COMPUTABILITY AND COMPLEXITY 24
LEARNING

AMIR YEHUDAYOFF

Remark. *Computational complexity studies the resources that are needed to achieve computational tasks. On a high-level, computational devices have costs (like time, memory size, energy, randomness, training data, etc.), and computational tasks have complexities (the minimum cost that is needed to achieve it).*

1. The basics

We have been discussing the theory of computation. The models we used are TMs, and circuits. The resources we focussed on were time, space, randomness, and non-determinism. These mostly capture "classical methods" for using computers (by developing algorithms). Machine learning (ML) is about "new methods" for using computers. Instead of developing algorithms, we develop ways to transform data into algorithms. ML leads to new models, theories and challenges.

Remark. *There are many models for ML. Most models contain components of the following ideas:*

- *Data.*
- *A way to measure "loss" or "risk".*
- *Algorithms or algorithmic principles.*

We shall not survey these models, and focus on one example—the Probably Approximately Correct (PAC) model.

Notation. *The domain is a set X. For example, X can be the set of images (e.g., grayscale or RGB matrices).*

Notation. *We are interested in binary classification. That is, there is a function $f : X \to \{0,1\}$ that we "wish to learn". For example, $f(x)$ is 1 if the picture x contains a dog, and 0 otherwise.*

Notation. *A data point is a pair (x, y) where $x \in X$ and $y \in \{0,1\}$. For example, it is a picture x with 1, which indicates that "x contains a dog".*

Remark. *The data we collect is modeled by a sequence of data points*

$$(x_1, y_1), \ldots, (x_n, y_n).$$

This sequence is called a sample and is denoted by S.

Definition. *A learning algorithm is a function that transform samples to functions:*

$$A : (X \times \{0,1\})^n \to \{0,1\}^X.$$

Remark. *The output $h = A(S)$ allows to make "predictions" on unseen data (on all of X).*

On a high-level, from data generated by f but without knowing f we wish to output h that is close to f.

Remark. *This task is "too hard". For example, what is the next element in the sequence*

$$1, 2, 3, 4, \ldots?$$

We think it is 5 *because we have some "context". Without the context, any number could be next.*

Remark. *The "context" of a learning problem is captured by a class of functions $\mathcal{H} \subseteq \{0,1\}^X$. We assume that the unknown target function f comes from $\mathcal{H}$. This is called the realizable setting.*

Example. *$X = \mathbb{R}^2$ and $\mathcal{H}$ is the class of linear threshold functions (LTFs). Draw* 5 *points in the plane that are labelled by some LTF. Choose a point x inside a triangle with same label. What is the label of x?*

Remark. *How is the data generated? In "real life" we need to collect data and this is complicated for many reasons. A standard assumption in the theory is that the data consist of i.i.d. data points from some underlying unknown distribution. We shall work in the "realizable" setting. Namely, there is a distribution μ on $X \times \{0,1\}$ with the following property: there is $f \in \mathcal{H}$ so that μ is supported on points of the form $(x, f(x))$. The sample S is*

$$S = ((x_1, y_1), \ldots, (x_n, y_n)) \sim \mu^n$$

comprises i.i.d. draws from μ. This is the input to the learning algorithm.

Remark. *The output of the learning algorithm is $h = A(S)$ which is a function $X \to \{0,1\}$. How can we measure the "distance" between h and f? A beautiful and natural idea is to use μ again.*

Definition. *The true loss function of $h : X \to \{0,1\}$ with respect to μ (a.k.a. population loss) is*

$$L_\mu(h) = \Pr_{(x,y)\sim\mu}[h(x) \neq y].$$

Remark. *We do not know the true loss.*

Definition. *The loss of algorithm A with respect to μ is*

$$L_\mu(A) = \Pr_S[L_\mu(A(S))]$$

where $S \sim \mu^n$.

Remark. *The distribution μ plays the role of "nature". It both generates the input data (from the past), and measures the success of the algorithm (on future data).*

Definition. *An algorithm A has error $\varepsilon > 0$ in the PAC model with respect to $\mathcal{H}$ if for every $\mathcal{H}$-realizable distribution μ,*

$$L_\mu(A) < \varepsilon.$$

(Note that, in our notation, A operates on sample of fixed length n.)

Definition. *We think of $\mathcal{H}$ as a "learning problem". Our "context" is that the target function comes from $\mathcal{H}$, and our goal is to learn it.*

Definition. *The sample complexity of PAC learning $\mathcal{H}$ is the minimum n so that there is an algorithm that uses n samples and PAC learn $\mathcal{H}$ with error $\frac{1}{3}$.*

Remark. *Again, devices have costs and problems have complexities. The cost we currently focus on is the number of data points we need to collect.*

Remark. *There are more parameters that we could have introduced, but we keep it simple for this introduction.*

2. A characterization

The complexity of a problem is an integer. Figuring out this integer is fundamental and usually hard. There is a theory for computing the PAC sample complexities of problems. This is often called VC theory (after Vapnik and Chervonenkis).

Remark. *We have a fixed learning task $\mathcal{H} \subseteq \{0,1\}^X$ and we want to figure out the sample complexity of PAC learning it. To do so, we need a mechanism to measure how complex $\mathcal{H}$ is. This is done via the VC dimension.*

Remark. *The VC dimension, as many other dimensions, can be defined via the following question:*

what is the largest "fully" complicated part that $\mathcal{H}$ contains?

Notation. *Functions can be projected. Given $T \subseteq X$ and $f : X \to \{0,1\}$, denote by $f|_T$ the projection of f to T. It is the function $T \to \{0,1\}$ that agrees with f.*

Notation. *Classes can be projected. Given $T \subseteq X$, let*

$$\mathcal{H}|_S = \{f|_T : f \in \mathcal{H}\} \subseteq \{0,1\}^T.$$

Definition. *A subset T of X is called shattered by $\mathcal{H}$ if*

$$\mathcal{H}|_T = \{0,1\}^T.$$

Remark. *In other words, every function on T can be realized by a function in $\mathcal{H}$.*

Remark. *The set $\{0,1\}^T$ is the "most complicated" set of functions on T.*

Example. *For LTFs in the plane, the VC dimension is three. There are 3 shattered points, and no four points are shattered (sketch picture).*

Definition. *The VC dimension of $\mathcal{H}$ is the maximum size of a set that is shattered by $\mathcal{H}$. It could be infinite.*

Remark. *Roughly speaking, if the VC dimension is d then there are d points in which "we have no context".*

Theorem 1. *The sample complexity of PAC learning $\mathcal{H}$ is equal, up to universal constants, to the VC dimension of $\mathcal{H}$.*

Remark. *This theorem is sometimes called the fundamental theorem of statistical learning theory.*

Remark. *The theorem includes two statements (where d is the VC dimension).*

- *The first is that $o(d)$ samples do not suffices for PAC learning.*
- *The second is that $O(d)$ samples suffices.*

The first was proved by Blumer, Ehrenfeucht, Haussler and Warmuth in 1989. The second was proved by Vapnik and Chervonenkis in 1971. The first is left as an exercise and we shall prove the second.

Remark. *There are other important notions of "dimension" that appear in the theory of ML (we shall not cover here).*

3. ERMs

There are algorithms and there are algorithmic principles. It is important to understand the properties of algorithmic principles because this can help guide our choices. A natural algorithmic principle in ML is Empirical Risk Minimization (ERM). It has pros and cons.

Definition. *The empirical loss of* $h : X \to \{0,1\}$ *on the sample* $S = ((x_1, y_1), \ldots, (x_n, y_n))$ *is*

$$L_S(h) = \frac{1}{n} \sum_i 1_{h(x_i) \neq y_i}.$$

Remark. *The expected value of the empirical loss is the true loss*

$$L_\mu(h) = \mathbb{E}_{S \sim \mu^n} L_S(h)$$

Remark. *When we get the input sample* S*, for each* $h \in \mathcal{H}$*, we can compute the empirical loss*

$$L_S(h).$$

For some h*'s, this loss is high, so it is natural to "remove" them. For some* h*'s, this loss is small or even zero. But there could be many such* h*'s. Which* h *should we choose?*

Remark. *The ERM principle says that we may choose any* h *that minimizes* $L_S(\cdot)$*. Amazingly, in the PAC model, this principle leads to good results.*

Theorem 2. *Let* $\mathcal{H}$ *be of VC dimension* d*, and let* $n = 100d$*. For each sample* S *of size* n*, let* $h_S \in \mathcal{H}$ *be a function that minimizes* L_S*. Then, for all* $\mathcal{H}$*-realizable* μ*,*

$$\mathbb{E}_{S \sim \mu^n} L_\mu(h_S) < \frac{1}{3}.$$

Remark. *One reason this is not trivial is that* h_S *depends on* S*.*

4. Structure

The analysis of ERMs as well as many other important results in PAC learning relies on a basic combinatorial statement (known as the Sauer-Shelah-Perles lemma).

Lemma 3. *If* $\mathcal{H} \subset \{0,1\}^X$ *has VC dimension* d *and* $|X| = k$ *then*

$$|\mathcal{H}| \leq \sum_{i=0}^{d} \binom{k}{i}.$$

Remark. *It is often helpful to find the simplest expression even if it is only approximately correct:*

$$\sum_{i=0}^{d} \binom{k}{i} \leq \left(\frac{ek}{d}\right)^d.$$

This say that roughly speaking the size of $\mathcal{H}$ is k^d which is much smaller than 2^k. Indeed, because $1 - x \le e^x$ for all $x \in \mathbb{R}$,

$$\begin{aligned}\Big(\frac{d}{k}\Big)^d \sum_{i=0}^{d} \binom{k}{i} &\le \sum_{i=0}^{d} \Big(\frac{d}{k}\Big)^i \binom{k}{i} \\ &\le \sum_{i=0}^{n} \Big(\frac{d}{k}\Big)^i 1^{n-i} \binom{k}{i} \\ &= \Big(1 + \frac{d}{k}\Big)^k \le e^d.\end{aligned}$$

Remark. *There are many proof of this lemma: by induction, using shifting, using algebra, etc. We shall not prove due to lack of time.*

Remark. *The lemma leads to the following dichotomy. For every class of functions $\mathcal{H}$, exactly one of the following holds:*

— *there are larger and larger sets $T \subseteq X$ so that*

$$|\mathcal{H}|_T| = 2^{|T|}$$

— *there is d so that for all T,*

$$|\mathcal{H}|_T| \le (|T| + 1)^d$$

In other words, either there is exponential growth or polynomial growth. There is nothing in between.

Remark. *On a high-level, if the VC dimension of $\mathcal{H}$ is d then the projection of $\mathcal{H}$ to sets of size $n = 100d$ is of size $\approx n^d \ll 2^d$. VC found a beautiful argument for proving that this property suffices to show that ERM is well-behaved.*

Remark. *This idea is called "double sampling". Here is a sketch. Let $S \sim \mu^n$. We want to upper bound the chance of the "bad" event*

$$B = \{\exists h \in \mathcal{H} \ |L_S(h) - L_\mu(h)| > \frac{1}{10}\}.$$

This is when ERMs might fail. To analyze this, imagine sampling another $S' \sim \mu^n$ independently of S. It can be shown (and makes intuitive sense) that

$$\Pr[B] \le 2 \Pr\Big[\exists h \in \mathcal{H} \ |L_S(h) - L_\mu(h)| > \frac{1}{10}, |L_{S'}(h) - L_\mu(h)| < \frac{1}{20}\Big].$$

Now, we can think of first taking a sample of size $2n$ and randomly partitioning it to two parts. We can upper bound the chance by an expression of the form:

$$\Big(\frac{e \cdot 2n}{d}\Big)^d 2^{-cn}.$$

This first term is the total number of function for a fixed choice of $2n$ values of x, and is based on the SSP lemma. The second term is the chance that a random partition to two parts is "highly unbalanced".