COMP 585 **Spring 2026**
Probabilistic Toolkit for Learning and Computing *Instructor: Maryam Aliakbarpour*
March 26, 2026 *Scriber: Maryam Aliakbarpour*

Lecture 19

PAC Learnability of Boolean Conjunctions

In this lecture, we prove that the class of Boolean conjunctions is PAC-learnable in the realizable setting by analyzing a explicit algorithm.

Boolean conjunctions

We now study a concrete concept class over the Boolean hypercube $\mathcal{X} = \{0,1\}^n$. For each coordinate $i \in [n]$, we have two literals: the positive literal x_i and the negative literal $\bar{x}_i$, where $\bar{x}_i = 1 - x_i$. Let $\mathcal{L} \coloneqq \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ be the set of all $2n$ literals. For any subset $S \subseteq \mathcal{L}$, define the hypothesis

$$h_S(x) \coloneqq \bigwedge_{\ell \in S} \ell(x). \tag{1}$$

Intuitively, a conjunction (also known as logical and) acts as a strict "all-or-nothing" checklist. The hypothesis $h_S(x)$ evaluates to 1 if and only if every single literal in the set S is satisfied simultaneously; if even one condition fails, the entire function outputs 0. We let

$$\mathcal{C}_{\text{conj}} \coloneqq \{h_S : S \subseteq \mathcal{L}\} \tag{2}$$

be the class of all Boolean conjunctions of literals. This definition includes the empty conjunction, which is the constant-1 function, and also inconsistent conjunctions such as $x_i \wedge \bar{x}_i$, which represent the constant-0 function.

Example 1. If $n = 3$ and $S = \{x_1, \bar{x}_2\}$, then

$$h_S(x_1, x_2, x_3) = x_1 \wedge \bar{x}_2 \,.$$

Hence

$$h_S(1,0,1) = 1, \qquad h_S(0,0,1) = 0 \,.$$

A learning algorithm for conjunctions

We assume the realizable setting: there exists an unknown target conjunction $h^\star = h_{S^\star} \in \mathcal{C}_{\text{conj}}$, and the training examples are drawn from a distribution D on $\mathcal{X} \times \{0,1\}$ satisfying

$$\forall (x,y) \sim D : \quad y = h^*(x)\,.$$

The algorithm starts from the most restrictive conjunction, namely the conjunction of all literals, and then deletes literals that are inconsistent with positive examples.

1. Initialize $S_0 = \mathcal{L}$.
2. Process the samples one by one. If the current sample is $(x_j, 0)$, do nothing. If the current sample is $(x_j, 1)$, delete from the current set every literal ℓ such that $\ell(x_j) = 0$.
3. After all m samples have been processed, let $\widehat{S}$ be the set of literals that remain, and output

$$\widehat{h} = h_{\widehat{S}}. \tag{3}$$

For example, suppose the current conjunction is $x_2 \wedge x_3 \wedge \bar{x}_4$ and we observe a positive example $x = (1,0,1,0)$. Since $x_2(x) = 0$, the literal x_2 is inconsistent with this positive example and must be deleted. The new conjunction becomes $x_3 \wedge \bar{x}_4$.

Theorem 2 (PAC guarantee for conjunctions)**.** *Let $\mathcal{C}_{\text{conj}}$ be the class defined in (2). For every $\epsilon, \delta \in (0,1)$, if*

$$m \geq \left\lceil \frac{2n}{\epsilon} \log\!\left(\frac{2n}{\delta}\right) \right\rceil, \tag{4}$$

then the algorithm above outputs a hypothesis $\widehat{h} \in \mathcal{C}_{\text{conj}}$ satisfying

$$\mathbf{Pr}_{T \sim D^m}\Big[\, err_D(\widehat{h}) \leq \epsilon \Big] \geq 1 - \delta.$$

Consequently, $\mathcal{C}_{\text{conj}}$ is PAC-learnable, with sample complexity

$$m_{\text{PAC},\mathcal{C}_{\text{conj}}}(\epsilon, \delta) = O\Big(\frac{n}{\epsilon} \log\Big(\frac{n}{\delta}\Big)\Big)\,.$$

Proof. Fix $\epsilon, \delta \in (0,1)$. Let $S^\star \subseteq \mathcal{L}$ be such that $h^\star = h_{S^\star}$ is the target conjunction.

We divide the proof into four steps.

Step 1: Target literals are never deleted. Consider any literal $\ell \in S^\star$. Let $(x_j, 1)$ be any positive sample. Since the labels are realizable and $h^\star(x_j) = 1$, every literal in the conjunction $h^\star$ must evaluate to 1 on x_j. In particular, we must have $\ell(x_j) = 1$. Therefore

the deletion rule can never remove a target literal. Since the algorithm starts from $S_0 = \mathcal{L}$, after all samples are processed we still have

$$S^\star \subseteq \widehat{S}. \tag{5}$$

Step 2: The output makes no false positive errors. We claim that if $h^\star(x) = 0$, then necessarily $\widehat{h}(x) = 0$. Indeed, if $h^\star(x) = 0$, then by the definition of conjunction there exists some literal $\ell^\star \in S^\star$ such that $\ell^\star(x) = 0$. By (5), the literal $\ell^\star$ also belongs to $\widehat{S}$. Hence the conjunction defining $\widehat{h}$ contains a literal that is zero at x, so

$$\widehat{h}(x) = 0\,.$$

Thus

$$\{x \in \mathcal{X} : \widehat{h}(x) = 1\} \subseteq \{x \in \mathcal{X} : h^\star(x) = 1\}.$$

Equivalently, every error made by $\widehat{h}$ is a false negative: the algorithm may predict 0 when the true label is 1, but it never predicts 1 when the true label is 0.

Step 3: If no bad literal survives, then the error is at most ϵ. For each literal $\ell \in \mathcal{L}$, define

$$p(\ell) \coloneqq \mathbf{Pr}_{x\sim D_{\mathcal{X}}}[h^\star(x) = 1 \text{ and } \ell(x) = 0]. \tag{6}$$

This is exactly the probability that a random example is positive for the target concept and simultaneously witnesses that the literal ℓ is inconsistent with the target. That is, if $\ell \in \widehat{h}$, then $\widehat{h}(x)$ will be zero erroneously.

Now fix any $x \in \mathcal{X}$ such that $h^\star(x) = 1$ and $\widehat{h}(x) = 0$. Since $\widehat{h}$ is the conjunction of the literals in $\widehat{S}$, there must exist at least one literal $\ell \in \widehat{S}$ with $\ell(x) = 0$. Therefore

$$\{x \in \mathcal{X} : h^\star(x) = 1 \text{ and } \widehat{h}(x) = 0\} \subseteq \bigcup_{\ell\in\widehat{S}} \{x \in \mathcal{X} : h^\star(x) = 1 \text{ and } \ell(x) = 0\}\,.$$

Using Step 2 and the union bound, we obtain

$$\begin{aligned} \mathrm{err}_D(\widehat{h}) &= \mathbf{Pr}_{x\sim D_{\mathcal{X}}}\Big[h^\star(x) = 1 \text{ and } \widehat{h}(x) = 0\Big] \\ &\le \sum_{\ell\in\widehat{S}} \mathbf{Pr}_{x\sim D_{\mathcal{X}}}[h^\star(x) = 1 \text{ and } \ell(x) = 0] \\ &= \sum_{\ell\in\widehat{S}} p(\ell)\,. \end{aligned} \tag{7}$$

We call a literal $\ell \in \mathcal{L}$ *bad* if

$$p(\ell) \geq \frac{\epsilon}{2n}\,.$$

Suppose no bad literal survives, that is, every literal in $\widehat{S}$ satisfies $p(\ell) < \epsilon/(2n)$. Since $\widehat{S} \subseteq \mathcal{L}$ and $|\mathcal{L}| = 2n$, Equation (7) yields

$$\begin{aligned}\mathrm{err}_D(\widehat{h}) &\leq \sum_{\ell \in \widehat{S}} p(\ell) \\ &< \left|\widehat{S}\right| \cdot \frac{\epsilon}{2n} \\ &\leq 2n \cdot \frac{\epsilon}{2n} = \epsilon\,.\end{aligned}$$

Thus, if $\mathrm{err}_D(\widehat{h}) > \epsilon$, then at least one bad literal must survive in $\widehat{S}$.

Step 4: The probability that a bad literal survives is exponentially small. Fix a bad literal $\ell \in \mathcal{L}$. On a single sample $(x^{(j)}, y^{(j)}) \sim D$, the algorithm deletes ℓ if and only if $y^{(j)} = 1$ and $\ell(x^{(j)}) = 0$. Because the setting is realizable, $y^{(j)} = 1$ is equivalent to $h^\star(x^{(j)}) = 1$. Hence, by the definition of $p(\ell)$,

$$\mathbf{Pr}_{(x_j,y_j)\sim D}\big[\ell \text{ is deleted by sample } j\big] = p(\ell)\,.$$

Therefore

$$\mathbf{Pr}_{(x_j,y_j)\sim D}\big[\ell \text{ survives sample } j\big] = 1 - p(\ell)\,.$$

Since the samples are independent, the probability that ℓ survives all m samples is

$$\begin{aligned}\mathbf{Pr}_{T\sim D^m}\big[\ell \text{ survives all } m \text{ samples}\big] &= \big(1 - p(\ell)\big)^m && (8)\\ &\leq \exp\big(-mp(\ell)\big) && (9)\\ &\leq \exp\Big(-\frac{m\epsilon}{2n}\Big)\,, && (10)\end{aligned}$$

where (9) uses the elementary inequality $1 - u \leq e^{-u}$ for all $u \in [0, \infty)$, and (10) uses that ℓ is bad. Let $\mathcal{L}_{\text{bad}} \subseteq \mathcal{L}$ denote the set of bad literals. Since $\mathcal{L}$ contains $2n$ literals, we have $|\mathcal{L}_{\text{bad}}| \leq 2n$. Hence, by the conclusion of Step 3 and another union bound,

$$\begin{aligned}\mathbf{Pr}_{T\sim D^m}\Big[\mathrm{err}_D(\widehat{h}) > \epsilon\Big] &\le \mathbf{Pr}_{T\sim D^m}[\exists \text{ a bad literal that survives}] \\ &\le \sum_{\ell\in\mathcal{L}_{\text{bad}}} \mathbf{Pr}_{T\sim D^m}[\ell \text{ survives all } m \text{ samples}] \\ &\le |\mathcal{L}_{\text{bad}}| \cdot \exp\Big(-\frac{m\epsilon}{2n}\Big) \\ &\le 2n \cdot \exp\Big(-\frac{m\epsilon}{2n}\Big)\ .\end{aligned}$$

Thus it is sufficient to choose m so that

$$2n \cdot \exp\Big(-\frac{m\epsilon}{2n}\Big) \le \delta. \tag{11}$$

Taking logarithms, (11) is implied by

$$m \ge \frac{2n}{\epsilon} \ln\Big(\frac{2n}{\delta}\Big)\,. \tag{12}$$

This is exactly (4). Therefore

$$\mathbf{Pr}_{T\sim D^m}\Big[\mathrm{err}_D(\widehat{h}) \le \epsilon\Big] \ge 1-\delta,$$

as required. □

Bibliographic Note

Valiant's landmark paper introduced the Probably Approximately Correct (PAC) learning model, providing a rigorous mathematical framework for computational learning theory [Val84]. In this work, Valiant demonstrates that the class of conjunctions (monomials) is PAC-learnable using a simple elimination algorithm. Although this class contains exponentially many possible concepts, he proved that it can be learned in a computationally efficient manner. His results show that by allowing for a small margin of error and a failure probability, we can achieve efficient learning within reasonable polynomial time.

References

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.