"Proof" Thm 7.2

Claim: $\mathcal{H}$ nonuniform learnable with A, $m^{NUL}_{\mathcal{H}} \Rightarrow \mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$, each $\mathcal{H}_n$ is agnostic PAC learnable

Recall Corollary 6.4 (NFL)

$\mathcal{X}$ domain, $VCdim(\mathcal{H}) \geq 2n$ $(\Rightarrow \exists C \subseteq \mathcal{X}, |C| = 2n, \mathcal{H}$ shatters $C)$

$\Rightarrow \forall$ learning algorithms A, $\exists \mathcal{D}: \exists h \in \mathcal{H}: L_{\mathcal{D}}(h) = 0$ with prob $\geq \frac{1}{7}$ over $S \sim \mathcal{D}^n$: $L_{\mathcal{D}}(A(S)) \geq 1/8$

[Holds also for A which learn one an $\mathcal{H}' \supseteq \mathcal{H}$]

Proof: $\mathcal{H}_n = \{h \in \mathcal{H} : m^{NUL}_{\mathcal{H}}(0.1, 0.1, h) \leq n\}$

$\mathcal{D}$ realizable wrt. $\mathcal{H}_n$ $\quad S \sim \mathcal{D}^n$

$\forall h \in \mathcal{H}_n$: $L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(h) + 0.1$

— " — $\leq \underbrace{\min_{h \in \mathcal{H}_n} L_{\mathcal{D}}(h)}_{=0} + 0.1 = 0.1 \qquad L_{\mathcal{D}}(A(S)) \leq 0.1$

Assume $VCdim(\mathcal{H}_n) \geq 2n$

So apply Cor. 6.4 $\forall$ learning alg A', $\exists \mathcal{D}$ with realiz. wrt. $\mathcal{H}_n$ s.t. with prob $\geq 1/7$ over $S \sim \mathcal{D}^n$ $L_{\mathcal{D}}(A'(S)) \geq \frac{1}{8}$

↯ It should hold for A, as well

$\Rightarrow VCdim(\mathcal{H}_n) < 2n$

$\overset{\text{Fund. Thm}}{\Rightarrow}$ $\mathcal{H}_n$ is agnostic PAC-learnable

---

Recall SRM: $\operatorname{argmin}_{h \in \mathcal{H}} L_S(h) + \varepsilon_{n(h)}(m, w(n(h)) \delta)$

$n(h) := \min\{n \in \mathbb{N} : h \in \mathcal{H}_n\}$

$w(n)$ weight fct. $\sum_{n=1}^{\infty} w(n) \leq 1$

$\varepsilon_n(m, \delta) = \min\{\varepsilon \in (0,1) : m^{UC}_{\mathcal{H}_n}(\varepsilon, \delta) \leq m\}$

MDL and Occam's Razor (as a special case of SRM)

$\mathcal{H}$ countable $\Rightarrow \bigcup_{n \in \mathbb{N}} \{h_n\} = \mathcal{H}$

$m^{UC}_{\{h_n\}} \overset{\text{finite}}{\leq} \frac{\log\left(\frac{2}{\delta}|\{h_n\}|\right)}{2\varepsilon^2} = \frac{\log(2/\delta)}{2\varepsilon^2}$

$\varepsilon_n(m, \delta) = \sqrt{\frac{\log(2/\delta)}{2m}}$, $n(h) = n$

$$\boxed{\text{SRM becomes } \operatorname{argmin}_{h \in \mathcal{H}} L_S(h) + \sqrt{\frac{-\log(w(n)) + \log(\frac{2}{\delta})}{2m}} = \text{"} \quad + \sqrt{\frac{-\log(w(h)) + \log(\frac{2}{\delta})}{2m}}}$$

Want to use a w based on a description language:

Σ: finite alphabet (e.g. $\{0,1\}$)

Σ^*: all finite strings over Σ

$d: \mathcal{H} \to \Sigma^*$ description language

Focus on prefix-free languages:

$\forall h \neq h' \in \mathcal{H}$: $d(h)$ is not a prefix of $d(h')$

Kraft inequality: $\mathcal{S} \subseteq \{0,1\}^*$ prefix free:

$$\sum_{\sigma \in \mathcal{S}} \frac{1}{2^{|\sigma|}} \leq 1$$

Proof: Draw 0,1 uniformly at random (and stop if it equals a string in $\mathcal{S}$)

$\forall \sigma \in \mathcal{S}$: $P(\sigma) = \frac{1}{2^{|\sigma|}}$

$$\sum_{\sigma \in \mathcal{S}} \frac{1}{2^{|\sigma|}} = P(\mathcal{S}) \leq 1$$

We can use $d(h)$ as a weight $w(h) = \frac{1}{2^{|h|}}$

$$\boxed{\text{MDL: } \operatorname{argmin}_{h \in \mathcal{H}} L_S(h) + \sqrt{\frac{|h| + \ln(2/\delta)}{2m}}}$$

tradeoff between emp. risk at "complexity" of describing h

Occam's Razor:

"A short explanation tends to be more valid than a long one"

Consistency

Algorithm A is consistent wrt. $\mathcal{H}$ and $\mathcal{P}$ (= set of distributions)

if: $\exists\, m^{CON}_{\mathcal{H}}\ (0,1)^2 \times \mathcal{H} \times \mathcal{P} \to \mathbb{N}$ s.t.

$\forall \varepsilon, \delta, \forall h \in \mathcal{H}, \forall \mathcal{D} \in \mathcal{P}$: $L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(h) + \varepsilon$

Example: MEMORIZE is consistent

Given a test instance x, MEMORIZE returns the majority of the labels of instances in the sample S, which are equal to x. (just predict the majority of all labels if there is no x)

$\Rightarrow$ This notion is to weak to capture "learning"

Comparison

	Bounds on true error by the emp. risk	How many exampl. are needed to be as best as any hypo. in $\mathcal{H}$	Encode prior knowledge
(agn) PAC	✓	(in advance) ✓	✓ (specify an $\mathcal{H}$)
Nonuniform	(Thm 7.4) ✓	✓ depends on the best $h \in \mathcal{H}$	✓ (weights)
Consistent	X	X	X

Runtime of Algorithms

Input size?

e.g. sample size is a bad idea.

$\to$ depend on ε and δ

Computational Complexity for Learning Algorithms

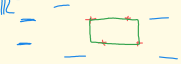
- A solves a learning task $(Z, \mathcal{H}, l)$ in time $\mathcal{O}(f(\varepsilon, \delta))$, if:
  - A terminates in $\mathcal{O}(f(\varepsilon, \delta))$ time.
  - output of A should applicable to (new) instances in $\mathcal{O}(f(\varepsilon, \delta))$ time.
  - agn. PAC learn $(Z, \mathcal{H}, l)$

A solves a sequence of learning problems $(Z_n, \mathcal{H}_n, l_n)_{n=1}^{\infty}$ in $\mathcal{O}(f_n(\varepsilon, \delta))$ if:

for each fixed n, A solves $(Z_n, \mathcal{H}_n, l_n)$ in time $\mathcal{O}(f_n(\varepsilon, \delta))$ (efficient if $f_n = \mathcal{O}(poly(\frac{1}{\varepsilon}, \frac{1}{\delta}, n))$)

Example: Rectangles (realizable) in $\mathbb{R}^n$

$\mathbb{R}^2$



$S \sim \mathcal{D}^m$

For each dim: find min and max in $\mathcal{O}(m)$ time

Total time $\mathcal{O}(nm) = \mathcal{O}(n\, m(\varepsilon, \delta))$

Rectangle (Agnostic)

It's NP-hard to compute the ERM-rectangle.

One can learn it in $\mathcal{O}(m^{\mathcal{O}(n)})$.