selection, hyperparameter tuning
- Algorithms: linear regression, regularization, classification methods, logistic regression, decision trees, ensemble methods, naive bayes, introduction to neural networks

Outline

1. Introduction
2. Introduction to ML
3. Formal approach to ML

Introduction

- Machine Learning: from experience to knowledge and expertise
- From Tom M. Mitchell:

  A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on task T, measured by P, improves with experience E.

- input: experience are learning data
- output: expertise is a computer program that solves a task.
- Important questions:
  - What kind of data?
  - How to automate the learning process?
  - How to evaluate the success of the learning step?

What is learning?

- Spam detection. Learning by heart: we can memorize all spam messages in a database; answer yes/no depending on the presence of a message in the DB. Is it learning?
- A generalisation step is required. In the context of spam: some words as evidence of spam status
- No free lunch theorem. Learning is impossible without the presence of some bias

Three learning paradigms

- Supervised learning
  - experience: input and output data (e.g. text messages and the spam status): the learner has labeled data and knows the possible set of labels
  - classification: the output is a label in a finite set (e.g. spam, associate a category with a text, medical diagnosis, ...)
  - regression: the output is a continuous value (e.g. a probability, a price,...)
- Unsupervised learning
  - experience: unlabeled data
  - clustering (find groups)
  - density estimation
- Reinforcement learning
  - experience: the learner performs an action and then receives a reward
  - the process is mainly online
  - the objective is to find a good policy of actions.
  - the algorithm should solve a dilemma between exploration and exploitation

Some variants

- Active/passive : the learner acts on the environment (e.g. chooses examples or not).
- Online or batch: prediction or decision taken at each example?
- Parametric: the model is defined by a set of parameters and the learning process has to find the best ones.
- Non parametric: e.g. decision based on the data (e.g. the closest labeled data)

Challenges

- We need data!
- We need good data!
  - sampling bias
  - outliers
  - unrepresentative data
  - insufficient data
- Overfitting
- Underfitting
- Evaluation and comparisons of ML models

Outline

1. Introduction
2. Introduction to ML
3. Formal approach to ML

Supervised Learning

- Spam detection: a first idea of representation, models, errors and generalization

Formalization

- Formalization in the supervised learning framework
- Data are described by some attributes. Data lie in a representation space $x \in \mathcal{X}$
- The target is $y \in \mathcal{Y}$
- A sample $S \subseteq \mathcal{X} \times \mathcal{Y}$
- Output: a prediction rule, a model is a function $f : \mathcal{X} \to \mathcal{Y}$

Vocabulary

- representation space: features, attributes, (voire champs)
- échantillons ou jeu de données, dataset
- instances, records, examples, records, enregistrements
- target, label, or class

Basic assumptions

- The data is generated from a fixed, but unknown data distribution $\mathcal{D}$.
- There exists a target function f that labels the data.
- We consider in the following a classification problem, where Y is a finite set of discrete values
- The true error, the generalization error, the true risk of an hypothesis h is:

$$L_{\mathcal{D},f}(h) = \mathcal{D}(\{x \mid h(x) \neq f(x)\}).$$

- $\mathcal{D}$ is unknown the learner cannot compute $L_{\mathcal{D},f}(h)$

ERM : Empirical Risk Minimization Principle

- The learner can compute the empirical risk or empirical error

$$L_S(h) = \frac{|\{i \in [m] \mid h(x_i) \neq y_i\}|}{m},$$

where $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.

ERM principle

The learner outputs an hypothesis h_S that minimizes the empirical risk

$$h_S = \operatorname*{argmin}_h L_S(h)$$

First limits of the ERM principle

What is the empirical error of this rule?

$$h(x) = \begin{cases} y_i \text{ if there exists } i \text{ s.t. } x = x_i \\ 0 \text{ otherwise} \end{cases}$$

First limits of the ERM principle

What is the empirical error of this rule?

$$h(x) = \begin{cases} y_i \text{ if there exists } i \text{ s.t. } x = x_i \\ 0 \text{ otherwise} \end{cases}$$

- Learning by heart minimizes the empirical error!
- Note that some classes of function can mimic learning by heart (the class of all polynomials)

ERM with a bias

- Bias: we fix a priori an hypothesis class $\mathcal{H}$

$$\mathrm{ERM}_{\mathcal{H}}(S) = h_S \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h)$$

- This can be a background knowledge
- Example: rectangles or lines
- Trade-off:
  - a large and expressive class allows to model more complex tasks but can be subject to overfitting
  - a small and poorly expressive class limits overfitting but increases the risk of having bad performances.

Question

Do we learn when we follow the ERM principle?

Confidence and Approximation

Confidence

- Note that $L_S(h_S)$ is a random variable. The stochasticity comes from the fact that S is drawn from $\mathcal{D}$.
- Let δ be the probability to sample a "bad" sample. Then $1 - \delta$ is the confidence.

Approximation

- The computation of an hypothesis h is error-prone.
- We can tolerate some error up to some threshold ϵ: $L_{\mathcal{D},f}(h_S) \leq \epsilon$
- Bad hypothesis corresponds to $L_{\mathcal{D},f}(h_S) > \epsilon$

The case of finite hypothesis classes

Theorem (Finite classes are learnable under the realizability assumption)

If $\mathcal{H}$ is finite, for any $\epsilon > 0$ and $\delta \in [0, 1]$, for any target function f, for any distribution $\mathcal{D}$ such that there exists $h \in \mathcal{H}, L_{\mathcal{D},f}(h) = 0$ (Realizability assumption), then with probability $1 - \delta$ over the choice of an i.i.d. sample S of size larger than $\frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$, the true error of and ERM hypothesis h_S is lower than ϵ

$$L_{\mathcal{D},f}(h_s) \leq \epsilon.$$

- $\mathcal{H}$ is finite, take $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$ examples in S, then
$$\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq \delta.$$
- Let us compute $\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\})$ to bound the probability that drawing a sample that leads to a failure of the learning process...

A bad sample...

- The ERM hypothesis h_S is wrong if $L_{\mathcal{D},f}(h_S) > \epsilon$ and $L_S(h_S) = 0$ (realizability assumption).
- Let

$$M = \bigcup_{h \text{ s.t. } L_{\mathcal{D},f}(h) > \epsilon} \{S' \mid L_{S'}(h) = 0\},$$

- if S belongs to M then

$$\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq \mathcal{D}^m(M).$$

- Hence applying the union bound ($\mathcal{D}(\bigcup_i A_i) \leq \sum_i \mathcal{D}(A_i)$)

$$\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq \sum_{h \text{ s.t. } L_{\mathcal{D},f}(h) > \epsilon} \mathcal{D}^m(\{S' \mid L_{S'}(h) = 0\}).$$

A bound for the right hand side

- $L_{S'}(h) = 0$ when for any x_i in S' we have $h(x_i) = f(x_i)$. Samples are i.i.d. hence

$$\mathcal{D}^m(\{S' \mid L_{S'}(h) = 0\}) = \prod_i^m \mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}).$$

- But $L_{\mathcal{D},f}(h) = \mathcal{D}(\{x \mid f(x) \neq h(x)\})$ hence

$$\mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}) = 1 - L_{\mathcal{D},f}(h).$$

- If h has an error larger than ϵ then

$$\mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}) \leq 1 - \epsilon,$$

and

$$\mathcal{D}^m(\{S' \mid L_{S'}(h) = 0\}) \leq (1 - \epsilon)^m.$$

End of the proof!

- We combine

$$\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq \sum_{h \text{ s.t. } L_{\mathcal{D},f}(h) > \epsilon} \mathcal{D}^m(\{S' \mid L_{S'}(h) = 0\})$$

and

$$\mathcal{D}^m(\{S' \mid L_{S'}(h) = 0\}) \leq (1-\epsilon)^m.$$

If we denote by $\mathcal{H}_B = \{h \text{ s.t. } L_{\mathcal{D},f}(h) > \epsilon\}$ the set of bad hypothesis

$$\mathcal{D}^m(\{S \mid L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq |\mathcal{H}_B|(1-\epsilon)^m \leq |\mathcal{H}_B| e^{-\epsilon m} \leq |\mathcal{H}| e^{-\epsilon m}.$$

ERM and error estimation

- The final objective is to minimize the true error, but the learner cannot compute it.
- She needs new data (different from the learning data) to estimate the true error.
- An important assumption is that new data will be generated by the same distribution $\mathcal{D}$.
- A practical approach: split the data into 2 sets:
  - a training set to learn a model
  - a test set to estimate the true error