$D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\} \subseteq X \times Y$

$X = \mathbb{R}^d$

$Y = \{0, 1\}$ ← binary classification

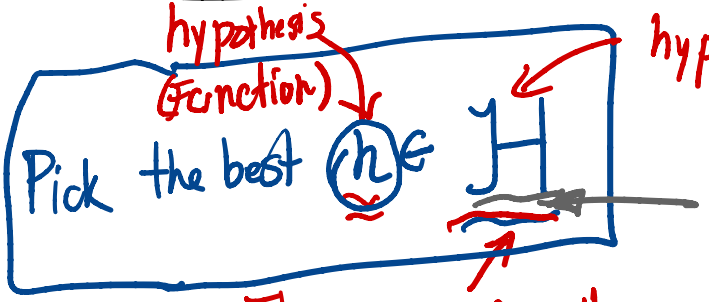
$Y = \{0, 1, \ldots, k-1\}$ ← multi-class classification

$Y = \mathbb{R}$ ← regression

Assume each $(\vec{x}_i, y_i) \in D$ is i.i.d. and is drawn from distribution P $((\vec{x}_i, y_i) \sim P)$.

Goal: Learn some function $h: X \to Y$

such that $h(\vec{x}) = y$ for any $(\vec{x}, y) \sim P$



IDEAL: We wish to find the function h that makes the fewest mistakes

Loss functions $L(\underset{\sim}{h}, D)$

(percentage error)

- 0/1 Loss $L_{0/1}(\underset{\sim}{h}, D) = \frac{1}{|D|} \sum_{\forall (\vec{x}, y) \in D} d(\underset{\sim}{h}(\vec{x}), y)$,

Binary classification:

+ Multi-class classification

where $d(h(\vec{x}), y) = \begin{cases} 0 & \text{if } h(\vec{x}) = y \\ 1 & \text{if } h(\vec{x}) \neq y \end{cases}$

- Square Loss $L_{sq}(h, D) = \frac{1}{|D|} \sum_{\forall (\vec{x}, y) \in D} (h(\vec{x}) - y)^2$

Regression:

Absolute loss $L_{ABS}(h, D) = \frac{1}{|D|} \sum_{\forall (\vec{x}, y) \in D} (|h(\vec{x}) - y|)$

An ML algorithm is to choose $h^* \in \mathcal{H}$ that minimizes the loss function

(local loss)

$$h^* = \arg\min_{h \in \mathcal{H}} L(h, D) =$$

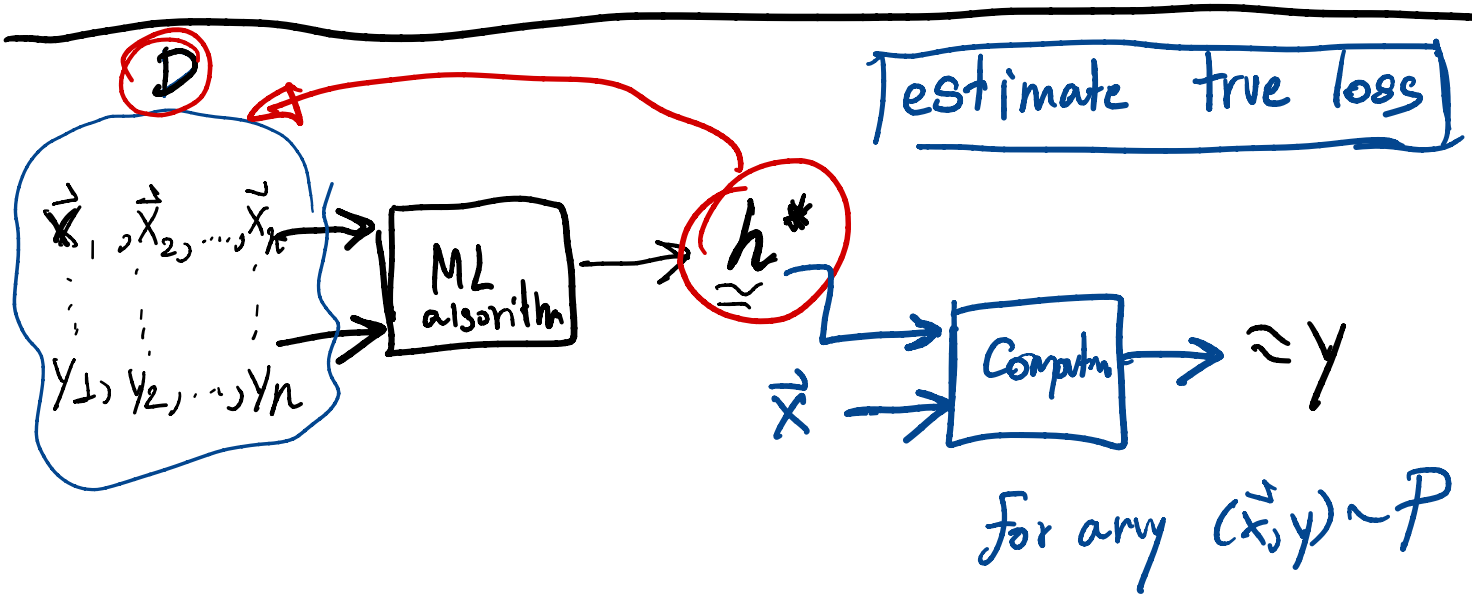
Exercise: Suppose we have the following h

$$h(\vec{x}) = \begin{cases} y_i, & \text{if } \vec{x} = \vec{x}_i \text{ s.t. } \exists (\vec{x}_i, y_i) \in D \\ 0, & \text{otherwise} \end{cases}$$

Overfitting

100% D

Overfitting: We have the function h that works well on our data set but cannot be generalized to general data



True loss / Generalization loss: The true goal of learning

is to minimize $\mathcal{E}_{true} = E_{(\vec{x},y)\sim P}[L(h\{(\vec{x},y)\})]$

True goal: To learn h^* s.t.

$$h^* = \arg\min_{h \in \mathcal{H}} E_{(\vec{x},y)\sim P}[L(h\{(\vec{x},y)\})]$$

(Global loss)

Unknown

Local loss: $L(h, D)$

ML alg. tries to minimize

Global loss: $\varepsilon = E[L(h, \{(\vec{x}, y)\})]$

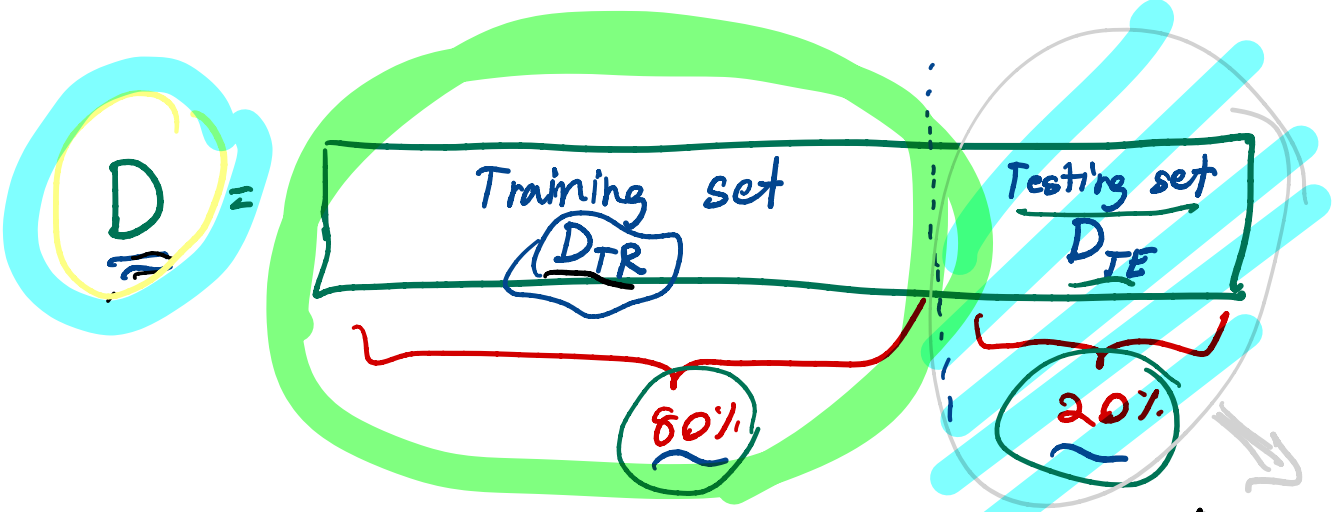
$(\vec{x}, y) \sim P$

X

Our expectation towards the entire distribution

$$h(\vec{x}) \approx y \quad \forall (\vec{x}, y) \sim P$$

D represents the whole distribution P.



- ML algs. pick h^* s.t.

$$h^* = \text{argmin}_{h \in \mathcal{H}} L(h, D_{TR})$$

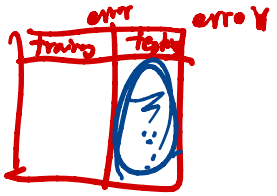
- Evaluation of testing loss

$$\varepsilon_{TE} = L(h^*, D_{TE})$$

Find loss that estimates the global loss!!

$$\varepsilon_{TE} \approx \varepsilon$$

$D = D_{TE} \cup D_{TE}$ 0/1 90%



h

Claim: As $|D_{TE}| \to +\infty$, $\varepsilon_{TE} \to \varepsilon$.

Proof idea: $\lim_{|D_{TE}| \to +\infty} L(h, D_{TE}) = E[L(h, \{(\vec{x}, y)\})]$

$(\vec{x}, y) \sim P$

$(\vec{x}, y)$ ϵ

Therefore, ε_{TE} is an unbias estimator of ε.

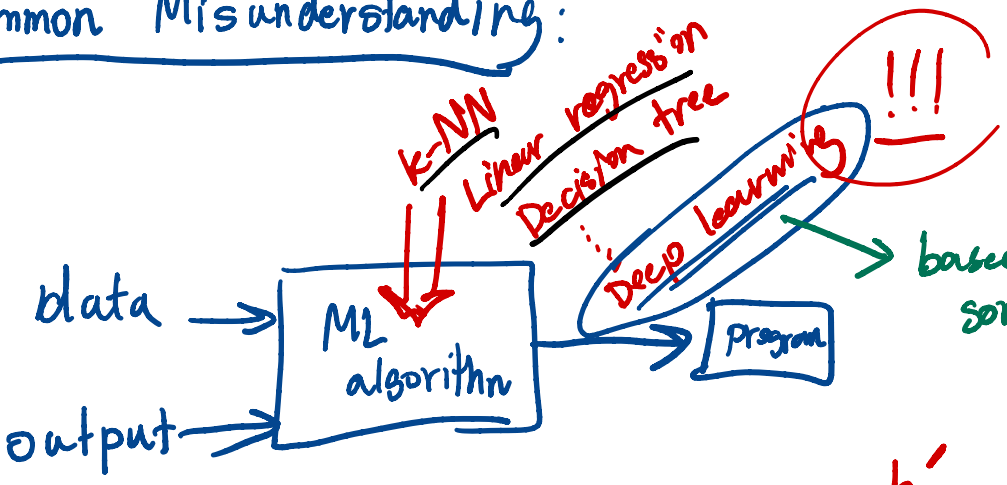
The weak law of large number.

Summary: Supervised learning in practice.

1. Define the setup
2. Collect data
3. Train the program via training data
   (choosing appropriate ML alg.)
4. Test the program via testing data.
   (measure program's accuracy.)
5. The testing loss/error will indicate the actual accuracy of the program.

Common Misunderstanding:

There is no a universal ML alg.

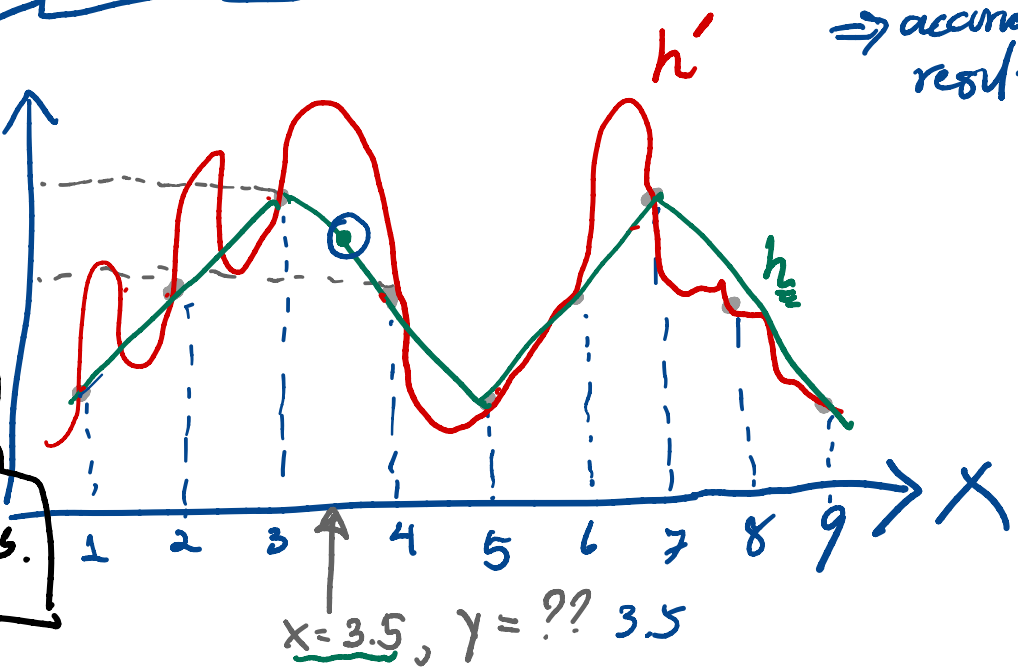


based on some assumption X about data

⇒ accurate results

No Free Lunch Theorem:

Every successful ML alg. must make assumptions.



x = 3.5, y = ?? 3.5

$$L(h_3, D) = 0^2 + 0^2 + \underset{36}{(12-8)^2} + \underset{36}{(12-4)^2} + \underset{16}{(8-4)^2}$$

$$= 36 + 36 + 16$$

$x=4 \rightarrow y=12 \mid h_3(4) = 8$

$x=8 \rightarrow y=12 \mid h_3(4) = 6$ = 88

$x=12 \rightarrow y=8 \mid h_3(12) = 4$