The Machine Learning prob

$f : \mathcal{X} \to \mathcal{Y}$

unknown func; $\mathcal{X}$ = input; $\mathcal{Y}$ = output

unknown input distribution $P(x)$

$x_1, x_2, \ldots, x_N \overset{\text{i.i.d.}}{\sim} P(x)$

$x \sim P$

Training examples

$\{(x_1, y_1), \ldots, (x_N, y_N)\}$

$y_n = f(x_n)$

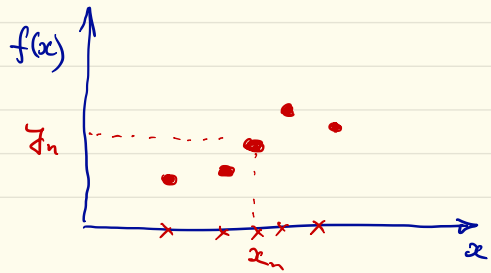
Learning algorithm $\mathcal{A}$

Hypothesis set $\mathcal{H} = \{h_\theta : \theta \text{ parameters}\}$

ex: linear predictor, deep neural net } θ

Learned hypothesis $g^{(\mathcal{D})}$

loss $(g(x), f(x))$ — true



Traing data $\mathcal{D} = \{ (x_1, y_1), \dots, (x_N, y_N) \}$

A hypothesis h: $h : \mathcal{X} \to \mathcal{Y}$, $h \in \mathcal{H}$

In-sample error (traing error)

(*) $E_{in}(h) = \frac{1}{N} \sum_{n=1}^{N} loss(h(x_n) ; f(x_n))$

Out-of-sample error:

(*) $E_{out}(h) = \underset{x \sim P}{\mathbb{E}} \left[loss(h(x) ; f(x)) \right]$

Training (learning): $\mathcal{D} \xrightarrow{\mathcal{A}} \mathcal{H}$ $\Rightarrow$ $g^{(\mathcal{D})}$

$g^{(\mathcal{D})} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} E_{in}(h)$

Evaluate $E_{out}(h) = \frac{1}{M} \sum_{m=1}^{M} loss(h(z_m), f(z_m))$

Law of large number

$P \to z_1, z_2, \dots, z_M$ test set

approx.

Key: Can NOT use $\mathcal{D} = \{(x_n, y_n)\}$ to evaluate $E_{out}(g^{(\mathcal{D})})$

$$T = \{ (x_n, f(x_n)) \}_{n=1}^{N}$$

Test set: ①

like to evaluate $g^{(D)}$

$$E_{out}(g^{(D)}) \approx \frac{1}{N} \sum_{n=1}^{N} \underbrace{loss(g^{(D)}(x_n), f(x_n))}_{\ell(x_n)}$$

② T cannot be contaminated by D (training set)

③ How large should T be?

$$\underbrace{E_{x \sim P}[\ell(x)]}_{\mu} \approx \underbrace{\frac{1}{N} \sum_{n=1}^{N} \ell(x_n)}_{}$$

$$\rightarrow P[|\cdot - \mu| \leq \text{tol}] \geq 1 - \frac{1}{a^2}$$

Trade off: