Foundations for Machine Learning

L. Y. Stefanus

TU Dresden, June-July 2019

Reference

- Shai Shalev-Shwartz and Shai Ben-David. UNDERSTANDING MACHINE LEARNING: From Theory to Algorithms. Cambridge University Press, 2014.

Definition of PAC (Probably Approximately Correct) Learnability

A hypothesis class H is PAC learnable if there exists a function $m_H: (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over X, and for every labeling function $f: X \to \{0,1\}$, if the realizable assumption holds with respect to $\mathrm{H}, \mathrm{D}, \mathrm{f}$, then when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D and labeled by f, the algorithm returns a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the samples), $L_{(D,f)}(h) \leq \epsilon$.

- The accuracy parameter ϵ determines how far the resulted classifier can be from the optimal one (this corresponds to the "approximately correct" part of PAC), and a confidence parameter $(1 - \delta)$ indicating how likely the classifier is to meet that accuracy requirement (corresponds to the "probably" part of PAC).
- Under the data access model that we are investigating, these approximations are inevitable. Since the training set is randomly generated, there may always be a small chance that it will happen to be non-representative (for example, the training set might contain only one domain point, sampled over and over again). Furthermore, even when we are lucky enough to get a training sample that does faithfully represent D, because it is just a finite sample, there may always be some fine details of D that it fails to reflect. The accuracy parameter, ϵ, allows "forgiving" the learner's classifier for making minor errors.

Sample Complexity

- The function $m_H(\epsilon, \delta)$ determines the sample complexity of learning H: that is, how many samples are required to guarantee a probably approximately correct solution.
- Note that if H is PAC learnable, there are many functions $m_H(\epsilon, \delta)$ that satisfy the requirements given in the definition of PAC learnability. Therefore, to be precise, we will define the sample complexity of learning H to be the minimal function, in the sense that $m_H(\epsilon, \delta)$ gives the minimal integer that satisfies the requirements of PAC learning.
- Using the terminology of sample complexity, Theorem 1 can be expressed as follows:

Corollary 2

- Every finite hypothesis class is PAC learnable with sample complexity

$$m_H(\epsilon, \delta) \leq \left\lceil \frac{\ln(|\mathrm{H}|/\delta)}{\epsilon} \right\rceil$$

Releasing the Realizability Assumption

- Recall that the realizability assumption requires that there exists $h^* \in H$ such that $\mathbb{P}_{x \sim D}[h^*(x) = f(x)] = 1$. In many practical problems this assumption does not hold. Furthermore, it is maybe more realistic not to assume that the labels are fully determined by the features we measure on input elements (in the case of the papayas learning, it is plausible that two papayas of the same color and softness will have different taste). In the following, we relax the realizability assumption by replacing the "target labeling function" with a more flexible notion, a data-labels generating distribution.

- Formally, let D be a probability distribution over $X \times Y$, where, X is our domain set and Y is a set of labels (usually $Y = \{0,1\}$). That is, D is a joint distribution over domain points and labels.
- One can view such a distribution as being composed of two parts: a distribution D_x over unlabeled domain points (sometimes called the marginal distribution) and a conditional probability over labels for each domain point, $D((x,y)|x)$. In the papaya example, D_x determines the probability of encountering a papaya whose color and softness fall in some color-softness values domain, and the conditional probability is the probability that a papaya with color and softness represented by x is tasty. Indeed, such modeling allows for two papayas that share the same color and softness to belong to different taste categories.

The Revised True Error

- For a probability distribution, D, over X × Y, one can measure how likely h is to make an error when labeled points are randomly drawn according to D. We redefine the true error (or risk or loss) of a prediction rule h to be

$$L_D(h) \stackrel{\text{def}}{=} \mathbb{P}_{(x,y)\sim D}[h(x) \neq y] \stackrel{\text{def}}{=} D(\{(x,y): h(x) \neq y\})$$

- We would like to find a predictor, h, for which this error will be minimized. However, the learner does not know the data generating D. What the learner does have access to is the training data, S.

- The definition of the empirical risk remains the same as before, namely,

$$L_S(h) \stackrel{\text{def}}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

- Given S, a learner can compute $L_S(h)$ for any function $h: X \rightarrow Y$.
- We wish to find some hypothesis, $h: X \rightarrow Y$, that probably approximately minimizes the true risk, $L_D(h)$.

Definition of Agnostic PAC Learnability

A hypothesis class H is Agnostic PAC learnable if there exists a function $m_H: (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over $\mathrm{X} \times \mathrm{Y}$, when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D, the algorithm returns a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the m training samples),

$$L_D(h) \leq \min_{g \in H} L_D(g) + \epsilon.$$

Agnostic PAC learning generalizes PAC learning

- If the realizability assumption holds, agnostic PAC learning provides the same guarantee as PAC learning. In that sense, agnostic PAC learning generalizes PAC learning.
- When the realizability assumption does not hold, no learner can guarantee an arbitrarily small error. Nevertheless, under the definition of agnostic PAC learning, a learner can still declare success if its error is not much larger than the best error achievable by a predictor from the class H.
- This is in contrast to PAC learning, in which the learner is required to achieve a small error in absolute terms and not relative to the best error achievable by the hypothesis class.

Extension to a variety of learning tasks

- Multiclass Classification
  - Our classification does not have to be binary. Take, for example, the task of document classification according to topics.
- Regression
  - We wish to find some simple pattern in the data, i.e., a functional relationship between the X and Y components of the data. Here the target set Y is the the set of real numbers.

Generalized Loss Functions

- To accommodate a wide range of learning tasks we generalize our formalism of the measure of success as follows:
  - Given any set H (that plays the role of our hypotheses, or models) and some domain Z let ℓ be any function from H $\times$ Z to the set of nonnegative real numbers,

$$\ell: H \times Z \to \mathbb{R}_+$$

  - The risk function is defined to be the expected loss of a classifier, h $\in$ H, with respect to a probability distribution D over Z, namely,

$$L_D(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim D} [\ell(h, z)] .$$

- Similarly, we define the empirical risk to be the expected loss over a given sample $S = (z_1, \ldots, z_m) \in Z^m$, namely,

$$L_S(h) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i).$$

- For example, the loss function used in the binary and multiclass classification tasks is the 0-1 loss:

$$\ell_{0-1}(h, (x, y)) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if} \quad h(x) = y \\ 1 & \text{if} \quad h(x) \neq y \end{cases}$$

The random variable z ranges over the set of pairs $X \times Y$.

Definition of Agnostic PAC Learnability for General Loss Functions

A hypothesis class H is Agnostic PAC learnable with respect to a set Z and a loss function $\ell: H \times Z \to \mathbb{R}_+$, if there exists a function $m_H: (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over Z, when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D, the algorithm returns a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the m training samples),

$$L_D(h) \leq \min_{g \in H} L_D(g) + \epsilon,$$

where $L_D(h) = \underset{z \sim D}{\mathbb{E}}[\ell(h, z)]$.

A Bit of History

- PAC learning was introduced by Leslie Valiant (1984).
  - Valiant, L. G. (1984), “A theory of the learnable,” *Communications of the ACM* 27(11), 1134-1142.
- Valiant was named the winner of the 2010 Turing Award for the introduction of the PAC model.

Machine Learning

+

Big Data

⇓

Industrially Useful Predictor

Learning via Uniform Convergence

- The idea behind uniform convergence is very simple.
- Recall that, given a hypothesis class, H, the ERM learning paradigm works as follows: Upon receiving a training sample, S, the learner evaluates the risk (or error) of each h in H on the given sample and outputs a member of H that minimizes this empirical risk.
- The hope is that an h that minimizes the empirical risk with respect to S is a risk minimizer w.r.t. the true data probability distribution as well. For that, it suffices to ensure that the empirical risks of all members of H are good approximations of their true risk. This condition is formalized as follows.

definition of $\in$-representative sample

A training set S is called $\in$-representative (w.r.t. domain Z, hypothesis class H, loss function ℓ, and distribution D) if

$$\forall h \in H, |L_S(h) - L_D(h)| \leq \epsilon.$$

LEMMA 4.2 *Assume that a training set S is $\frac{\epsilon}{2}$-representative (w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function ℓ, and distribution $\mathcal{D}$). Then, any output of* $\mathrm{ERM}_{\mathcal{H}}(S)$, *namely, any* $h_S \in \mathrm{argmin}_{h\in\mathcal{H}} L_S(h)$, *satisfies*

$$L_{\mathcal{D}}(h_S) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon.$$

Proof

For every $h \in H$,

$$L_D(h_S) \leq L_S(h_S) + \frac{\epsilon}{2} \leq L_S(h) + \frac{\epsilon}{2} \leq L_D(h) + \frac{\epsilon}{2} + \frac{\epsilon}{2}$$
$$= L_D(h) + \epsilon,$$

where the first and third inequalities are due to the assumption that S is $\frac{\epsilon}{2}$-representative and the second inequality holds since h_S is an ERM predictor.

definition of Uniform Convergence

A hypothesis class H has the uniform convergence property (w.r.t. a domain Z and a loss function ℓ) if there exists a function $m_H^{\mathrm{UC}}: (0,1) \times (0,1) \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and for every probability distribution D over Z, if S is a sample of $m \geq m_H^{\mathrm{UC}}(\epsilon, \delta)$ instances drawn i.i.d. according to D, then, with probability of at least $1 - \delta$,

S is ϵ-representative.

- Note: The term uniform here refers to having a fixed sample size that works for all members of H and over all possible probability distributions over the domain.

- The following corollary follows directly from Lemma 4.2 and the definition of uniform convergence.

COROLLARY 4.4 *If a class $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostically PAC learnable with the sample complexity $m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\epsilon/2, \delta)$. Furthermore, in that case, the $\mathrm{ERM}_{\mathcal{H}}$ paradigm is a successful agnostic PAC learner for $\mathcal{H}$.*

every finite hypothesis class is agnostic PAC learnable

COROLLARY 4.6 *Let $\mathcal{H}$ be a finite hypothesis class, let Z be a domain, and let $\ell : \mathcal{H} \times Z \to [0,1]$ be a loss function. Then, $\mathcal{H}$ enjoys the uniform convergence property with sample complexity*

$$m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil .$$

Furthermore, the class is agnostically PAC learnable using the ERM algorithm with sample complexity

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\epsilon/2, \delta) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil .$$

The Bias-Complexity Tradeoff

The No-Free-Lunch Theorem

The Bias-Complexity Tradeoff

- We have seen that unless one is careful, the training data can mislead the learner, and result in overfitting.
- To overcome this problem, we restricted the search space to some hypothesis class H. Such a hypothesis class can be viewed as reflecting some prior knowledge that the learner has about the task (a belief that one of the members of the class H is a low-error model for the task).
- For example, in our papayas problem, on the basis of previous experience with other fruits, we may assume that some rectangle in the color-softness plane predicts (at least approximately) the papaya's tastiness.

- Is such prior knowledge really necessary for the success of learning?
- Maybe there exists some kind of universal learner, that is, a learner who has no prior knowledge about a certain task and is ready to be challenged by any task?
- What is a universal learner? A specific learning task is defined by an unknown distribution D over $X \times Y$, where the goal of the learner is to find a predictor $h: X \to Y$, whose risk, $L_D(h)$, is small enough. The question is therefore whether there exists a learning algorithm A and a training set size m, such that for every distribution D, if A receives m i.i.d. instances from D, there is a high chance it outputs a predictor h that has a low risk.

- We will study the No-Free-Lunch theorem which states that no such universal learner exists.

Theorem (No Free Lunch)

Let A be any learning algorithm for the task

of binary classification with respect to the 0-1 loss function over a domain X. Let m be any number smaller than $|X|/2$, representing a training set size. Then, 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. With probability of at least $1/7$ over the choice of $S \sim D^m$ we have that $L_D(A(S)) \geq 1/8$.

- This theorem states that for every learner, there exists a task on which it fails, even though that task can be successfully learned by another learner.
- In this case, a successful learner would be an ERM_H learner w.r.t. any finite hypothesis class that contains f and whose sample size satisfies the condition $m \geq 8 \ln(7 |H| / 6)$, where $\epsilon = \frac{1}{8}, \delta = \frac{6}{7}$.

Proof of No-Free-Lunch Theorem

1. Let C be a subset of X of size 2m. The intuition of the proof is that any learning algorithm that observes only half of the instances in C has no information on what should be the labels of the rest of the instances in C. Therefore, there exists some target function f, that would contradict the labels that A(S) predicts on the unobserved instances in C.
2. Note that there are $T = 2^{2m}$ possible functions from C to {0,1}. Let us denote these functions by $f_1, \ldots, f_T$. For each such function, let D_i be a distribution over $C \times \{0,1\}$ defined by $D_i(\{(x,y)\}) = \begin{cases} 1/|C| & \text{if } y = f_i(x) \\ 0 & \text{otherwise} \end{cases}$

3. That is, the probability to choose a pair (x, y) is $1/|C|$ if the label y is indeed the true label according to f_i, and the probability is 0 if $y \neq f_i(x)$. Therefore, $L_{D_i}(f_i) = 0$.
4. We will first show that for every algorithm, A, that receives a training set of m instances from $C \times \{0,1\}$ and returns a function $A(S): C \to \{0,1\}$, it holds that

$$\max_{i \in [T]} \mathbb{E}_{S \sim D_i^m} [L_{D_i}(A(S))] \geq \frac{1}{4}. \qquad (5.1)$$

5. There are $k = (2m)^m$ possible sequences of m instances from C. Let us denote these sequences by $S_1, \ldots, Sk$. Also, for $S_j = (x_1, \ldots, x_m)$ we denote by S_j^i the sequence containing the instances in S_j labeled by the function f_i, namely, $S_j^i = ((x_1, fi(x_1)), \ldots, (x_m, fi(x_m)))$. If the distribution is D_i then the possible training sets that A can receive are S_1^i, …, S_k^i, and all these training sets have the same probability of being sampled. Therefore,

$$\mathbb{E}_{S \sim D_i^m}[L_{D_i}(A(S))] = \frac{1}{k} \Sigma_{j=1}^{k} L_{D_i}(A(S_j^i)). \quad (5.3)$$

6. Using the facts that “maximum” is larger than “average” and that “average” is larger than “minimum,” we have

$$\begin{aligned}
\max_{i\in[T]} \frac{1}{k}\sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{k}\sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) \\
&= \frac{1}{k}\sum_{j=1}^{k}\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)) \\
&\geq \min_{j\in[k]}\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)). \qquad (5.4)
\end{aligned}$$

7. Next, we fix some $j \in [k]$. Denote $S_j = (x_1, \ldots, x_m)$ and let $v_1, \ldots, v_p$ be the instances in C that do not appear in S_j. We have that $p \geq m$. Therefore, for every function $h: C \to \{0,1\}$ and every i we obtain

$$
\begin{aligned}
L_{\mathcal{D}_i}(h) &= \frac{1}{2m} \sum_{x \in C} \mathbb{1}_{[h(x) \neq f_i(x)]} \\
&\geq \frac{1}{2m} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]} \\
&\geq \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]}.
\end{aligned}
$$

where $\mathbb{1}_{[\text{boolean expression}]}$ denotes indicator function (equals 1 if boolean expression is true and 0 otherwise).

8. Hence,

$$\begin{aligned}
\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S^i_j)) &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2p}\sum_{r=1}^{p}\mathbb{1}_{[A(S^i_j)(v_r)\neq f_i(v_r)]} \\
&= \frac{1}{2p}\sum_{r=1}^{p}\frac{1}{T}\sum_{i=1}^{T}\mathbb{1}_{[A(S^i_j)(v_r)\neq f_i(v_r)]} \\
&\geq \frac{1}{2}\cdot\min_{r\in[p]}\frac{1}{T}\sum_{i=1}^{T}\mathbb{1}_{[A(S^i_j)(v_r)\neq f_i(v_r)]}. \qquad (5.6)
\end{aligned}$$

9. Next, we fix some $r \in [p]$. We can partition all the functions in $f_1, \ldots, f_T$ into T/2 disjoint pairs, where for a pair (fi, fi') we have that for every $c \in C$, $f_i(c) \neq f_{i'}(c)$ if and only if $c = v_r$. Since for such a pair we must have $S_j^i = S_j^{i'}$, it follows that

$$\mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} + \mathbb{1}_{[A(S_j^{i'})(v_r) \neq f_{i'}(v_r)]} = 1,$$

which yields

$$\frac{1}{T} \sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} = \frac{1}{2}.$$

10. Combining the previous equation with Equation (5.6), Equation (5.4) and Equation (5.3), we obtain Equation (5.1):

$$\max_{i \in [T]} \mathbb{E}_{S \sim D_i^m} [L_{D_i}(A(S))] \geq \frac{1}{4}.$$

This means that for every algorithm A' that receives a training set of m instances from $X \times \{0,1\}$ there exists a function $f: X \to \{0,1\}$ and a distribution D over $X \times \{0,1\}$, such that $L_D(f) = 0$ and

$$\mathbb{E}_{S \sim Dm} [L_D(A'(S))] \geq \frac{1}{4}. \quad (*)$$

11. Applying Markov's Inequality from the Theory of Probability to Equation (*), we obtain

$$\mathbb{P}[LD(A'(S)) \geq 1/8] \geq 1/7$$

which is what we need to prove.

Q.E.D

Markov's Inequality

Let X be a non-negative random variable and suppose that $E(X)$ exists. For any $t > 0$,

$$\mathbb{P}[X > t] \leq \frac{E(X)}{t}.$$

- Let Z be a random variable that takes values in [0,1]. Assume that E[Z] = μ. Let Y = 1 - Z. Then Y is a non-negative random variable with E[Y] = 1 – E(Z) = 1- μ. Applying Markov's inequality on Y with $a \in (0,1)$, we obtain

$$\mathbb{P}[Z < 1 - a] = \mathbb{P}[1 - Z > a] = \mathbb{P}[Y > a] \leq \frac{E(Y)}{a} = \frac{1-\mu}{a}$$

Therefore, $\mathbb{P}[Z \geq 1 - a] \geq 1 - \frac{1-\mu}{a} = \frac{a+\mu-1}{a}$, which implies that

$$\mathbb{P}[Z \geq a] \geq \frac{\mu - a}{1 - a}$$

No-Free-Lunch Theorem and Prior Knowledge

- How does the No-Free-Lunch result relate to the need for prior knowledge?
- Let us consider an ERM predictor over the hypothesis class H of all the functions f from X to {0,1}. This class represents lack of prior knowledge: Every possible function from the domain to the label set is considered a good candidate.
- According to the No-Free-Lunch theorem, any algorithm that chooses its output from hypotheses in H, and in particular the ERM predictor, will fail on some learning task. Therefore, this class is not PAC learnable.

- How can we prevent such failures?
- We can escape the hazards foreseen by the No-Free-Lunch theorem by using our prior knowledge about a specific learning task, to avoid the distributions that will cause us to fail when learning that task.
- Such prior knowledge can be expressed by restricting our hypothesis class.
- But how should we choose a good hypothesis class?

- On one hand, we want to believe that this class includes the hypothesis that has no error at all (in the PAC setting), or at least that the smallest error achievable by a hypothesis from this class is rather small (in the agnostic setting). On the other hand, we have just seen that we cannot simply choose the richest class (the class of all functions over the given domain).

The Bias-Complexity Tradeoff

Error Decomposition

How should we choose a good hypothesis class?

- To answer this question we decompose the error of an ERM_H predictor into two components as follows.
- Let h_S be an ERM_H hypothesis. Then, we can write

$$L_D(h_S) = \varepsilon_{\text{app}} + \varepsilon_{\text{est}}$$

- where

$$\varepsilon_{\text{app}} = \min_{h \in H} L_D(h)$$

$$\varepsilon_{\text{est}} = L_D(h_S) - \varepsilon_{\text{app}}$$

The Approximation Error ε_{app}

- The approximation error is the minimum risk achievable by a predictor in the hypothesis class. This term measures how much risk we have because we restrict ourselves to a specific class, namely, how much inductive bias we have.
- The approximation error does not depend on the sample size and is determined by the hypothesis class chosen.
- Enlarging the hypothesis class can decrease the approximation error.
- Under the realizability assumption, the approximation error is zero. In the agnostic case, however, the approximation error can be large.

The Estimation Error ε_{est}

- The estimation error is the difference between the approximation error and the error achieved by the ERM predictor.
- The estimation error occurs because the empirical risk (i.e., training error) is only an estimate of the true risk, and so the predictor minimizing the empirical risk is only an estimate of the predictor minimizing the true risk.
- The quality of this estimation depends on the training set size and on the size, or complexity, of the hypothesis class.

The Estimation Error ε_{est}

- As we have studied, for a finite hypothesis class, ε_{est} increases (logarithmically) with |H| and decreases with m.
- We can think of the size of H as a measure of its complexity. Later we will study another complexity measure of hypothesis classes, called VC dimension.

the bias-complexity tradeoff

- Since our goal is to minimize the total risk, we face a tradeoff, called the bias-complexity tradeoff.
- On one hand, choosing H to be a very rich class decreases the approximation error but at the same time might increase the estimation error, as a rich H might lead to overfitting.
- On the other hand, choosing H to be a very small set reduces the estimation error but might increase the approximation error or, in other words, might lead to underfitting.

the bias-complexity tradeoff

- A great choice for H is the class that contains only one classifier -- the Bayes optimal classifier. But the Bayes optimal classifier depends on the underlying distribution D, which we do not know. Indeed, learning would have been unnecessary if we had known D.
- Learning theory studies how rich we can make H while still maintaining reasonable estimation error.
- In many cases, empirical research focuses on designing good hypothesis classes for a certain domain.

The VC-Dimension

Which classes H are PAC learnable?

- So far we have seen that finite classes are learnable, but that the class of all functions (over an infinite size domain) is not.
- What makes one class learnable and the other not learnable? Can infinite-size classes be learnable, and, if so, what determines their sample complexity?

Infinite-Size Classes Can Be Learnable

- To show that the size of the hypothesis class is not the right characterization of its sample complexity, we first present a simple example of an infinite-size hypothesis class that is learnable.

Infinite-Size Classes Can Be Learnable

Example

- Let H be the set of threshold functions over the real line, namely,

$$H = \{h_a : a \in \mathbb{R}\}$$

where $h_a : \mathbb{R} \to \{0,1\}$ is a function such that

$$h_a(x) = \mathbf{1}_{[x<a]} = \begin{cases} 1 & \text{if } x < a \\ 0 & \text{otherwise} \end{cases}$$

- This H is of infinite size. Nevertheless, the following lemma shows that H is learnable in the PAC model using the ERM algorithm.

Lemma 6.1

Let H be the class of threshold functions as defined earlier. Then, H is PAC learnable, using the ERM rule, with sample complexity of

$$m_H(\epsilon, \delta) \leq \left\lceil \frac{\ln\left(\frac{2}{\delta}\right)}{\epsilon} \right\rceil.$$

Proof: [See the textbook]

Motivation for VC-dimension

- Lemma 6.1 shows that while finiteness of H is a sufficient condition for learnability, it is not a necessary condition.
- We demonstrate that a property called the VC-dimension of a hypothesis class gives the correct characterization of its learnability.
- In the proof of the No-Free-Lunch theorem, we have shown that without restricting the hypothesis class, for any learning algorithm, an adversary can construct a distribution for which the learning algorithm will perform poorly, while there is another learning algorithm that can succeed on the same distribution. To do so …

Motivation for VC-dimension

- To do so, the adversary used a finite set $C \subset X$ and considered a family of distributions that are concentrated on elements of C. Each distribution was derived from a "true" target function from C to {0,1}.
- To make any algorithm fail, the adversary used the power of choosing a target function from the set of all possible functions from C to {0,1}.
- When considering PAC learnability of a hypothesis class H, the adversary is restricted to constructing distributions for which some hypothesis $h \in H$ achieves a zero risk. Since we are considering distributions that are concentrated on elements of C, we should study how H behaves on C, which leads to the following definition.

DEFINITION 6.2 (Restriction of $\mathcal{H}$ to C) Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$ and let $C = \{c_1, \ldots, c_m\} \subset \mathcal{X}$. The restriction of $\mathcal{H}$ to C is the set of functions from C to $\{0,1\}$ that can be derived from $\mathcal{H}$. That is,

$$\mathcal{H}_C = \{(h(c_1), \ldots, h(c_m)) : h \in \mathcal{H}\},$$

where we represent each function from C to $\{0,1\}$ as a vector in $\{0,1\}^{|C|}$.

If the restriction of H to C is the set of all functions from C to {0,1}, then we say that H shatters the set C.

DEFINITION 6.3 (Shattering) A hypothesis class $\mathcal{H}$ shatters a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions from C to $\{0,1\}$. That is, $|\mathcal{H}_C| = 2^{|C|}$.

Examples of shattering (1)

- Let H be the class of threshold functions over $\mathbb{R}$.
- Take a set $C = \{c_1\}$. Now, if we take $a = c_1 + 1$, then we have $h_a(c_1) = 1$, and if we take $a = c_1 - 1$, then we have $h_a(c_1) = 0$. Therefore, H_C is the set of all functions from C to {0,1}, and H shatters C.
- Now take a set $C = \{c_1, c_2\}$, where $c_1 \leq c_2$. No $h \in H$ can account for the labeling (0, 1), because any threshold that assigns the label 0 to c_1 must assign the label 0 to c_2 as well. Therefore not all functions from C to {0,1} are included in H_C; hence C is not shattered by H.

Examples of shattering (2)

- Let H be the class of intervals over $\mathbb{R}$, namely,

$$H = \{h_{a,b} : a, b \in \mathbb{R}, a < b\},$$

  where $h_{a,b} : \mathbb{R} \to \{0,1\}$ is a function such that

$$h_{a,b}(x) = \mathbf{1}_{[x \in (a,b)]}.$$

- Take the set $C = \{c_1, c_2\}$, where $c_1 \leq c_2$. Then H shatters C.
- Now take a set $C = \{c_1, c_2, c_3\}$ where $c_1 \leq c_2 \leq c_3$. Then the labeling (1,0,1) cannot be obtained by an interval and therefore H does not shatter C.

VC-dimension

Definition

The VC-dimension of a hypothesis class H, denoted VCdim(H), is the maximal size of a set $C \subset X$ that can be shattered by H. If H can shatter sets of arbitrarily large size we say that H has infinite VC-dimension.

- Note: VC = Vapnik-Chervonenkis,
  from Vladimir Vapnik and Alexey Chervonenkis

How to compute VCdim

To show that VCdim(H) = d we need to show that

1. There exists a set C of size d that is shattered by H.
2. Every set C of size $d + 1$ is not shattered by H.

Example 1

Threshold Functions

- Let H be the class of threshold functions over R. In previous example, we have shown that for an arbitrary set $C = \{c_1\}$, H shatters C; therefore $VCdim(H) \geq 1$.
- We have also shown that for an arbitrary set $C = \{c_1, c_2\}$ where $c1 \leq c2$, H does not shatter C; therefore $VCdim(H) \leq 1$.
- We conclude that $VCdim(H) = 1$.

Example 2

Axis Aligned Rectangles

- Let H be the class of axis aligned rectangles:

$$H = \{ h_{(a1,a2,b1,b2)} : a1 \leq a2 \text{ and } b1 \leq b2 \}$$

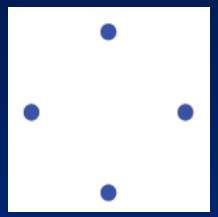
where

$$h_{(a1,a2,b1,b2)}(x,y) = \begin{cases} 1 & \text{if } a1 \leq x \leq a2 \text{ and } b1 \leq y \leq b2 \\ 0 & \text{otherwise} \end{cases}$$

- VCdim(H) = 4.

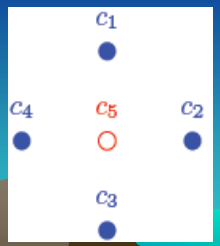
- Proof:

- We need to find a set of 4 points that are shattered by H, and show that no set of 5 points can be shattered by H.
- There is a set of 4 points that is shattered by H:



- Next consider any set $C \subset \mathbf{R}^2$ of 5 points. In C, select a leftmost point, a rightmost point, a lowest point, and a highest point. Without loss of generality, denote C = { c1, c2, c3, c4, c5 } and let c5 be the point that was not selected.
- Now, define the labeling (1, 1, 1, 1, 0). It is impossible to obtain this labeling by an axis aligned rectangle. Indeed, such a rectangle must contain c1, c2, c3, c4; but it must also contain c5, because the coordinates of c5 are within the intervals defined by the selected points. So, C is not shattered by H, and therefore VCdim(H) = 4.

■



Example 3

Finite Classes

- Let H be a finite class.
- Then, for any set C we have $|H_C| \leq |H|$ and thus C cannot be shattered if $|H| < 2^{|C|}$, namely, if $|C| > \log_2(|H|)$.
- This implies that $VCdim(H) \leq \log_2(|H|)$.
- However, the VC-dimension of a finite class H can be significantly smaller than $\log_2(|H|)$.
- For example, let $X = \{1, \ldots, k\}$, for some integer k, and consider the class of threshold functions. Then, $|H| = k$ but $VCdim(H) = 1$.
- Since k can be arbitrarily large, the gap between $\log_2(|H|)$ and $VCdim(H)$ can be arbitrarily large.

Theorem 6.7
(The Fundamental Theorem of Statistical Learning)

Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0,1\}$ and let the loss function be the 0−1 loss. Then, the following are equivalent:

1. $\mathcal{H}$ has the uniform convergence property.
2. Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.
3. $\mathcal{H}$ is agnostic PAC learnable.
4. $\mathcal{H}$ is PAC learnable.
5. Any ERM rule is a successful PAC learner for $\mathcal{H}$.
6. $\mathcal{H}$ has a finite VC-dimension.