What is MACHINE LEARNING?

Prof. Dan A. Simovici

UMB

What is Machine Learning?

Machine learning (ML) studies the construction and analysis of algorithms that learn from data.

- ML algorithms construct models starting from samples of data and use these models to make predictions or decisions.
- ML and its applied counterpart, data mining, mostly deal with problems that present difficulties in formulating algorithms that can be readily translated into programs, due to their complexity.
- ML techniques tend to avoid the difficulties of standard problem solving techniques where a complete understanding of data is required at the beginning of the problem solving process.

Typical ML Activities

Example

- finding diagnosis for patients starting with a series of their symptoms;
- determining credit worthiness of customers based on their demographics and credit history;
- document classification based on their main topic;
- speech recognition;
- computational biology applications.

Supervised Learning

Often ML aims to compute a label for each analyzed piece of data that depends of the characteristics of data.
The general approach known as supervised learning is to begin with a number of labelled examples (where answers are known or are provided by a supervisor) known as training set.
The goal is to generate an algorithm that computes the function that gives the the labels of remaining examples.

Unsupervised Learning

In unsupervised learning the challenge is to identify structure that is hidden in data, e.g. identifying groups of data such that strong similarity exists between objects that belong to the same group and also, that objects that belong to different groups are sufficiently distinct.
This activity is known as clustering and it is a typical example of unsupervised learning.
The term "unsupervised" refers to the fact that this type of learning does not require operator intervention. Other machine learning activities of this type include outlier identification, density estimation, etc.

Semi-supervised Learning

An intermediate type of activity, referred as semi-supervised learning requires a limited involvement of the operator.
For example, in the case of clustering, this may allow the operator to specify pairs of objects that must belong to the same group and pairs of objects that may not belong to the same group.

Quality of the Learning Process

The quality of the learning process is assessed through its capability for *generalization*, that is, the capacity of the produced algorithm for computing correct labels for yet unseen examples.

- the correct behavior of an algorithm relative to the training data is no guarantee, in general, for its generalization prowess;
- sometimes in the pursuit of a perfect fit of the learning algorithm to the training data leads to *overfitting*; this term describes the situation when the algorithm acts correctly on the training data but is unable to predict unseen data;
- in an extreme case, a *rote learner* will memorize the labels of its training data and nothing else. Such a learner will be perfectly accurate on its training data but lack completely any generalization capability.

Active and Reinforcement Learning

- A machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns, that is, to apply active learning.
  An active learner may pose queries soliciting a human operator to label a data instance. Since unlabelled data is abundant and, in many cases, easily obtained there are good reasons to use this learning paradigm.
- Reinforcement learning is a machine-learning paradigm inspired by psychology which emphasizes learning by an agent from its direct interaction with the data in order to attain certain goals of learning e.g. accuracy of label prediction.
  The framework of this type of learning makes use of states and actions of an agent, and the rewards and deals with uncertainty and nondeterminism.

The Learner's Input

- **The domain set** $\mathcal{X}$ consists of the objects that we wish to label; usually objects are represented by a vector of features. We refer to these objects as instances.
- **The label set** $\mathcal{Y}$ is generally a finite set, e.g. $\{0, 1\}$ or $\{-1, 1\}$.
- **Training data** $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ is a finite sequence of pairs in $\mathcal{X} \times \mathcal{Y}$, that is, a sequence of labelled objects. Each pair (x_i, y_i) is a training example.

The Learner's Output

The learner is required to produce a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ starting from

$$f(x_1) = y_1, f(x_2) = y_2, \ldots, f(x_n) = y_n,$$

as provided by the training data S.
This function is known as a

- a predictor, or
- a hypothesis, or
- a classifier

The hypothesis provided by the learning algorithm $\mathcal{A}$ starting from S is denoted as $f = \mathcal{A}(S)$.

A Data Generation Model

Assumptions:

- data has a probability distribution function $\mathcal{D}$;
- the learner ignores the probability distribution function $\mathcal{D}$;
- there exists some correct labelling function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ which we seek to determine knowing that $f(x_i) = y_i$ for $1 \leqslant i \leqslant n$.

Measures of Success

Definition

The error of a prediction rule $h : \mathcal{X} \longrightarrow \mathcal{Y}$ is

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

The error is measured with respect to probability distribution $\mathcal{D}$ and the correct labelling function f.
Term used for $L_{(\mathcal{D},f)}(h)$:

- generalization error;
- risk;
- the true error of h.

The letter L suggest that L measures the loss to the learner.

The Training Error

The true error of h is not known to the learner because $\mathcal{D}$ and f are unknown.
Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a sample. The training error of a predictor h on S is the number

$$L_S(h) = \frac{|\{i \mid 1 \leqslant i \leqslant m, h(x_i) \neq y_i\}|}{m}.$$

Alternative terminology:

- empirical error;
- empirical risk.

Empirical Risk Minimization (ERM)

ERM is an approach that seeks a predictor that minimizes $L_S(h)$.
Let h be the predictor defined as $h(x_i) = y_i$ for all $x_i \in S$ and $h(x_i) = k$, where k does not label any object in S. The empirical error will be 0 but h will fail miserably on unseen data. This phenomenon is called overfitting: designing a predictor to fit the sample.

Inductive Bias

ERM can lead to overfitting. Therefore, we seek supplementary conditions that ensure that ERM will not overfit (conditions under which a predictor with good performance on the training data will have good performance on unseen data).
Common solution:

> *Use a restricted hypothesis class $\mathcal{H}$ chosen in advance, that is before seeing the data.*

This approach is known as the inductive bias.

- For a given class $\mathcal{H}$ (known as hypothesis class) and a training sample S the hypothesis

$$h = \text{ERM}_{\mathcal{H}}(S)$$

  uses the ERM rule to chose a predictor $h \in \mathcal{H}$ with the lowest possible error over S.
- Both large $L_S(h)$ values and strong inductive bias are negative; the question is achieve a balance between these factors.
- Let $\text{argmin}_{h \in \mathcal{H}} L_S(h)$ be the set of hypothesis in $\mathcal{H}$ that achieve the minimum values of $L_S(h)$. This approach aims to have $\text{ERM}_{\mathcal{H}}(S) \in \text{argmin}_{h \in \mathcal{H}} L_S(h)$.
- h_S denotes the result of applying $\text{ERM}_{\mathcal{H}}$ to S, namely

$$h_S \in \text{argmin}_{h \in \mathcal{H}} L_S(h).$$

Finite Hypothesis Classes

A simple inductive bias: class $\mathcal{H}$ is finite.
The Realizability Assumption: There exists $h^* \in \mathcal{H}$ such that

$$L_{(\mathcal{D},f)}(h^*) = 0.$$

This implies that with probability 1 over random samples S, where the instances of S are sampled according to $\mathcal{D}$ and are labelled by f we have $L_S(h^*) = 0$.
Realizability assumption implies that for every ERM hypothesis we have $L_S(h_S)$ with probability 1.

- Samples are obtained by drawing values from a distribution $\mathcal{D}$ independently of each other.
- Since samples are drawn randomly from $\mathcal{D}$, the risk $L_{(\mathcal{D},f)}(h_S)$ is a random variable.
- We cannot predict with certainty that a sample S will suffice to direct the learner towards a good classifier.

Approximately Correct Predictors

- The probability of getting a non-representative sample is denoted by δ.
- $1 - \delta$ is the confidence parameter.
- The accuracy parameter ϵ: the event $L_{(\mathcal{D},f)}(h) > \epsilon$ is a failure of the learner.
  If $L_{(\mathcal{D},f)}(h) \leqslant \epsilon$ then the output of the learner is an approximately correct predictor.

Fix f and seek an upper bound for the probability of sampling m instances that will lead to a failure of the learner.
Let $S_x = (x_1, \ldots, x_m)$. We would like to upper bound $D^m(\{S_x \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\})$.

- The set $\mathcal{H}_b$ of bad hypothesis is

$$\mathcal{H}_b = \{h \in H \mid L_{(\mathcal{D},f)}(h) > \epsilon\}.$$

- Define the set of misleading examples:

$$M = \{S_x \mid \exists h \in \mathcal{H}_b, L_S(h) = 0\}.$$

  Namely, for every $S_x \in M$ there exists a bad hypothesis $h \in \mathcal{H}_b$ that looks like a good hypothesis on S_x.

- The realizability assumption implies $L_S(h_S) = 0$. Therefore, the event $L_{(\mathcal{D},f)}(h_S) > \epsilon$ can happen only if for some $h \in \mathcal{H}_b$ we have $L_S(h) = 0$, that is only if $\{S_x \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subseteq M$.
- M can be written as:

$$M = \bigcup_{h \in \mathcal{H}_b} \{S_x \mid L_S(h) = 0\},$$

hence

$$\mathcal{D}^m(\{S_x \mid L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \leqslant \mathcal{D}^m(M) = \mathcal{D}^m(\bigcup_{h \in \mathcal{H}_b} \{S_x \mid L_S(h) = 0\}.$$

Elementary probability theory implies

$$\mathcal{D}^m(\bigcup_{h \in \mathcal{H}_b} \{S_x \mid L_S(h) = 0\} \leqslant \sum_{h \in \mathcal{H}_b} \mathcal{D}^m(\{S_x \mid L_S(h) = 0\}).$$

Fix some bad hypothesis $h \in \mathcal{H}_b$. The event $L_S(h) = 0$ is equivalent to $h(x_i) = f(x_i)$ for $1 \leqslant i \leqslant m$. Since the examples in the training sets are sampled independently and identically distributed (iid) we get

$$\begin{aligned} \mathcal{D}^m(\{S_x \mid L_S(h) = 0\}) &= \mathcal{D}^m(\{S_x \mid h(x_i) = f(x_i) \text{ for } 1 \leqslant i \leqslant m\}) \\ &= \prod_{i=1}^{m} \mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}). \end{aligned}$$

For each individual sampling we have

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

where the last inequality follows from the fact that $h \in \mathcal{H}_b$.
Note that $1 - \epsilon \leqslant e^{-\epsilon}$.
Thus,

$$\mathcal{D}^m(\{S_x \mid L_S(h) = 0\}) \leqslant (1 - \epsilon)^m \leqslant e^{-\epsilon m}.$$

Since

$$\mathcal{D}^m(\bigcup_{h\in\mathcal{H}_b}\{S_x \mid L_S(h)=0\} \leqslant \sum_{h\in\mathcal{H}_b}\mathcal{D}^m(\{S_x \mid L_S(h)=0\})$$

we conclude that

$$\mathcal{D}^m(\bigcup_{h\in\mathcal{H}_b}\{S_x \mid L_S(h)=0\} \leqslant |\mathcal{H}_b|e^{-\epsilon m}.$$

Theorem

Let $\mathcal{H}$ be a finite hypothesis class, $\delta \in (0,1)$, $\epsilon > 0$ and let m be an integer such that

$$m \geqslant \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$$

Then, for any labelling function f and for any distribution $\mathcal{D}$ for which the realizability distribution holds (that is, for some $h \in \mathcal{H}$, $L_{(\mathcal{D},f)}(f) = 0$), with probability at least $1-\delta$ over the choice of an iid sample of size m, we have that for every ERM hypothesis h_S it holds that $L_{(\mathcal{D},f)}(h) \leqslant \epsilon$.