# Machine Learning

Federico Matteoni

A.A. 2021/22

# Index

## 0.1  Introduction

What is ML? Area of research combining aims of creating computers that could learn and powerful and adaptive statistical tools with rigorous foundation in computational science. Luxury or necessity? Growing availability and need for analysis of empirical data and difficult to provide intelligence and adaptivity by programming it. Change of paradigm.

Examples: spam classification, written text recognition. . . No or poor prior knowledge and rules for solving the problem, but easier to have a source of training experience.

ML is considered the latest general-purpose technology, capable of drastically affect pre-existing economic and social structures. And already has. The ultimate aim is to bring benefits to the people by solving big and small problems, accelerating human progress and empowering humans to add intelligence in any other science field.

**Machine Learning**   We restrict to the computational framework: principles, methods and algorithms for learning and prediction, from experience. Building a model to be used for predictions. Common framework: infer a model or a **function** from a set of examples which allows the generalization (accurate response to new data).

When can we use ML? Be aware of the opportunity and awareness. ML is useful when there's no or poor theory surrounding the phenomenon, or uncertain, noisy or incomplete data which hinders formalization of solutions. The requests are: source of training experience (representative data) and a tolerance on the precision of results. The best examples are models to solve real-world problems that are difficult to be treated with traditional techniques: face and voice recognition (knowledge too difficult to formalize in an algorithm), predicting biding strength of molecules to proteins (not enough human knowledge) and personalized behavior, such as recommendation systems, scoring messages according to user preferences. . .

**Definition**   The ML studies and proposes methods to build functions/hypothesis from examples of observed data that fits the known examples and able to generalize, with reasonable accuracy, for new data (according to verifiable results and under statistical and computational conditions and criteria.

**Data**   Data **represents the available experience**. Representation problem: capturing the structure of the analyzed objects. Flat (attribute-value), structured. . . , categorical or continuous, missing data. . . **preprocessing**: variable scaling, encoding, selection. . .

**Task**   The task defines the purpose of the application: knowledge that we want to achieve? which is the helpful nature of the result? what information are available?

**Predictive** task, classification and regression: function approximation

**Descriptive** task, cluster analysis and association rules: find subsets or groups of unclassified data.



Also as a guide to the **key design choices**
(ML system **"ingredients"**)
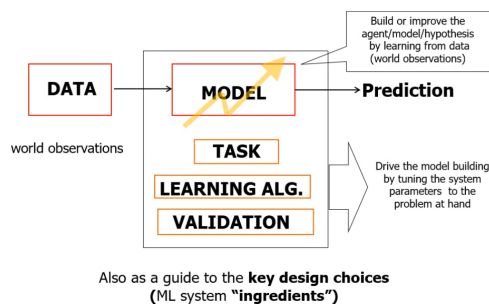
**Supervised learning**

**Learning algorithm**   Basing on data, task and model: heuristic search through the hypothesis space $H$ of the **best hypothesis**. I. e. the best approximation of the unknown target function, typically searching ofr the $h$ with the minimum *error*. $H$ may not coincide with the set of all possbile function and the search cannot be exhaustive, we need to make **assumpitons** (**inductive bias**).

**Learning**   Also called:

Inference, in statistics

Adapting, in biology and systems

Optimizing, in mathematics

Training, in neural networks

Function approximations, in mathematics

. . .

After introducing data, task, model and learning algorithm we will focus on: inductive bias, loss and concepts of generalization and validation.

**Inductive bias**   To set up a model we can make assumptions about the nature of the target function, concerning either:

constraints in the model, **language bias** (in the hypothesis space $H$, due to the set of hypothesis that we can express or consider

constraints or preferences in learning algorithm/search strategy, **search bias** which is preferred

or both

Such assumptions are needed to obtain an useful model for the ML aims, i.e. a model with generalization capabilities. We can imagine learning a discrete function with discrete inputs assuming **conjunctive rules**, so using a **language bias** to work with a restricted hypothesis space.

**Version Space**   An hypothesis $h$ is consistent with the TR if $h(x) = d(x)$ for each training example $\langle x, d(x) \rangle$.
The **version space** $VS_{H,TR}$ is the subset of $H$ of the hypothesis consistent with all the training examples in the TR.

**Unbiased Learner**   The language bias (ex: using only conjunctive rules, may be too restrictive: if the target concept is not in $H$ it cannot be represented in $H$. We can use an $H$ that expresses every teachable concept (among propositions), that means that $H$ is the set of all possible subsets of $X$: the power set $P(X)$. If $n = 10$ binary inputs, then $|X| = 2^{10} = 1024$ and $|P(X)| = 2^{1024} = 10^{308}$ possible concepts, which is much more than the number of the atoms in the universe.
An unbiased learner is unable to generalize: the only examples that are unambiguously classified by an unbiased learner represented with the VS are the training examples themselves. Each unobserved instance will be classified positively by exactly half of the hypothesis in the VS and negative by the other half. Indeed: $\forall\, h$ consistent with $x_i$, $\exists\, h'$ identical to $h$ except $h'(x_i) \neq h(x_i)$, $h \in \mathrm{VS} \Rightarrow h' \in \mathrm{VS}$ (because they are identical on the TR)

Why prefer the search bias? In ML we use flexible approaches (expressive hypothesis spaces with universal capability of the models, for example neural networks or decision trees. We avoid the language bias, so we do not exclude a priori the unknown target function, but we focus on the search bias (ruled by the learning algorithm).

**Loss**   How to measure the quality of an approximation? We want to measure the distance between $h(x)$ and $d$, using a loss function/measure $L(h(x), d)$ for a pattern $x$ which has high value in cases of bad approximation. The error (or risk or loss) is an expected value of this $L$, for example $E(w) = \frac{1}{l} \sum_{p=1}^{l} L(h(x_p), d_p)$. Different $L$ for different tasks. Examples of loss functions:

Regression: $L(h(x_p), d_p) = (d_p - h(x_p))^2$, the squared error. MSE (mean squared error) over the data set

Classification: $L(h(x_p), d_p) = \begin{cases} 0 & h(x_p) = d_p \\ 1 & else \end{cases}$