PAC learning

Charlotte Aten

University of Denver

2022 October 17

Introduction

- This talk is an introduction to the formal theory of statistical learning.
- We will introduce the Probably Approximately Correct (PAC) learning model, which was described by Valiant in 1984 following foundational work by Vapnik and Chervonenkis in the 1970s.
- These slides follow the treatment in Understanding Machine Learning by Shai Shalev-Shwartz and Shai Ben-David.

The papaya story

- You are on a Pacific Island where papayas are a significant part of the local diet.
- Initial condition: You have never tasted papayas.
- Goal: Learn how to predict whether the papayas you see at the market are tasty or not.
- Features: You will make your predictions based on color and softness, as per your experience with other fruit.

A bit more formally...

- We will work with a *domain set* X, which in this case is the set of all possible papayas.
- This set is often a vector of features. In this case a "papaya" is a pair of a color and a softness.
- We also have a *label set* Y, which in this case is $\{0, 1\}$ where 1 means "tasty" and 0 means "not tasty".
- Our experience gives us a *training set*

  $$S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$$

  of pairs in $X \times Y$.
- For example, $((\text{red}, \text{firm}), 1)$ may be a member of S in our case.

A bit more formally...

- We imagine that we feed all of this information into a (perhaps abstract) machine, our *learner*.
- We would like our learner to output a function

$$h\colon X \to Y,$$

  which we call a *predictor* (or *classifier*).
- This function is supposed to determine whether a papaya with given features is tasty or not.
- In order to understand whether our learner has done a good job, we need to understand how the training set is generated.

Generating training data

- Assumption #1: The instances (the papayas in this case) are generated by some probability distribution D which is not known to the learner.
- Assumption #2: There exists a “correct” labeling function $f: X \to Y$ which is also unknown to the learner.
- The training set S is then generated by choosing the x_i according to the probability distribution D and the labeling function f which maps the vector of features x_i to the label y_i.
- We are now ready to give a formal measure of the success of our learner’s predictor.

Measuring success

- The *error* of a classifier is the probability that the classifier does not predict the correct label on a random data point generated by the probability distribution D.
- Formally, given an event $A \subset X$ we have that $D(A)$ is a number which determines how likely it is to observe some $x \in A$.
- Even more formally, D defines a probability measure on X which assigns to each (measurable) $A \subset X$ its measure $D(A) \in [0, 1]$.

Measuring success

- We define the *prediction rule error* of a classifier h with given distribution D and correct labeling function f by

$$L_{D,f}(h) := P_{x\sim D}[h(x) \neq f(x)] := D(\{\, x \in X \mid h(x) \neq f(x) \,\}).$$

- Remember that the only way the learner can interact with the environment is through the training set, so the learner is blind to the underlying distribution D and the correct labeling function f. In our papayas example, we have just arrived on a new island and have no idea as to how papayas are distributed or how to judge their tastiness before actually eating them.

A simple learning paradigm: Empirical Risk Minimization

- We now give an example of an algorithm for learning.
- Algorithm input: A training set S, sampled from an unknown distribution D and labeled by some target function f.
- Algorithm output: The function $h_S \colon X \to Y$ that minimized the error $L_{D,f}(h)$ with respect to the unknown D and f.
- Difficulty: We don't know what D and f are, so the true value of $L_{D,f}(h)$ is also unknown to us.
- We instead use the *training error*

$$L_S(h) := \frac{1}{m} \left|\{\, i \in [m] \mid h(x_i) \neq y_i \,\}\right| .$$

A simple learning paradigm: Empirical Risk Minimization

- Since the training sample is the only information about the world available to the learner it makes sense to look for a solution $h\colon X \to Y$ which works well on that data.
- The learning paradigm to generate a predictor h which minimizes $L_S(h)$ is called *Empirical Risk Minimization (ERM)*.
- It is not obvious how to implement ERM in practice, but we will leave that for another time.

Another nightmare: Overfitting

- Consider a new learning task where $X := [0,1]^2$, $Y := \{0,1\}$, $f := 1_{x_1 \le \frac{1}{2}}$, and D is the distribution given by the Lebesgue measure on the square.
- Given any finite training set S we can make a predictor

$$h_S(x) := \begin{cases} y_i & \text{when } x = x_i \text{ for some } i \in [m] \\ 0 & \text{otherwise} \end{cases}.$$

- Clearly we have $L_S(h_S) = 0$ so this predictor looks good on our training set.
- However, the true error for such a predictor is

$$L_{D,f}(h_S) = D\left(\left\{ x \in [0,1]^2 \;\middle|\; h_S(x) \neq 1_{x_1 \le \frac{1}{2}} \right\}\right) = \frac{1}{2}.$$

Another nightmare: Overfitting

- This situation is not as artificial as it may seem, and it will not always be obvious when a predictor of this form arises in the real world.
- Also, we must consider extremal situations like this when formulating a general approach to a learning task.

ERM with inductive bias

- One way to deal with overfitting is by introducing a *hypothesis class* $\mathcal{H} \subset Y^X$ from which will will assume our correct labeling function $f: X \to Y$ has been chosen.
- Our ERM learner can then use this additional assumption that $f \in \mathcal{H}$ along with the training set S to make a predictor $h: X \to Y$.
- Ideally $\mathcal{H}$ should be chosen appropriately for the problem at hand, but we will come to that another time.

Finite hypothesis class

- The easiest way to restrict the class of hypotheses is by imposing an upper bound on its size.
- It turns out that if $\mathcal{H}$ is finite then $\mathrm{ERM}_{\mathcal{H}}$ will not overfit provided that it is based on a sufficiently large training sample as a function of the size of $\mathcal{H}$.
- Given a training set S and correct labeling $f: X \to Y$ we choose

  $$h_S \in \mathrm{argmin}_{h \in \mathcal{H}} L_S(h),$$

  which is a hypothesis which achieves the minimum value of L_S over $\mathcal{H}$.

Realizability hypothesis

- We assume there exists some $h^* \in \mathcal{H}$ such that $L_{D,f}(h^*) = 0$.
- This implies that with probability 1 over random samples S we have that $L_S(h^*) = 0$.
- This assumption is not very realistic, but it is a good place to start. What we would like to know is the *true risk* $L_{D,f}(h_S)$.
- We now will make a reasonable assumption about the relationship between D and S.

The i.i.d. assumption

- We assume the examples in the training set are independent and identically distributed (i.i.d.) according to the probability distribution D.
- The issue here is that $L_{D,f}(h_S)$ depends on a randomly chosen S, so there is a randomness in the choice of the predictor. That is, $L_{D,f}(h_S)$ is a random variable.

Accuracy parameter

- We can never guarantee that the set S we choose will suffice to direct the learner toward a good classifier.
- We also cannot guarantee perfect label prediction, so we introduce the *accuracy parameter* ϵ.
- Success is choosing h_S with $L_{D,f}(h_S) \leq \epsilon$ and failure is choosing h_S with $L_{D,f}(h_S) \geq \epsilon$.
- We can only have $L_{D,f}(h_S) > \epsilon$ if our sample is in the set of misleading examples.

Accuracy parameter

- One can show that if $\mathcal{H}$ is a finite hypothesis class, $\delta \in (0, 1)$, $\epsilon > 0$, and $m \in \mathbb{Z}$ satisfies

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

  then for any labeling function f and any distribution D for which the realizability assumption holds we have that with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m for every ERM hypothesis h_S it holds that $L_{D,f}(h_S) \leq \epsilon$.

VC-dimension

- Typically our hypothesis class $\mathcal{H}$ will not be finite but we can still obtain a similar result about how many samples we need to take in order to guarantee a choice of ERM hypothesis h_S it holds that $L_{D,f}(h_S) \leq \epsilon$ with a probability of at least $1 - \delta$.
- The key idea is to measure the complexity of the hypothesis class, rather than its size, and we can do this using the notion of VC-dimension.

VC-dimension

Definition (Restriction of $\mathcal{H}$ to C)

Let $\mathcal{H}$ be a class of functions from X to $\{0,1\}$ and let $C = \{c_1, \dots, c_m\} \subseteq X$. 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,

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

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

VC-dimension

Definition (Shattering)

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

VC-dimension

Definition (VC-dimension)

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

VC-dimension

Definition (Uniform convergence)

We say that a hypothesis class $\mathcal{H}$ has the *uniform convergence property* (with respect to a domain Z and a loss function ℓ) when there exists a function $m_{\mathcal{H}}^{UC}\colon (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and every probability distribution D over Z we have that if S is a sample of $m \geq m_{\mathcal{H}}^{UC}(\epsilon, \delta)$ examples drawn i.i.d. according to D then with probability at least $1 - \delta$ the set S is ϵ-representative.

VC-dimension

Theorem (The Fundamental Theorem of Statistical Learning)

Let $\mathcal{H}$ be a hypothesis class of functions from a domain X to $\{0, 1\}$ and let the loss function be the 01 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.*

VC-dimension

Theorem (The Fundamental Theorem of Statistical Learning (Quantitative Version))

Let $\mathcal{H}$ be a hypothesis class of functions from a domain X to $\{0, 1\}$ and let the loss function be the 01 loss. Assume that $VCdim(H) = d < \infty$. Then, there are absolute constants C_1, C_2 such that:

1. *$\mathcal{H}$ has the uniform convergence property with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

2. *$\mathcal{H}$ is agnostic PAC learnable with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

3. *$\mathcal{H}$ is PAC learnable with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

References

- Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. 32 Avenue of the Americas, New York, NY 10013-2473, USA: Cambridge University Press, 2014. ISBN: 978-1-107-05713-5