PAC (Take I) Learning: Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is probably approximately learnable if there exists a learning algorithm L with the following property: For every target concept $c \in \mathcal{C}$, for every distribution D over X, for all $0 < \varepsilon < 1/2$, $0 < \delta < 1/2$ if L is given access to the example oracle $EX(c, D)$ and inputs ε, δ, then with probability at least $1-\delta$, L outputs a hypothesis $h \in \mathcal{C}$ satisfying $err(h) \leq \varepsilon$.

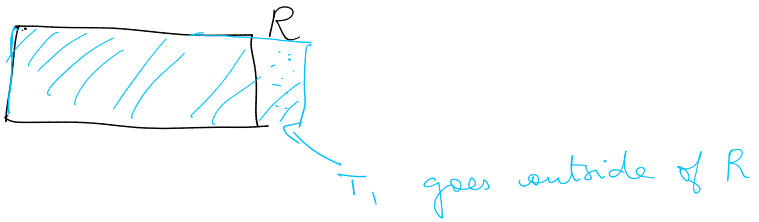
- The probability is taken over random draws from $EX(c, D)$ as well as any internal randomization of L.
- If L runs in time polynomial in $1/\varepsilon, 1/\delta$, we say that $\mathcal{C}$ is efficiently PAC learnable

---

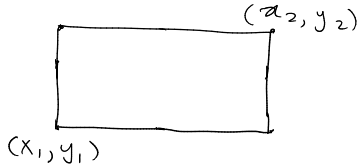
Theorem: The concept class $\mathcal{C}$ of axis-aligned rectangles in $\mathbb{R}^2$ is efficiently PAC learnable

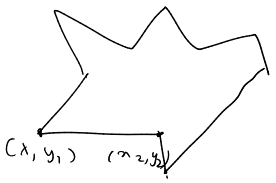
---

- We saw a proof of the above last time, with some "missing" details.
- What if D is not "smooth" enough?
- what if $\mathbb{P}_{x \sim D}[x \in R] < \varepsilon/4$, where R is the target rectangle, in that case T_i is not contained in R!



Representation size





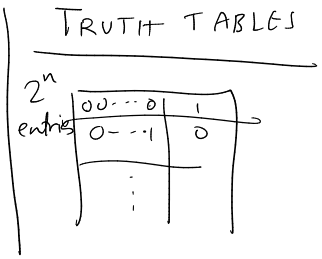
"complexity of the target function/ concept class!

"Can store real numbers at unit cost"

Boolean Functions:

Let $X = \{0,1\}^n$

$f: X \to \{0,1\}$



Every boolean function requires space 2^n.

- Boolean circuits, AND, OR and NOT gates.
- Disjunctive Normal Form (DNF)

$f \equiv (x_1 \wedge \bar{x}_3 \wedge \cdots \wedge x_9) \vee (x_3 \wedge \bar{x}_7 \wedge \cdots) \vee (x_1 \wedge x_{10} \wedge \cdots)$

$\{0,1\}^n$

$f = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ } PARITY FUNCTION — 1 if odd no. of x_i are 1, 0 otherwise.

- Circuit of depth $\log(n)$ and size $O(n \log n)$
- Need a DNF formula of size $2^{\Omega(n)}$ to represent parity.

- How many boolean functions? 2^{2^n}
- We will use representation to restrict the class of Boolean functions that we will aim to learn!
  - → Decision trees
  - → Neural networks

Representation Scheme:

A representation scheme for a concept class C is a mapping $R: \Sigma^* \to C$, where Σ^* is a finite alphabet.

($R: (\Sigma \cup \mathbb{R})^* \to C$ if we need real numbers).

$size: \Sigma^* \to \mathbb{N}$ (length of the string)

$size(c) = \min_{R(\sigma)=c} \{size(\sigma)\}$

Instance Size:

Typically we will consider $X = \{0,1\}^n$, $X = \mathbb{R}^n$, and "n" will be the size parameter with the instance space.

$X = \langle X_n \rangle_{n \geq 1}$, $X = \bigcup_{n \geq 1} X_n$ where X_n represents instances of size n.

C_n is a concept class over X_n,

$C = \bigcup_{n \geq 1} C_n$ a concept class over X.

PAC Learning (Take II): Let C_n be a representation/concept class over X_n. Let $C = \bigcup_{n \geq 1} C_n$, & $X = \bigcup_{n \geq 1} X_n$. We say C is PAC learnable if there exists a learning algorithm L, s.t. $\forall n \in \mathbb{N}$, $\forall c \in C_n$, $\forall$ distribution D over X_n, $\forall 0 < \varepsilon < 1/2$, $\forall 0 < \delta < 1/2$ if L is given access to the example oracle $EX(c,D)$ and inputs ε, δ, $size(c)$, outputs w.p. $\geq 1-\delta$, a hypothesis $h \in C_n$ satisfying $err(h) \leq \varepsilon$. (Exercise: Drop this)

(Probability over all sources of randomness)

C is efficiently PAC learnable if the running time of L is polynomial in n, $size(c)$, $1/\varepsilon$, $1/\delta$.

Example: Learning Boolean Functions

$X = \{0,1\}^n$ $\quad z_i$: boolean variable

$\underline{x}$

CONJUNCTION : $z_1 \wedge \bar{z}_3 \wedge z_7 \wedge \dots \wedge z_{91}$

$f : X \to \{0,1\}$

C_n : The class of all conjunctions on n boolean variables!

$|C_n| = 3^n$ $\quad$ $\underline{n}$

$\forall c \in C_n, \text{size}(c) \leq 2n$ $\quad$ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔

Q: Can we design an algorithm that learns conjunctions in time polynomial $n, 1/\varepsilon, 1/\delta$?

Examples: $(\underline{x}_1, y_1), (\underline{x}_2, y_2), \dots, (\underline{x}_m, y_m)$

$\underline{x}_i \in \{0,1\}^n, \quad y_i \in \{0,1\}$

$\exists$ conjunction c, st. $c(\underline{x}_i) = y_i$

$(1\,1\,0\,1\,1\,1,\ 1), (1\,0\,0\,1\,1\,0,\ 0), \dots$)

ALGORITHM:

$h = z_1 \wedge \bar{z}_1 \wedge z_2 \wedge \bar{z}_2 \wedge \dots \wedge z_n \wedge \bar{z}_n$

for $i = 1, \dots, m$, {

if ($y_i = 1$) {

we drop each $\bar{z}_j$ from h if $\underline{x}_{i,j} = 1$

" " " z_j from h if $\underline{x}_{i,j} = 0$

} }

Return h

Remarks: • Algorithm only uses positive examples.

• if $h(\underline{x}) = 1$ then $c(\underline{x}) = 1$

Let's look at a literal ℓ,

$p(\ell) = \underset{\underline{x} \sim D}{\mathbb{P}} [\, c(\underline{x}) = 1 \wedge$ the literal ℓ evaluates to 0 in $\underline{x}\,]$

ℓ is "bad" if $p(\ell) \geq \varepsilon/2n$.

A_ℓ : Event that after drawing m examples the literal ℓ is not eliminated from h.

$$\mathbb{P}(A_\ell) \le \left(1 - \frac{\varepsilon}{2n}\right)^m$$

$$\mathbb{P}(\text{err}(h) \ge \varepsilon) \le \mathbb{P}\left(\bigcup_{\substack{\ell \text{ bad} \\ \text{literal}}} A_\ell\right) \le \sum_{\substack{\ell \text{ bad} \\ \text{liter}}} \mathbb{P}(A_\ell) \le 2n \cdot \left(1 - \frac{\varepsilon}{2n}\right)^m \le \delta$$

WANT

If we eliminate all the bad literals

$$\text{err}(h) \le \sum_{\substack{\ell \text{ "good"} \\ \text{literals}}} p(\ell) \le 2n \cdot \frac{\varepsilon}{2n} = \varepsilon.$$

Confidence Analysis:

want $2n \cdot \left(1 - \frac{\varepsilon}{2n}\right)^m \le \delta$.

Sufficient if $2n \cdot e^{-\frac{m\varepsilon}{2n}} \le \delta$, $\quad \left(1 - x \le e^{-x}\right)$

" " $m \ge \frac{2n}{\varepsilon} \log\left(\frac{2n}{\delta}\right)$. □

Theorem: The class of conjunctions is efficiently PAC (Take II) learnable.