Computational learning theory

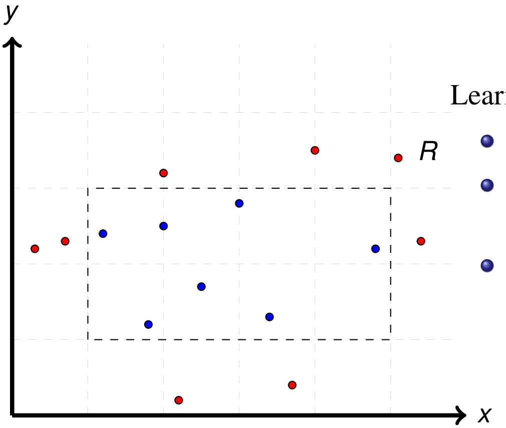
PAC model

Jan Otop

October 6, 2023

Motivating example

Learning axis-aligned rectangles.



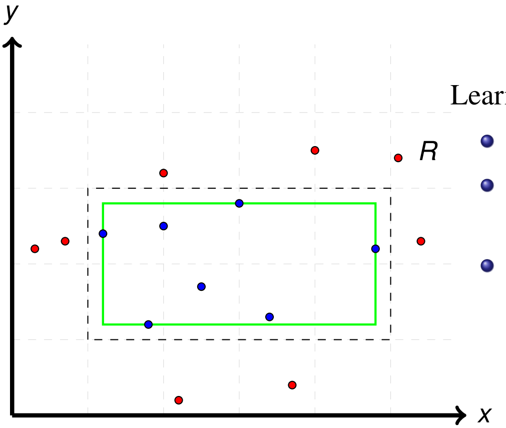
Learning framework:

- stochastic: points drawn i.i.d.
- probably: $\mathbb{P}(S) > 1 - \delta$
- approximately: $\mathbb{P}(\{p \mid p \in R \oplus R'\}) \leq \epsilon$

Can we learn a rectangle given ϵ, δ? $\Rightarrow$ blackboard

Motivating example

Learning axis-aligned rectangles.



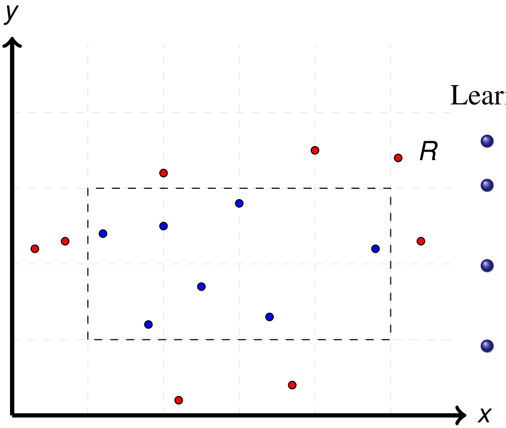
Learning framework:

- stochastic: points drawn i.i.d.
- probably:
  $\mathbb{P}(S) > 1 - \delta$
- approximately:
  $\mathbb{P}(\{p \mid p \in R \oplus R'\}) \leq \epsilon$

Can we learn a rectangle given ϵ, δ? $\Rightarrow$ blackboard

Motivating example

Learning axis-aligned rectangles.



Learning framework:

- stochastic: points drawn i.i.d.
- probably:
  $\mathbb{P}(S) > 1 - \delta$
- approximately:
  $\mathbb{P}(\{p \mid p \in R \oplus R'\}) \leq \epsilon$
- **Distribution independent.**

Can we learn a rectangle given ϵ, δ? $\Rightarrow$ blackboard

Terminology

- Instance space: X
- Concept $\mathbf{c}\colon X \to \{0,1\}$
- Concept and Hypothesis classes $\mathcal{C}, \mathcal{H} \subseteq 2^X$.
- Empirical error (risk) (w.r.t. a concept $\mathbf{c}$ and a sample S):

$$\widehat{\mathrm{err}}_S(\mathbf{h}) = \frac{1}{|S|} \sum_{x \in S} \mathbf{1}_{\mathbf{c}(x) \neq \mathbf{h}(x)} = \frac{|\{x \in S \mid \mathbf{c}(x) \neq \mathbf{h}(x)\}|}{|S|}$$

- Generalization error (risk) (w.r.t. a concept $\mathbf{c}$):

$$\mathrm{err}(\mathbf{h}) = \mathbb{E}_{x \sim D}(\mathbf{1}_{\mathbf{c}(x) \neq \mathbf{h}(x)}) = \mathbb{P}_{x \sim D}(\{x \mid \mathbf{c}(x) \neq \mathbf{h}(x)\})$$

PAC definition

Leslie Valiant. 1984.

Probabilistically approximately correct (PAC) learning Ver 1

A class C over X is (efficiently) PAC-learnable if
there is an algorithm that for every concept $\mathbf{c} \in C$ and distribution D over X:

- **Parameter Input**: $\epsilon, \delta \in \mathbb{Q}^+$.
- **Draws random samples**: $x_1, \ldots, x_m$ independently with probability distribution D
- **Number of samples**: m is polynomially bounded in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$
- works in polynomial time in m
- **Output**: a hypothesis $\mathbf{h} \in C$ such that:
  - with probability $1 - \delta$ (**probabilistically**)
  - $\mathrm{err}(\mathbf{h}) \leq \epsilon$ (**approximately**)

$$\mathbb{P}_{S \sim D^m}(\{S \mid \mathrm{err}(\mathbf{h}) \leq \epsilon\}) \geq 1 - \delta$$

PAC definition

Leslie Valiant. 1984.

Probabilistically approximately correct (PAC) learning Ver 2

A class C over X is (efficiently) PAC-learnable if
there is an algorithm that for every concept $\mathbf{c} \in C$ and distribution D over X:

- **Parameter Input**: $\epsilon, \delta \in \mathbb{Q}^+$.
- **Draws random samples**: $x_1, \ldots, x_m$ independently with probability distribution D
- **Number of samples**: m is polynomially bounded in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$
- works in polynomial time in m, $size(c)$ and $rep(X)$
- **Output**: a hypothesis $\mathbf{h} \in C$ such that:
  - with probability $1 - \delta$ (**probabilistically**)
  - $\mathrm{err}(\mathbf{h}) \leq \epsilon$ (**approximately**)

$$\mathbb{P}_{S \sim D^m}(\{S \mid \mathrm{err}(\mathbf{h}) \leq \epsilon\}) \geq 1 - \delta$$

PAC definition

Leslie Valiant. 1984.

Probabilistically approximately correct (PAC) learning Ver 3

A class C over X is (efficiently) PAC-learnable using $\mathcal{H}$ if
there is an algorithm that for every concept $\mathbf{c} \in C$ and distribution D over X:

- **Parameter Input**: $\epsilon, \delta \in \mathbb{Q}^+$.
- **Draws random samples**: $x_1, \ldots, x_m$ independently with probability distribution D
- **Number of samples**: m is polynomially bounded in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$
- works in polynomial time in m, $size(c)$ and $rep(X)$
- **Output**: a hypothesis $\mathbf{h} \in \mathcal{H}$ such that:
  - with probability $1 - \delta$ (**probabilistically**)
  - $\mathrm{err}(\mathbf{h}) \leq \epsilon$ (**approximately**)

$$\mathbb{P}_{S \sim D^m}(\{S \mid \mathrm{err}(\mathbf{h}) \leq \epsilon\}) \geq 1 - \delta$$

PAC definition

Leslie Valiant. 1984.

Probabilistically approximately correct (PAC) learning   Ver 1

A class C over X is (efficiently) PAC-learnable if
there is an algorithm that for every concept $\mathbf{c} \in C$ and distribution D over X:

- **Parameter Input**: $\epsilon, \delta \in \mathbb{Q}^+$.
- **Draws random samples**: $x_1, \ldots, x_m$ independently with probability distribution D
- **Number of samples**: m is polynomially bounded in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$
- works in polynomial time in m
- **Output**: a hypothesis $\mathbf{h} \in C$ such that:
  - with probability $1 - \delta$ (**probabilistically**)
  - $\text{err}(\mathbf{h}) \leq \epsilon$ (**approximately**)

$$\mathbb{P}_{S \sim D^m}(\{S \mid \text{err}(\mathbf{h}) \leq \epsilon\}) \geq 1 - \delta$$

Today's lecture

- Learning finite concept classes. (Occam's Razor)
- Example: learning conjunctions

Learning finite hypothesis classes

Fitting algorithms

A *fitting algorithm* for C, gets a labeled sample S as an input and returns $\mathbf{c} \in C$ consistent with S or return NO.

The derived decision question is called *the consistency problem*.

Learning finite hypothesis classes

Fitting algorithms

A *fitting algorithm* for C, gets a labeled sample S as an input and returns $\mathbf{c} \in C$ consistent with S or return NO.

The derived decision question is called *the consistency problem*.

Learning finite classes — Occam's Razor

Let C be a finite class over X. Assume that C has a polynomial-time fitting algorithm A. Then, for any $\epsilon, \delta \in \mathbb{Q}^+$, if

$$m \geq \frac{1}{\epsilon}\left(\log(|C|) + \log\left(\frac{1}{\delta}\right)\right)$$

then with probability at least $1 - \delta$ the hypothesis $\mathbf{h}$ returned by A satisfies

$$\mathrm{err}(\mathbf{h}) \leq \epsilon.$$

Learning conjunctions

- $X = \{0,1\}^n$ is the set of n-variable Boolean assignments.
- C concepts defined by conjunctions of variables $x_1, \ldots, x_n$. E.g. $x_1 \wedge \neg x_2 \wedge x_4$.
- $|C| \leq 3^n + 1$.

Polynomial consistency algorithm

1. Start with the maximal conjunction

$$x_1 \wedge \neg x_1 \wedge \cdots \wedge x_n \wedge \neg x_n$$

2. For every positive example $\sigma \in X$, remove all literals conflicting with σ:
   If $\sigma(x_i) = 1$, then $\neg x_i$ is in conflict with σ.
   If $\sigma(x_i) = 0$, then x_i is in conflict with σ.

The resulting conjunction is **maximal**.