Lecture 3:
Learning via uniform convergence

Machine Learning 2025

Federico Chiariotti (federico.chiariotti@unipd.it)

Università degli Studi di Padova

Lecture plan

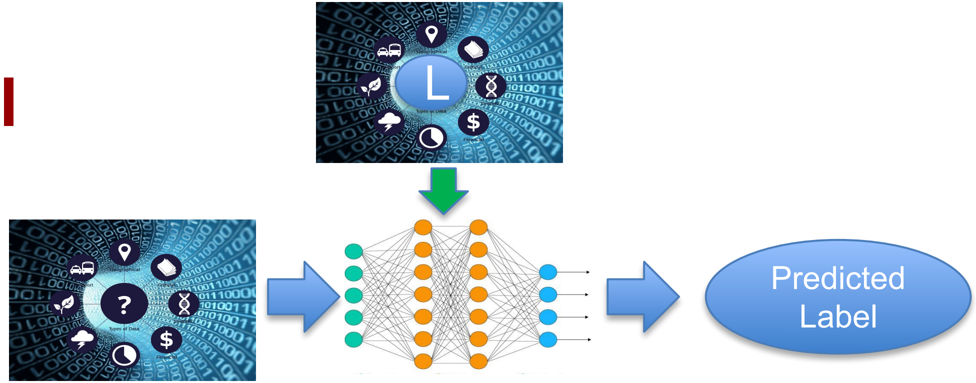
Date	#	Topic	Date	#	Topic	Date	#	Topic
Sep. 30	1	Introduction	Nov. 4	L2	*Model selection*	Nov. 28	12	CNNs
Oct. 7	2	PAC learning	Nov. 7	8	SVMs	???	13	PCA
Oct. 10	3	Uniform convergence	Nov. 11	L3	*Linear models*	Dec. 5	14	Clustering models
Oct. 14	L1	*Python basics*	???	9	Kernels	Dec. 9	L6	*Neural networks*
Oct. 17	4	VC dimension	Nov. 14	10	Ensemble models	Dec. 16	L7	*Clustering*
Oct. 21	5	Model selection	Nov. 18	L4	*SVMs*	Dec. 19	15	Reinforcement
Oct. 24	6	Linear classification	Nov. 21	11	Neural networks	???	L8	*Reinforcement*
Oct. 31	7	Linear regression	Nov. 25	L5	*Random forests*	???	16	Exercises and Q&A

IMPORTANT: no lecture on October 28!

Recap

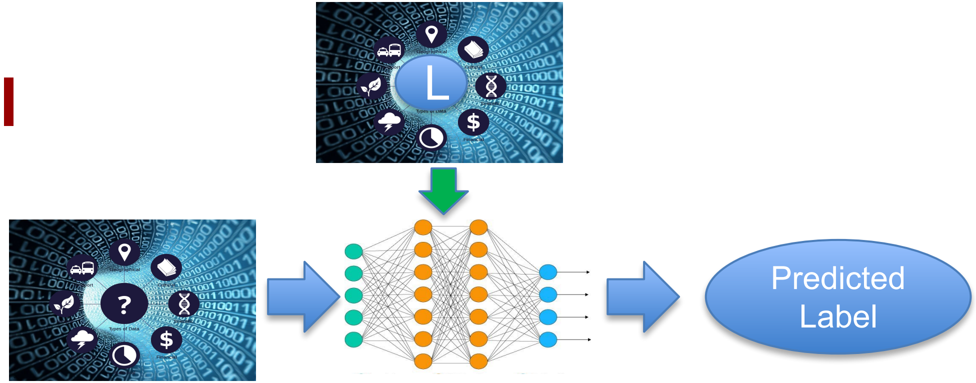
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

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 $((x_1, y_1), (x_2, y_2), \dots, (x_m, y_m))$
  - Each point x_i is associated to its label y_i

Supervised learning model



- Prediction rule $h : X \to Y$
  - 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 $f : X \to Y$ **(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

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 $1-\delta$

PAC learnability

A hypothesis class H is PAC learnable if there is a function $m_H : (0,1)^2 \rightarrow \mathbb{N}$ 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 $m \geq m_H$

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 $m \geq m_H$:

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

Part 1:
Uniform convergence

Università
degli Studi
di Padova

Empirical risk as an approximation

An ERM learning algorithm takes a training set S as input and selects a hypothesis $h_S \in H$ with the lowest possible empirical error

Why do we do this?

➔ We see the training set as a representative sample, so we consider the empirical risk as a fair approximation of the true risk

Empirical risk as an approximation

If the empirical risk is a good approximation of the true risk for all hypotheses, learning will generalize:

$$L_S(h) \simeq L_D(h) \forall h \in H \implies \text{PAC}$$

This is a relatively restrictive condition: it is **sufficient** for learning, but not **necessary**

Definition: ε-representative sets

$$L_S(h) \simeq L_D(h) \forall h \in H \implies \text{PAC}$$

What does it mean for empirical risk to be a good approximation? We need a definition before we can get any results from this intuition

A training set S is ε-representative with respect to domain Z, distribution D, loss function ℓ, and hypothesis class H, if

$$|L_D(h) - L_S(h)| \leq \varepsilon \; \forall h \in H$$

Representativeness lemma

If training set S is ε-representative with respect to domain Z, distribution D, loss function ℓ, and hypothesis class H, any ERM rule satisfies

$$L_D(h_S) \leq \arg\min_{h \in H} L_D(h) + 2\varepsilon$$

If we can prove that this happens with probability $1-\delta$, we have agnostic PAC learnability!

Proof

Let us take the best hypothesis h^* and the ERM hypothesis h_S

Due to ε-representativeness, we know that

$$L_S(h^*) - \varepsilon \leq L_D(h^*) \leq L_S(h^*) + \varepsilon$$

The same holds for h_S

$$L_S(h_S) - \varepsilon \leq L_D(h_S) \leq L_S(h_S) + \varepsilon$$

Proof: first step

Due to the ERM rule, we know that

$$L_S(h_S) \leq L_S(h^*)$$

Combining this with the consequences of ε-representativeness,

$$L_S(h_S) - \varepsilon \leq L_D(h_S) \leq L_S(h_S) + \varepsilon$$

We get

$$L_D(h_S) - \varepsilon \leq L_S(h^*)$$

Proof: second step

Due to the consequences of ε-representativeness,

$$L_S(h^*) - \varepsilon \leq L_D(h^*) \leq L_S(h^*) + \varepsilon$$

In the first step, we had

$$L_D(h_S) - \varepsilon \leq L_S(h^*)$$

We can substitute the term on the right and get

$$L_D(h_S) - \varepsilon \leq L_D(h^*) + \varepsilon$$

Uniform convergence and PAC learnability

A hypothesis class H has the uniform convergence property with respect to a domain Z and a loss function ℓ if there exists a function $m_H^{\mathrm{UC}} : (0,1)^2 \to \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$ and every distribution D, any set S of size $m \geq m_H^{\mathrm{UC}}(\varepsilon, \delta)$ is ε-representative with probability higher than $1-\delta$

If a class H has the uniform convergence property with function m_H^{UC}, we have:

1. The class is agnostically PAC learnable with sample complexity
$$m_H(2\varepsilon, \delta) \leq m_H^{\mathrm{UC}}(\varepsilon, \delta)$$
2. ERM is a valid PAC learning algorithm for H

Proof

1. By the definition of uniform convergence, a training set of size $m_H^{\mathrm{UC}}(\varepsilon, \delta)$ is ε-representative with probability higher than $1 - \delta$
2. The representativeness lemma tells us that
$$L_D(h_S) \leq \arg\min_{h \in H} L_D(h) + 2\varepsilon$$
3. Combining the two, we have the definition of agnostic PAC

Finite hypothesis classes are agnostic PAC learnable

Let H be a finite hypothesis class, let Z be a domain and let $\ell : H \times Z \to \mathbb{R}^+$ be a loss function.

1. H enjoys the uniform convergence property, with sample complexity

$$m_H^{\mathrm{UC}}(\varepsilon, \delta) \leq \left\lceil \frac{\log\left(\frac{2|H|}{\delta}\right)}{2\varepsilon^2} \right\rceil$$

2. H is agnostic PAC learnable using ERM, with sample complexity

$$m_H(\varepsilon, \delta) \leq m_H^{\mathrm{UC}}(\varepsilon, \delta) \leq \left\lceil \frac{2\log\left(\frac{2|H|}{\delta}\right)}{\varepsilon^2} \right\rceil$$

Proof: first step

The definition of uniform convergence tells us that

$$D^m(\{S : \exists h \in H : |L_S(h) - L_D(h)| > \varepsilon\}) \leq \delta$$

This can be rewritten as the union over all hypotheses

$$D^m\left(\bigcup_{h \in H} \{S : |L_S(h) - L_D(h)| > \varepsilon\}\right) \leq \delta$$

We can apply the union bound:

$$D^m(\{S : \exists h \in H : |L_S(h) - L_D(h)| > \varepsilon\}) \leq \sum_{h \in H} D^m(\{S : |L_S(h) - L_D(h)| > \varepsilon\})$$

Hoeffding's inequality

Let $\theta_1, \ldots, \theta_m$ be a sequence of IID random variables, with $\mathbb{E}[\theta_i] = \mu$ and $P[a \leq \theta_i \leq b] = 1$. For any $\varepsilon > 0$, we have

$$P\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \varepsilon\right] \leq 2e^{\frac{-2m\varepsilon^2}{(b-a)^2}}$$

Proof: second step

We can consider Hoeffding's inequality, with $\theta_i = \ell(h, z_i)$. We have $\mu = L_D(h)$

Let us assume that $\ell(h, z) \in [0, 1]$, so $[a, b] = [0, 1]$

$$P\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \varepsilon\right] \leq 2e^{\frac{-2m\varepsilon^2}{(b-a)^2}} \Longrightarrow P\left[\left|\frac{1}{m}\sum_{i=1}^{m}\ell(h, z_i) - L_D(h)\right| > \varepsilon\right] \leq 2e^{-2m\varepsilon^2}$$

Proof: third step

$$P\left[\left|\frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i)-L_D(h)\right|>\varepsilon\right]\leq 2e^{-2m\varepsilon^2}$$

Let us take the constant term out of the sum:

$$P\left[\left|\left(\frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i)\right)-L_D(h)\right|>\varepsilon\right]\leq 2e^{-2m\varepsilon^2}$$

We can see that the sum is simply the empirical risk:

$$D^m(S:|L_S(h)-L_D(h)|>\varepsilon)\leq 2e^{-2m\varepsilon^2}$$

Proof: fourth step

$$D^m(S : |L_S(h) - L_D(h)| > \varepsilon) \leq 2e^{-2m\varepsilon^2}$$

Let us then go back to our union bound:

$$D^m(\{S : \exists h \in H : |L_S(h) - L_D(h)| > \varepsilon\}) \leq \sum_{h \in H} D^m(\{S : |L_S(h) - L_D(h)| > \varepsilon\})$$

If we substitute the bound, we get:

$$D^m(\{S : \exists h \in H : |L_S(h) - L_D(h)| > \varepsilon\}) \leq 2|H|e^{-2m\varepsilon^2}$$

Proof: fifth step

We need to impose an unlucky training set probability δ using our bound:

$$D^m(\{S : \exists h \in H : |L_S(h) - L_D(h)| > \varepsilon\}) \leq 2|H|e^{-2m\varepsilon^2} \leq \delta$$

We can try to isolate the training set size:

$$e^{-2m\varepsilon^2} \leq \frac{\delta}{2|H|}$$

Taking the logarithm on both sides,

$$-2m\varepsilon^2 \leq \log\left(\frac{\delta}{2|H|}\right)$$

Proof: sixth step

$$-2m\varepsilon^2 \leq \log\left(\frac{\delta}{2|H|}\right)$$

We can change the sign by using the logarithm properties

$$2m\varepsilon^2 \geq \log\left(\frac{2|H|}{\delta}\right)$$

This is equivalent to the m_H^{UC} bound in the theorem

Proof: final step

We proved that we have uniform convergence with

$$m_H^{\mathrm{UC}}(\varepsilon,\delta) \leq \left\lceil \frac{\log\left(\frac{2|H|}{\delta}\right)}{2\varepsilon^2} \right\rceil$$

We proved PAC learnability earlier: we know that $m_H(2\varepsilon,\delta) \leq m_H^{\mathrm{UC}}(\varepsilon,\delta)$, so we simply get the second half of the theorem

$$m_H(\varepsilon,\delta) \leq m_H^{\mathrm{UC}}\left(\frac{\varepsilon}{2},\delta\right) \leq \left\lceil \frac{2\log\left(\frac{2|H|}{\delta}\right)}{\varepsilon^2} \right\rceil$$

Part 2:
No free lunch

Can we create a universal learner?

Ideally, given a training set S and a loss function ℓ, we would like to find a hypothesis $\hat{h}$ with a small $L_D(\hat{h})$

Learning depends on the hypothesis class H and on the algorithm A

- ➔ Can we build a **universal learner**, i.e., an algorithm A that finds $\hat{h}$ for any distribution D?
- ➔ What if we use the set of all functions as a hypothesis class?

No free lunch theorem

Let A be any learning algorithm for binary classification, with 0-1 loss, over a domain X. Let m be any number smaller than $\frac{|X|}{2}$. There exists a distribution D over $X \times \{0,1\}$ such that:

1. There exists a function $f : X \to \{0,1\}$ with $L_D(f) = 0$
2. We have $L_D(A(S)) \geq \frac{1}{8}$ with probability $\frac{1}{7}$ over the choice of $S \sim D^m$

Corollary: if H is the set of all functions and X is an infinite set, we do **not** have PAC learnability

Consequences

- ➔ We can design a task to make any ML algorithm fail (even if another algorithm is able to solve it)
- ➔ Idea of the proof *(not part of the course, it's in the book if you're curious)*: since our training set is smaller than half the domain, we have no idea what happens in the other half, so we can design a function that contradicts our predictor on that part
- ➔ We have to use **prior knowledge** to restrict the hypothesis class

Proof of the corollary

Let us proceed by contradiction, and assume that H is PAC learnable

We can set $\varepsilon < \frac{1}{8}$ and $\delta < \frac{1}{7}$

The definition of PAC says that we must have a true risk higher than ε with a probability below δ for any value of ε and δ (as long as the training set is large enough)

However, this is **impossible** for any finite size due to the no free lunch theorem!