Lecture 2:
PAC learning

Machine Learning 2025

Federico Chiariotti (federico.chiariotti@unipd.it)

Università
degli Studi
di Padova

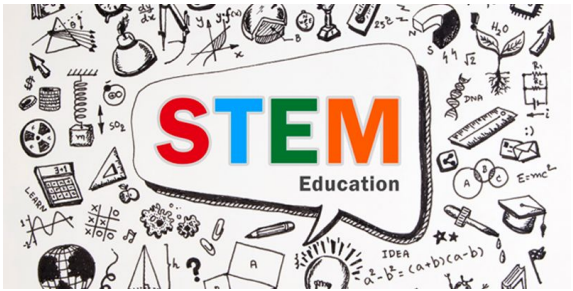
Recap

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Elearning

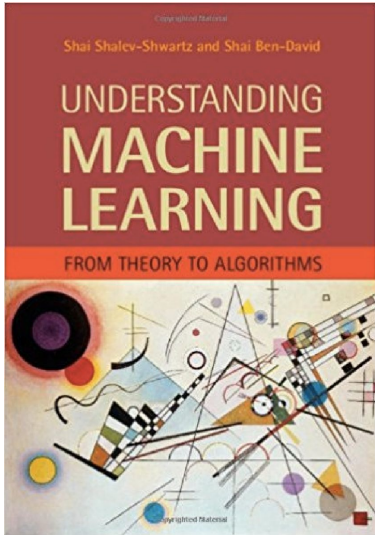
ALL course material and announcements will be published through the course Elearning page. The password for manual enrollment is **ML2526_pw** – but most of you should be enrolled automatically.

https://stem.elearning.unipd.it

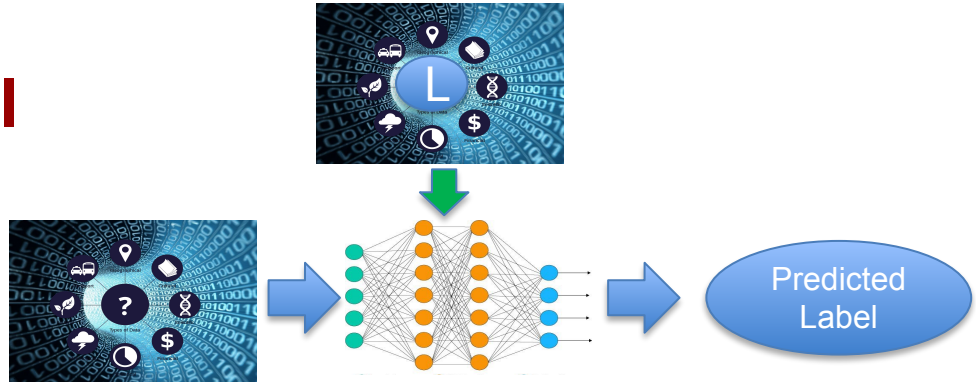


Course material

- Main book: Shalev-Shwartz, Shai, & Ben-David, Shai, *Understanding machine learning: From theory to algorithms*, Cambridge University Press, 2014
- PDF available from the authors at http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html
- Lecture slides: available on Elearning
- Lab solutions and notebooks: available on Elearning

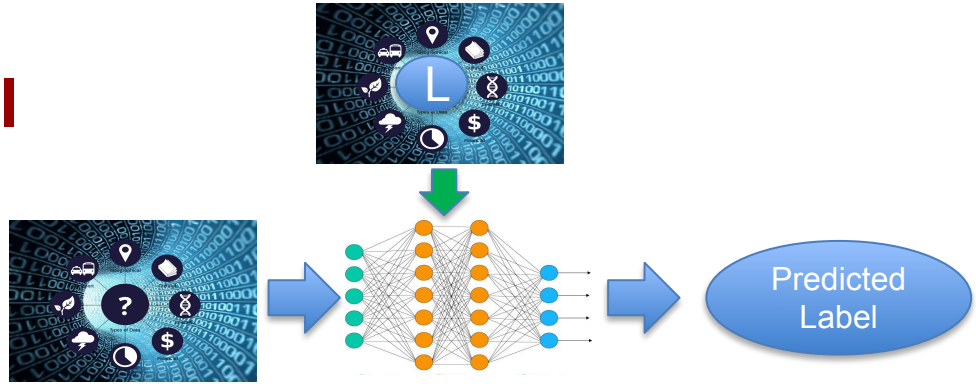


Supervised learning model



- Domain set (or instance space) X
  - $x \in X$ is a domain point or instance
  - Usually, x is represented by a tensor of *features*
- Label set Y
  - Simplest case: binary classification, $Y = \{0, 1\}$
- Training set S
  - Finite sequence of labeled points $((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$
  - Each point x_i is associated to its label y_i

Supervised learning model



- Prediction rule $h : X \rightarrow Y$
  - The learner's output (hypothesis)
  - $A(S)$: hypothesis produced by algorithm A when it is fed training set S
- Data generation model
  - D is a distribution over X **(unknown to the machine learning algorithm)**
  - Instances are labeled according to $f : X \rightarrow Y$ **(unknown to the ML algorithm)**
  - Training set: sampling according to $D, y_i = f(x_i)\ \forall x_i \in S$
- Success metric
  - Classifier error: probability of predicting the wrong label over D

Loss functions

- ➔ Assume a domain subset $A \subset X$
- ➔ A is an event expressed by $\pi : X \to \{0, 1\}$, i.e., $A = \{x \in X : \pi(x) = 1\}$
- ➔ We get $P_{x \sim D}[\pi(x) = 1] = D(A)$

Error of prediction rule $h : X \to Y$

$$L_{D,f}(h) \stackrel{\text{def}}{=} P_{x \sim D}[h(x) \neq f(x)] = D(x : h(x) \neq f(x))$$

$L_{D,f}(h)$ (often written just as $L_D(h)$) is called true loss, true risk, or generalization loss

Empirical Risk Minimization (ERM)

- ➔ Learner outputs $h_S : X \to Y$
- ➔ **Goal: find $\mathbf{h}^*$ that minimizes $\mathbf{L_{D,f}(h)}$**
- ➔ Both D and f are unknown!

ERM: we minimize the loss on the training set $L_S(h) = \frac{\sum_{i=1}^{m} |h(x_i) - y_i|}{m}$

This works if we use a 0-1 loss: the definition can be generalized

The empirical risk is also called training error or training loss

Hypothesis classes

We can apply ERM over a *restricted* class H of possible hypotheses

$h \in H$ is a function $h : X \to Y$

Restricted ERM finds $h^* \in \arg\min_{h \in H} L_S(h)$

There might be multiple solutions

How can we choose a class that avoids overfitting?

Assumptions

1. Let us start with a *finite* class H
2. Let $h_S = \mathrm{ERM}_H \in \arg\min_{h \in H} L_S(h)$

We need two further assumptions:

3. We are lucky, and have *realizability*: $f \in H$, i.e., $\exists h^* \in H : L_{D,f}(h^*) = 0$
4. IID assumption: samples are drawn independently from D

Can we learn h^* with ERM?

Part 1:
PAC learning

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

PAC learning

PAC stands for **probably** **approximately** **correct**

1. **Accuracy**: we are happy with a loss $L_{D,f}(h_S) < \varepsilon$
2. **Confidence**: we can never fully rule out unlucky training sets, but we can limit their probability to a given δ

PAC learning theorem

Let H be a **finite** hypothesis class. Let $\delta, \varepsilon \in (0,1)$ and $m \in \mathbb{N}$ such that:

$$m \geq \frac{1}{\varepsilon} \log\left(\frac{|H|}{\delta}\right)$$

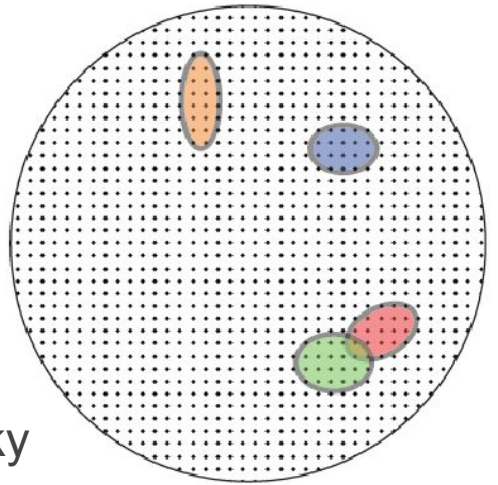
- Inversely proportional to ε
- Logarithmic growth with the size of H and $1/\delta$

For any D and f for which **realizability holds**, we have that any h_S computed with ERM over training set S of size m respects $L_{D,f}(h_S) \leq \varepsilon$ with probability greater than $1 - \delta$

Graphical idea

Let us define the set of *bad hypotheses* $H_B = \{h \in H : L_{D,f}(h) > \varepsilon\}$

We can consider each possible training set S as a point in the circle. The colored areas are sets that lead to $h_S \in H_B$



➔ Increasing m leads to a wider circle (fewer unlucky training sets

➔ Increasing ε narrows down H_B

The theorem places a bound on the probability of being unlucky

Proof

Let us define the set of training sets in which a bad hypothesis works

$$M = \{S \in X^m : \exists h \in H_B, L_S(h) = 0\}$$

We know that, in order for ERM to fail, we must be in M (remember, ERM selects one of the hypotheses with zero training loss):

$$p(L_{D,f}(h_S) > \varepsilon) \leq D^m(M)$$

IID hypothesis

The union bound

Let us consider two event sets A and B and distribution D. We have

$$D(A \cup B) \leq D(A) + D(B)$$

This is easy to prove: the union operation includes the intersection only once, so if an outcome is in both A and B it gets counted twice on the right side

First step

We can rewrite set M as

$$M = \bigcup_{h \in H_B} \{S \in X^m : L_S(h) = 0\}$$

We can apply the union bound:

$$p(L_{D,f}(h_S) > \varepsilon) \leq D^m(M) \leq \sum_{h \in H_B} D^m(\{S : L_S(h) = 0\})$$

Second step

Saying that a hypothesis has zero training loss is equivalent to saying it correctly classifies all points in the training set:

$$D^m(\{S : L_S(h) = 0\}) = \prod_{i=1}^{m} D(\{x_i : h(x_i) = f(x_i)\})$$

By the definition of a bad hypothesis, we have

$$D(\{x : h(x) = f(x)\}) < 1 - \varepsilon$$

As the sample is IID, we know that

$$D^m(\{S : L_S(h) = 0\}) < (1 - \varepsilon)^m$$

Third step

As the sample is IID, we know that

$$D^m(\{S : L_S(h) = 0\}) < (1 - \varepsilon)^m$$

We can use inequality $(1 - x) \leq e^{-x}$ to get

$$D^m(\{S : L_S(h) = 0\}) < e^{-\varepsilon m}$$

Since we need to sum over H_B, we get

$$D^m(M) < |H_B|e^{-\varepsilon m} < |H|e^{-\varepsilon m}$$

Realizability: at least one good hypothesis

Putting it all together

We just proved that

$$p(L_{D,f}(h_S) > \varepsilon) \leq D^m(M) < |H|e^{-\varepsilon m}$$

We can then guarantee condition $p(L_{D,f}(h_S) > \varepsilon) < \delta$ if

$$\delta < |H|e^{-\varepsilon m}$$

This is equivalent to the theorem statement

$$m \geq \frac{1}{\varepsilon} \log \left(\frac{|H|}{\delta} \right)$$

PAC learnability

A hypothesis class H is PAC learnable if there is a function $m_H : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0,1)$ and for every learning problem D, f with the realizability assumption, the algorithm will satisfy the PAC condition over an IID training set of size $m \geq m_H$

Corollary

Any finite H is PAC learnable, as we just proved the existence of a function that satisfies the requirement:

$$m \geq \frac{1}{\varepsilon} \log \left(\frac{|H|}{\delta} \right)$$

However, this is a sufficient, not a necessary condition: some infinite classes might still be PAC learnable!

Remarks

- ➔ PAC learnability is a property of the hypothesis class
- ➔ It must hold for all functions and distributions
- ➔ It requires the realizability assumption (strong!)

If we follow the finite hypothesis bound, we see that larger classes require larger training sets, as does restricting the conditions to be more accurate or more confident

Part 2:
Generalizing our results

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Adding noise

In real life, labels are not always perfectly determined from the inputs

We can replace f by considering a joint distribution D over $X \times Y$

The problem is to learn the marginal distribution $D(y|x)$

In this case, we do not have realizability (if the label is random, we cannot get 0 loss)

Bayesian prediction

If we know D, we can directly identify the optimal predictor by applying Bayes' rule:

$$f_D(x) = \begin{cases} 0, & \text{if } P(y=1|x) < 0.5; \\ 1, & \text{otherwise.} \end{cases}$$

This predictor is not achievable in practice (it would require knowing the solution in advance), but it represents an upper bound to learning performance

Agnostic PAC learnability

Since we dropped the realizability assumption, we cannot reach 0 loss. What we can try and guarantee is a bound on the loss with respect to the minimum possible loss in the hypothesis class

A hypothesis class H is **agnostic** PAC learnable if there is a function $m_H : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0,1)$ and for every learning problem D, the algorithm will satisfy the following condition over an IID training set of size $m \geq m_H$:

$$L_D(h) \leq \min_{h^* \in H} L_D(h^*) + \varepsilon$$

A more general learning problem

1. Binary classification: $Y = \{0, 1\}$

   Is it a goalkeeper or a field player?

2. Multiclass classification: $Y = \{0, 1, \ldots, K-1\}$

   Is it a keeper, defender, midfielder, or striker?

3. Regression: $Y \subseteq \mathbb{R}$

   The target set is uncountably infinite

Image by Steffen Prößdorf, CC BY-SA 4.0

How can we adapt our loss function?

Generalizing the loss function

- ➔ Hypothesis class H
- ➔ Domain $Z = X \times Y$

Any function $\ell : H \times Z \to \mathbb{R}^+$ can be a valid loss function

Risk function: expected loss of a hypothesis $h \in H$

$$L_D(h) = \mathbb{E}_{(x,y)\sim D}[\ell(h,(x,y))]$$

Empirical risk: average loss over the training set S

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h,(x_i,y_i))$$

Common loss functions

- ➔ 0-1 loss: commonly used in classification

$$\ell_{0-1}(h,(x,y)) \triangleq I(h(x) \neq y)$$

- ➔ Cross-entropy: often used in deep learning classification (we will study it later in the course)
- ➔ L2 (squared) loss:

$$\ell_{\text{sq}}(h,(x,y)) \triangleq (h(x)-y)^2$$

- ➔ L1 (linear) loss:

$$\ell_{\text{lin}}(h,(x,y)) \triangleq |h(x)-y|$$

Loss functions are application-dependent, but computational cost also plays a part

Agnostic PAC learnability (generalized loss)

A hypothesis class H is **agnostic** PAC learnable if there is a function $m_H : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0,1)$ and for every learning problem D, the algorithm will satisfy the following condition over an IID training set of size $m \geq m_H$:

$$L_D(h) \leq \min_{h^* \in H} L_D(h^*) + \varepsilon$$

$$L_D(h) = \mathbb{E}_{(x,y)\sim D}[\ell(h,(x,y))]$$

Finding the right loss function

The same application (e.g., fingerprint-based identification) can have different loss functions depending on the context:

- ➔ If the fingerprint is used to let subscribers into a gym, the worst that can happen is that a person trains without paying (and customers at a high-end gym do not want the hassle of scanning multiple times or getting locked out)
- ➔ If the fingerprint is used to access dangerous chemicals, the consequences for a false positive are very serious