PAC Learning of Threshold Classifiers

1 Problem Setting

We consider threshold classifiers on $\mathbb{R}$. Let the true (unknown) target hypothesis be

$$h^*(x) = \mathbf{1}[x < a^*],$$

where $a^* \in \mathbb{R}$ is the true threshold.

Our goal is to learn a hypothesis h_{b_S} such that

$$\mathbb{P}_{x\sim D}\big[h_{b_S}(x) \neq h_{a^*}(x)\big] \leq \varepsilon,$$

i.e., the misclassification probability is at most ε. This can be interpreted as: *out of 100 students, at most* $\varepsilon \cdot 100$ *are misclassified.*

2 Risk Regions Around the True Threshold

Given $\varepsilon > 0$, define two points $a_0 < a^* < a_1$ such that

$$\mathbb{P}(x \in (a_0, a^*)) = \varepsilon, \qquad \mathbb{P}(x \in (a^*, a_1)) = \varepsilon.$$

These intervals represent the *risk regions*:

- Left risk region (a_0, a^*) of probability mass ε
- Right risk region (a^*, a_1) of probability mass ε

Errors occurring inside these regions are tolerated.

3 Sampling

We draw an i.i.d. sample

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \sim D^m.$$

4 Empirical Risk Minimization (ERM)

Define

$$b_0 = \max\{x : (x, 1) \in S\}, \qquad b_1 = \min\{x : (x, 0) \in S\}.$$

The ERM algorithm selects any threshold

$$b_S \in (b_0, b_1).$$

5 Failure Condition

The learned hypothesis has excess error if

$$L_D(h_{b_S}) > \varepsilon.$$

This can occur only if

$$b_0 < a_0 \quad \text{or} \quad b_1 > a_1.$$

Interpretation:

- No sample point falls in the left risk region, or
- No sample point falls in the right risk region.

6 Union Bound

Using the union bound,

$$\mathbb{P}\big[L_D(h_{b_S}) > \varepsilon\big] \leq \mathbb{P}[b_0 < a_0] + \mathbb{P}[b_1 > a_1]. \tag{$\star$}$$

7 Bounding Each Term

Since $\mathbb{P}(x \in (a_0, a^*)) = \varepsilon$,

$$\mathbb{P}[b_0 < a_0] = \mathbb{P}\big[\forall i,\ x_i \notin (a_0, a^*)\big] = (1-\varepsilon)^m \leq e^{-\varepsilon m}.$$

Similarly,

$$\mathbb{P}[b_1 > a_1] \leq e^{-\varepsilon m}.$$

8 Final Bound

Combining the above,

$$\mathbb{P}\big[L_D(h_{b_S}) > \varepsilon\big] \leq 2e^{-\varepsilon m}.$$

To ensure this probability is at most δ, it suffices that

$$2e^{-\varepsilon m} \leq \delta,$$

which yields the sample complexity bound

$$m \geq \frac{\log(2/\delta)}{\varepsilon}.$$

9 Key Insight

Even if $\mathcal{H}$ is infinite, PAC learning is possible if $\mathbb{P}$[samples hit the risk regions] $\to 1$.

10 Why VC-Dimension?

This section aims to answer a single fundamental question:

> Why are some hypothesis classes learnable, while others are not?

The precise criterion that separates the two is the *VC-dimension*.

10.1 Motivation: No-Free-Lunch

The No-Free-Lunch theorem states that if no restriction is imposed on the hypothesis class, then no learning algorithm can perform well in all cases.

The reason is simple:

- Any labeling of the data can be explained,
- Hence no explanation carries real predictive power.

In other words:

> A theory that can explain everything, in fact explains nothing.

10.2 What Changes in PAC Learning

PAC learning introduces a crucial restriction:

> The data must be perfectly realizable by some hypothesis in $\mathcal{H}$.

That is, the target hypothesis h^* is assumed to belong to $\mathcal{H}$.

This shifts the question to:

> Even under this restriction, can an adversary still force learning to fail?

10.3 Adversarial Perspective

For an adversary to break learning, it must be able to:

- Choose a finite set of points C,
- Assign labels to these points arbitrarily,
- While remaining consistent with some hypothesis in $\mathcal{H}$.

This motivates studying how flexibly $\mathcal{H}$ can behave on finite sets.

10.4 Definition 6.2 (Restriction to a Finite Set)

Let $C = \{c_1, c_2, \ldots, c_m\} \subseteq \mathcal{X}$ be a finite set of points.

We consider the restriction of $\mathcal{H}$ to C:

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

This is the collection of all labelings that hypotheses in $\mathcal{H}$ can induce on C.

10.5 Definition 6.3 (Shattering)

The hypothesis class $\mathcal{H}$ is said to *shatter* C if

$$\mathcal{H}_C = \{0,1\}^m.$$

That is, for every possible labeling of the points in C, there exists a hypothesis in $\mathcal{H}$ that realizes it.

Intuition. On the set C, $\mathcal{H}$ can explain *any* pattern whatsoever.

10.6 Example: Threshold Classifiers

Consider threshold hypotheses on $\mathbb{R}$.

One point. For a single point c_1, both labels 0 and 1 are achievable by placing the threshold appropriately. Hence, one point can be shattered.

Two points. Let $c_1 \leq c_2$. The labeling $(0,1)$ is impossible, since threshold classifiers preserve order. Thus, two points cannot be shattered.

Therefore, the VC-dimension of threshold classifiers is 1.

10.7 Corollary 6.4 (Why Shattering Is Dangerous)

If $\mathcal{H}$ can shatter $2m$ points, then no algorithm can learn $\mathcal{H}$ using only m samples.

Reason. Given only m labeled examples from a shattered set of size $2m$, the labels of the remaining m points are completely unconstrained. Every possible labeling is consistent with some hypothesis in $\mathcal{H}$.

Thus, the learner has no information about half of the domain.

10.8 Definition 6.5 (VC-Dimension)

The VC-dimension of $\mathcal{H}$ is defined as

$$\mathrm{VC}(\mathcal{H}) = \max\{|C| : C \text{ is shattered by } \mathcal{H}\}.$$

10.9 Theorem 6.6 (One Direction)

If $\mathrm{VC}(\mathcal{H}) = \infty$, then $\mathcal{H}$ is not PAC learnable.

Reason. For any sample size m, the adversary can select a shattered set of size $2m$, rendering learning impossible.

10.10 Summary

> VC-dimension measures how many points a hypothesis class can label arbitrarily. If it can lie too well, learning is impossible.