AA 2019-2020

Computational Learning Theory

Machine Learning

Daniele Loiacono

POLITECNICO MILANO 1863

Outline and References

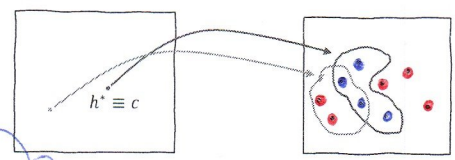
- Outline
  - Basics [ML 7.1,7.2]
  - PAC-Learning [ML 7.3]
  - VC Dimension [ML 7.4]
- References
  - Machine Learning, Mitchell [ML]

What is computational learning theory?

— Something that allows us to bound the test error based on the training error. (obviously under some assumptions)

- It aims at studying the general laws of inductive learning, by modeling:
  - Complexity of hypothesis space
  - Bound on training samples
  - Bound on accuracy
  - Probability of succesfull learning
  - ...
- This allows to answer to questions like:
  - How many training samples do a learner need to **converge** (with some probability) to a **successful** (with some minimum accuracy) hypothesis?
  - How many training samples will be misclassified by the learner before converging to a succesful hypothesis?
  - ...

(Let's Go Back to) The Big Picture



In this case each point of the hypothesis space corresponds to a decision boundary. Some points lead to good decision boundaries, other not.

Hypothesis space (all the points represent a function)

(each point corresponds to a different function)

- A learner (L) wants to learn a *concept* (c) that maps the data in the input space (X) to a target (t)
- Let assume that L found an hypothesis h^* with no errors on the training data
- How many training samples of X are necessary to be sure that L actually learned the true concept, i.e., $h^* \equiv c$?

Supposing that we have no noise, if we don't have any particular assumption and we consider all the possible concepts that we can learn, unless we observe all the points, we cannot tell if we're learning the true concept or not. This is the basis of the "No Free Lunch" theorem.

«No Free Lunch» Theorems

In absence of any assumption about the data, there is no reason to prefer one model over another. There is no apriori system for determining the best model for a specific dataset.

- Let $ACC_G(L)$ be the **generalization accuracy** of learner L, i.e., the accuracy of L on samples that are **not** in the training set
- Let $\mathcal{F}$ be the set of all the possible concepts $y = f(\mathbf{x})$ (concepts = functions)
- For any learner L and any possible training set:

$$\frac{1}{|\mathcal{F}|}\sum_{\mathcal{F}} Acc_G(L) = \frac{1}{2}$$

  - **Proof Sketch**: for every concept f where $ACC_G(L) = 0.5 + \delta$, exists a concept f' where $ACC_G(L) = 0.5 - \delta$: $\forall \mathbf{x} \in \mathcal{D}, f'(\mathbf{x}) = f(\mathbf{x}); \forall x \notin \mathcal{D}, f'(\mathbf{x}) \neq f(\mathbf{x})$
  - **Corollary:** for any two learners, L_1 and L_2, if $\exists f$ where $ACC_G(L_1) > ACC_G(L_2)$ then $\exists f'$ where $ACC_G(L_2) > ACC_G(L_1)$

IF we consider any learner applied on an arbitrary concept, on average w.r.t. all the possible samples the generalized accuracy will be $\frac{1}{2}$. (gen. accuracy = "TRUE" error)

«No Free Lunch» Theorems

- Let $ACC_G(L)$ be the **generalization accuracy** of learner L, i.e., the accuracy of L on samples that are **not** in the training set
- Let
- For

What does this mean in practice?

There is no such thing as a **winner-takes-all** in ML!

In ML we always operate under some assumptions!

- cept f'
- (L_2) then

$\exists f'$ where $ACC_G(L_2) > ACC_G(L_1)$

for example we assume that the training data is distributed like the test data (strong assumpt.!) we need to be aware of the assumptions we're using. We assume that what we see (training set) is representative. A violation of this assumption leads to non-negligible errors. ML is useless if the training set is not representative.

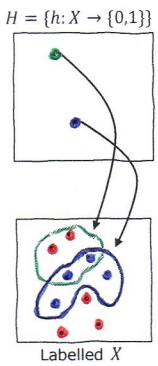
Probably Learning an Approximately Correct Hypothesis

Basics

(for now we focus on binary classification)

- Problem setting
  - Let X be the instance space (input space)
  - Let $H = \{h: X \to \{0,1\}\}$ be the hypothesis space of L
  - Let $C = \{c: X \to \{0,1\}\}$ be the set of all the possible target functions (**concepts**) we might want to learn
  - Let be $\mathcal{D}$ be training data drawn from a **stationary** distribution $P(X)$ and labeled (**without noise**) according to a concept c we want to learn *
- A learner L outputs a hypothesis $h \in H$ such that

$$h^* = \arg\min_{h \in H} error_{train}(h)$$



H: space of all the possible functions that map a point in the input space into a target (all the possible models) that we can produce

C: space of all the possible functions that map a point in the input space into a target THAT WE MAY WANT TO LEARN (≠ from the ones we can produce/obtain (H)) We have no guarantee that our learner L is able to learn any function

Assumptions: (big!) we want to learn something that is correlated with our inputs and it's well represented by our training set. This is formally specified by *

How do we compute the error?

- We compute the error of an hypothesis as the probability of misclassfyng a sample:

$$error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}}[h(x) \neq c(x)] = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} I(h(x) \neq c(x))$$

$\mathcal{D}$ is the training data

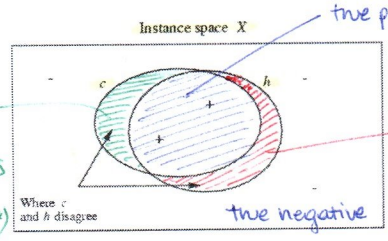
- This is the training error, instead we are interested in the **true error** of h:

$$error_{true}(h) = \Pr_{x \sim P(X)}[h(x) \neq c(x)]$$

$P(X)$ is the input space distribution

Our goal is to find a model that is accurate on points that we never saw

How do we compute the error?



true positive

false negative (our model (h) says they're "−" but they're actually "+")

false positive (our model (h) says they're "+" instead they're "−" (c = true))

(c and h are two models, this is binary classification)

true negative

- But we have to remember that...

$$error_{true}(h) = \Pr_{x \sim P(X)}[h(x) \neq c(x)]$$

What now?

- We say that h overfits the training data if $error_{true} > error_{\mathcal{D}}$ ...
- ... but can we bound $error_{true}$ given $error_{\mathcal{D}}$?
- Let assume...
  - $error_{true}$ is the probability of making a mistake on a sample
  - we can compute $error_{\mathcal{D}}$ that is the average error probability on $\mathcal{D}$
  - assuming a Bernoulli distribution for the error probability, the 95% CI is:

$$error_{true}(h) = error_{\mathcal{D}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{n}}$$

- Is this correct? No! Because $\mathcal{D}$ is the training data and not **independent** of h
- So, **we need to bound the error under more strict assumptions**

It could be cool, but

what we are observing ($error_D$) is the empirical average on the training set D

because of this we're introducing a bias in the estimate of $error_D$ (the probability of making an error on a point of D is not equal to the probability of making an error on a point $\notin$ D)

let's start in a simpler setting:

Version Space

- A hypothesis h is **consistent** with a training dataset $\mathcal{D}$ of the concept c if and only if $h(x) = c(x)$ for each training sample in $\mathcal{D}$

$$Consistent(h, \mathcal{D}) \stackrel{\text{def}}{=} \forall \langle x, c(x) \rangle \in \mathcal{D}, h(x) = c(x)$$

- The version space, $VS_{H,\mathcal{D}}$ with respect to hypothesis space H and labelled dataset $\mathcal{D}$, is the subset of hypotheses in H consistent with $\mathcal{D}$

$$VS_{h,\mathcal{D}} \stackrel{\text{def}}{=} \{h \in H | Consistent(h, \mathcal{D})\}$$

- From now on, we consider only **consistent learners**, that always output a **consistent** hypothesis, i.e., an hypothesis in $VS_{H,\mathcal{D}}$, **assuming it is not empty**
- Can we bound the $error_{true}$ of a consistent learner?

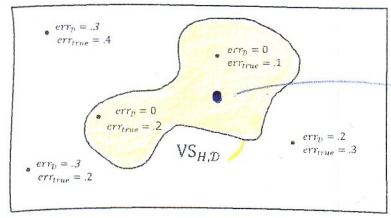
h is consistent with D if the output of the hypothesis ($h(\,)$) is equal to the output of the concept ($c(\cdot)$) $\forall x \in D$.

Basically the version space is the set of all the hypothesis $h \in H$ that have zero training error (no errors on the training set)

Note: even if we establish that $VS_{h,D} \neq \emptyset$, it doesn't mean that our learner will be able to find the optimum (the learner has to solve an optimization problem!)

Version Space (2)

Hypothesis space H



every point here has: $error_D = 0$, $error_{true} = ?$
We want to bound $error_{true}$ for all these points that have $error_D = 0$. Clearly we cannot rely on $error_D$ to bound $error_{true}$.

- If we wish to bound the $error_{true}$ of a consistent learner, we need to find a bound for all the hypotheses in $VS_{H,\mathcal{D}}$

Bound for Consistent Learners

If the hypothesis space H is **finite** and $\mathcal{D}$ is a sequence of $N \geq 1$ independent random examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that $VS_{H,\mathcal{D}}$ contains a hypothesis error greater then ε is less than $|H|e^{-\varepsilon N}$

$$Pr(\exists h \in H : error_{\mathcal{D}}(h) = 0 \wedge error_{true}(h) \geq \varepsilon) \leq |H|e^{-\varepsilon N}$$

we're bounding the probability that the version space contains an hypothesis with an error (true) greater than ε. In this way, if our learner is inside the version space also its error is bounded.

Proof

$$\begin{aligned}
&Pr((error_{\mathcal{D}}(h_1) = 0 \wedge error_{true}(h_1) \geq \varepsilon) \vee \cdots \vee (error_{\mathcal{D}}(h_{|VS_{H,\mathcal{D}}|}) = 0 \wedge error_{true}(h_{|VS_{H,\mathcal{D}}|}) \geq \varepsilon)) \\
&\leq \sum_{h \in VS_{H,\mathcal{D}}} Pr(error_{\mathcal{D}}(h) = 0 \wedge error_{true}(h) \geq \varepsilon) && \text{(Union bound)} \\
&\leq \sum_{h \in VS_{H,\mathcal{D}}} Pr(error_{\mathcal{D}}(h) = 0 | error_{true}(h) \geq \varepsilon) && \text{(Bound using Bayes' rule)} \\
&\leq \sum_{h \in VS_{H,\mathcal{D}}} (1 - \varepsilon)^N && \text{(Bound on individual } h\text{)} \\
&\leq |H|(1 - \varepsilon)^N && (|VS_{H,\mathcal{D}}| \leq |H|) \\
&\leq |H|e^{-\varepsilon N} && (1 - \varepsilon \leq e^{-\varepsilon}, \text{ for } 0 \leq \varepsilon \leq 1)
\end{aligned}$$

the probability of correct classification is 1 - prob. of incorrect classification $\Rightarrow \leq 1 - \varepsilon$ (on D)
We need this to happen to every sample: $\Rightarrow \leq (1-\varepsilon)^N$

What does it mean in practice?

- Let say that δ is the probability to have $error_{true} > \varepsilon$ for a consistent hypotesis:

$$|H|e^{-\varepsilon N} \leq \delta$$

confidence level that we want

- We can bound N after setting ε and δ:

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

- We can bound ε after setting N and δ:

$$\varepsilon \geq \frac{1}{N}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

(*)
$$\mathbb{P}(error_D(h) = 0 \mid error_T(h) \geq \varepsilon) = \prod_{j=1}^{N} \mathbb{P}(error_{x_j}(h) = 0 \mid error_T(h) \geq \varepsilon) = \prod_{j=1}^{N} \left[1 - \mathbb{P}(error_{x_j}(h) \neq 0 \mid error_T(h) \geq \varepsilon)\right]$$

$\mathbb{P}(error_{x_j}(h) \neq 0 \mid error_T(h) \geq \varepsilon) \geq \varepsilon$

$1 - \mathbb{P}(\cdot) \leq 1 - \varepsilon$

$\prod(1 - \mathbb{P}(\cdot)) \leq \prod(1-\varepsilon) = (1-\varepsilon)^N$

What does it mean in practice?

- Let say that δ is the probability to have $error_{true} > \varepsilon$ for a consistent hypotesis:

$$|H|e^{-\varepsilon N} \leq \delta$$

- We can bound N after setting ε and δ:

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

- We can bound ε after setting N and δ:

Can be expeonential in #features

$$\varepsilon \geq \frac{1}{N}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

Example: Conjunction of up to N Boolean Literals

- Consider a classification problem
  - Instance space is $X = < x_1, x_2, x_3, x_4 >$, where x_i is a boolean variable
  - Each hypothesis h is a rule like this:
    $if\ (x_1 = 1, x_2 = ?, x_3 = 0, x_4 = 1)\ then\ y = 1, otherwise\ y = 0$
- How many samples N are necessary to guarantee that, with a probability at least of 0.99, the error of a consistent hypothesis is not greater than 0.05?

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right) \Rightarrow N \geq 180$$

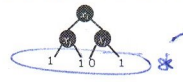
0.05 · 3^4 · 0.01

- How does it scale with respect to the number of variables (M)?
  - M=10 → N≥312 ($|H| = 3^{10}$)
  - M=100 → N≥2290 ($|H| = 3^{100}$)

How big is the hypothesis space? We have 3 options for all the 4 variables ⇒ 3^4
(All the rules are achieved by combining: 0, 1, ? :
x1 = 0, 1, ?
x2 = 0, 1, ?
x3 = 0, 1, ?
x4 = 0, 1, ?)

Example: Decision Tree (depth=2)

- Consider a classification problem
  - Instance space is $X = < x_1, \ldots, x_M >$, where x_i is a boolean variable
  - Each hypothesis h is a rule is a decision trees of depth 2 using only two variables:



for each we have $\frac{M(M-1)}{2}$ possibility of choosing x_j and x_i. Moreover, we have the possibility of * ⇒ $2^4 = 16$

- How many samples N are necessary to guarantee that, with a probability at least of 0.99, the error of a consistent hypothesis is not greater than 0.05?

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

0.05 · $\frac{M(M-1)}{2}16$ · 0.01

Probably Learning an Approximately Correct Hypothesis

- Considering a class C of possible target concepts defined over an instance space X with an encoding lenght M, and a learner L using an hypotesis space H we define:

> C is PAC-learnable by L using H if for all $c \in C$, for any distribution $P(X)$, ε (such that $0 < \varepsilon < 1/2$), and δ (such that $0 < \delta < 1/2$), learner L will with a probability at least $(1-\delta)$ output a hypotehsis $h \in H$ such that $error_{true}(h) \leq \varepsilon$, in time that is polynomial in $1/\varepsilon$, $1/\delta$, M, and $size(c)$.

- So, PAC-learnability is only about computational complexity? What about the complexity with respect to the number of training samples N?
- **A sufficient condition to prove PAC-learnability is proving that a learner L requires only a polynomial number of training examples, and processing per example is polynomial**

Agnostic Learning

- So far, we assumed that $c \in H$, or at least that $VS_{H,\mathcal{D}}$ is not empty, and the learner L will always output a hypothesis h such that $error_{\mathcal{D}}(h) = 0$
- But in general (**agnostic**) leaner will output a hypothesis h such that $error_{\mathcal{D}}(h) > 0$
- Can we bound $error_{true}(h)$ given $error_{\mathcal{D}}(h)$?

> If the hypothesis space H is **finite** and $\mathcal{D}$ is a sequence of $N \geq 1$ i.i.d. examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, and for any learned hypothesis h, the probability that $error_{true}(h) - error_{\mathcal{D}}(h) > \varepsilon$ is less than $|H|e^{-2N\varepsilon^2}$
>
> $$Pr(\exists h \in H : error_{true}(h) > error_{\mathcal{D}}(h) + \varepsilon) \leq |H|e^{-2N\varepsilon^2}$$

We try to bound not anymore the true error but the difference between the true error and the target error
⇒ we're trying to compute the if that exists an hypothesis in H that has a true error that is greater than the training error of a given treshold ε

Agnostic Learning - Proof

- **Additive Hoeffding Bound**: let $\hat{\theta}$ be the empirical mean of N i.i.d. Bernoulli random variables with mean θ:

$$Pr(\theta > \hat{\theta} + \varepsilon) \leq e^{-2N\varepsilon^2}$$

$\mathbb{P}(\text{true mean} > \text{empirical mean} + \varepsilon) \leq e^{-2N\varepsilon^2}$

- So for any single hypothesis h:

$$Pr(error_{true}(h) > error_{\mathcal{D}}(h) + \varepsilon) \leq e^{-2N\varepsilon^2}$$

← translated in our case

- As we want this to be true for all the hypothesis in H:

$$Pr(\exists h \in H : error_{true}(h) > error_{\mathcal{D}}(h) + \varepsilon) \leq |H| e^{-2N\varepsilon^2}$$

Bounds for Agnostic Learning

- Similarly to what done before, we can bound the sample complexity:

$$N \geq \frac{1}{2\varepsilon^2}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

- We can also bound the true error of the hypotesis as:

$$error_{true}(h) \leq error_{\mathcal{D}}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2N}}$$

- We found the bias and variance decomposition we previously saw in the course!

we assume no noise ⇒ no inreducible error. The role of the bias is played by $error_{\mathcal{D}}(h)$ and the variance is represented by $\sqrt{\frac{1}{2N}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)}$.

PAC-Learning with Infinite Hypotheses Spaces

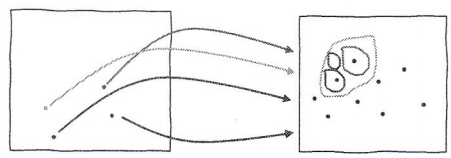
- Previously we found this PAC-Learning bound for the number of samples:

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

- If |H| is infinite, what does this mean? What can we used instead of |H|?
- The answer is the largest subset of X for which |H| can guarantee a zero training error (regardless of the target function c)
- We call **VC dimension** the size of this subset

We see from the variance term that in order to afford a large hypothesis space ($|H|$) we need a large amount of data (N), so that variance doesn't explode. The assumption (strong) that we're making is that $|H| < \infty$. However, in most of the cases the hypothesis space is not finite.

Intuition Behind Using VC Dimension



$H = \{h: X \to t\}$ $\quad X$

We have N samples: how many concepts we can learn? 2^N, because each point can be classified as +1/-1. (or 0/1). Assuming $|H| = 2^N$ = same dimension as the number of possible concepts, we can always find $h \in H$ s.t. $error_{\mathcal{D}}(h) = 0$.

- Let assume that |X|=N, how big is |C|?
- Assuming $|H| = 2^N$, we can always find $h \in H$ with $error_{\mathcal{D}}(h)=0$
- Does $error_{\mathcal{D}}(h)$ tells something more on the error on other samples in X?
- What happens instead if with H we can classify correctly no more than 2 training samples?

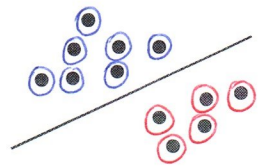
VC Dimension

- We define a **dichotomy** of a set S of instances as a partition of S into two disjoint subsets, i.e., labeling each instance in S as positive or negative
- We say that a set of instances S is **shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy
- The **Vapnik-Chervonenkis dimension**, VC(H), of hypothesis space H over instance space X, is the largest finite subset of X shattered by H

} if and only if any labelling can be learned in **H**

Example: VC dimension of linear classifier

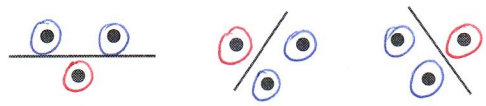
- What about a linear classifier in 2D input space?



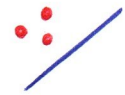
What is the **VC** dimension of this linear classifier?

Example: VC dimension of linear classifier

- What about a linear classifier in 2D input space?



Also extreme case: (3 points equally labelled)

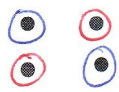


If we consider **3** points we're sure that any choice of labeling (so, any dicotomy) allows to find (through a linear classifier) a consistent hypothesis.
Is this possible with an arbitrary number of points?

Example: VC dimension of linear classifier

- What about a linear classifier in 2D input space?

No!



Cannot be correctly classified!

NOT any subset of **4** points can guarantee a linear model that classifies all the points correctly

Example: VC dimension of linear classifier

- What about a linear classifier in 2D input space?

Cannot be correctly classified!

- A linear classifier in a 2D input space has VC(h)=3
- We can prove that a linear classifier in M-D input space has VC(h)=M+1 !

For this reason, a linear classifier in a 2D space has **VC(h)=3**. If we choose 4 points (any) we won't be able to always classify all of them correctly.

VC Dimension

- We define a **dichotomy** of a set S of instances as a partition of S into two disjoint subsets, i.e., labeling each instance in S as positive or negative
- We say that a set of instances S is **shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy
- The **Vapnik-Chervonenkis dimension**, VC(H), of hypothesis space H over instance space X, is the largest finite subset of X shattered by H
- If an arbitrarily large set of X can be shattered by H, VC(H)=∞
- If $|H| < \infty$ then $VC(H) \leq \log_2(|H|)$
  - If $VC(H) = d$ it means there are in H at least 2^d hypotheses to label d instances
  - Thus, $|H| \geq 2^d$

Sample Complexity based on VC-Dimension

- How many randomly drawn examples suffice to guarantee that any hypothesis that perfectly fits the training data is probably $(1-\delta)$ approximately (ε) correct ?

$$N \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

$$N \geq \frac{1}{\varepsilon}\left(8VC(H)\log_2(13/\varepsilon) + 4\log_2(2/\delta)\right)$$

Agnostic Learning: VC Bounds

- With probability at least $(1-\delta)$ every $h \in H$ satisfies the following inequality:

$$error_{true}(h) \leq error_{\mathcal{D}}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2N}}$$

$$error_{true}(h) \leq error_{\mathcal{D}}(h) + \sqrt{\frac{VC(H)(\ln\frac{2N}{VC(H)} + 1) + \ln\frac{4}{\delta}}{N}}$$