Lecture 23 :

$S = \{(x_i, y_i)\}_{i=1}^{n}$ → learner →

↓

ultimate goal:

$$\min_h \Pr_{(x,y)\sim D}\{h(x) \neq y\}$$

↓

the first relaxation:

$$\min_h \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h(x_i) \neq y_i\} \longrightarrow \text{bad}$$

$= 0$

↓

restricted ERM_H

$$\min_{h \in H} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h(x_i) \neq y_i\}$$

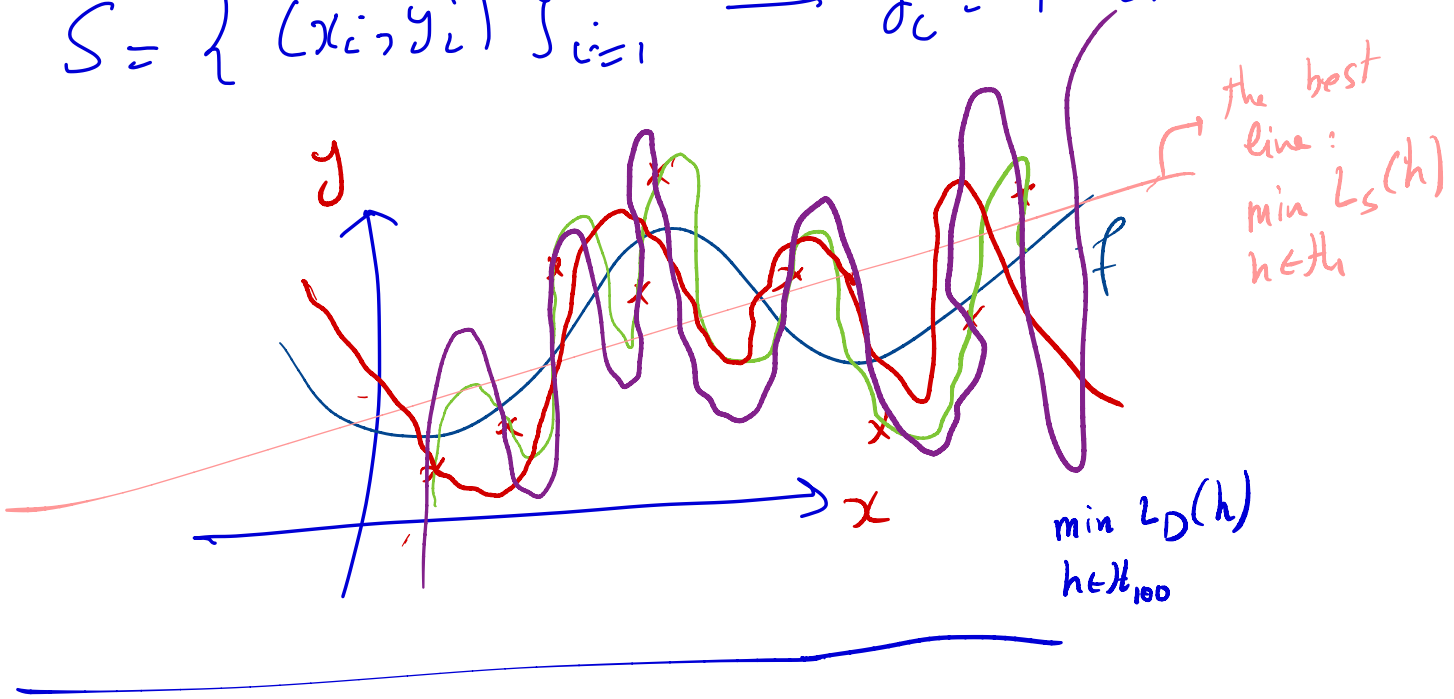
↙

the predictor is inside the class H.

Example:

$$y = f(x) + \varepsilon \qquad E[\varepsilon] = 0, \quad \mathrm{var}[\varepsilon] = \sigma^2$$

$$S = \{(x_i, y_i)\}_{i=1}^{n} \longrightarrow y_i = f(x_i) + \varepsilon_i$$



$$L_D(h) = \underset{(x,y)\sim D}{E}\left[(h(x) - y)^2 \right]$$

$$\min_h L_D(h) \Longrightarrow \text{the minimizer is } f$$

$$\rightarrow \min_{h \in \mathcal{H}} L_D(h) = E\left[(y - f(x))^2\right]$$

$$= E[\varepsilon^2] = \sigma^2$$

training error

$$\min_h L_S(h) = 0$$

$$\min_h L_D(h) = 6^2$$

$$\min_h L_S(h) = 0$$

$$\min_{h \in \mathcal{H}} L_S(h)$$

examples for $\mathcal{H}$:

$\mathcal{H}_1 = \{$ all the linear functions
i.e. polynomials of degree 1 $\}$

↳ since $\mathcal{H}_1$ is not rich enough, we will have underfitting

⇒ Hence the function class $\mathcal{H}$ that we're going to use, should be sufficiently rich / complex.

$H_6 = \{$ all the polynomials of degree 6 $\}$

$\Rightarrow$ the problem with using rich/complex function classes is: overfitting

In summary: we have 2 problems:

Problem 1:

if H is not rich enough, it'll underfit:

$$\min_{h \in H} L_D(h) \gg \min_{h} L_D(h)$$

Solution: use a class H that's sufficiantly rich.

Problem 2:

if H is complex, then it may suffer from overfitting:

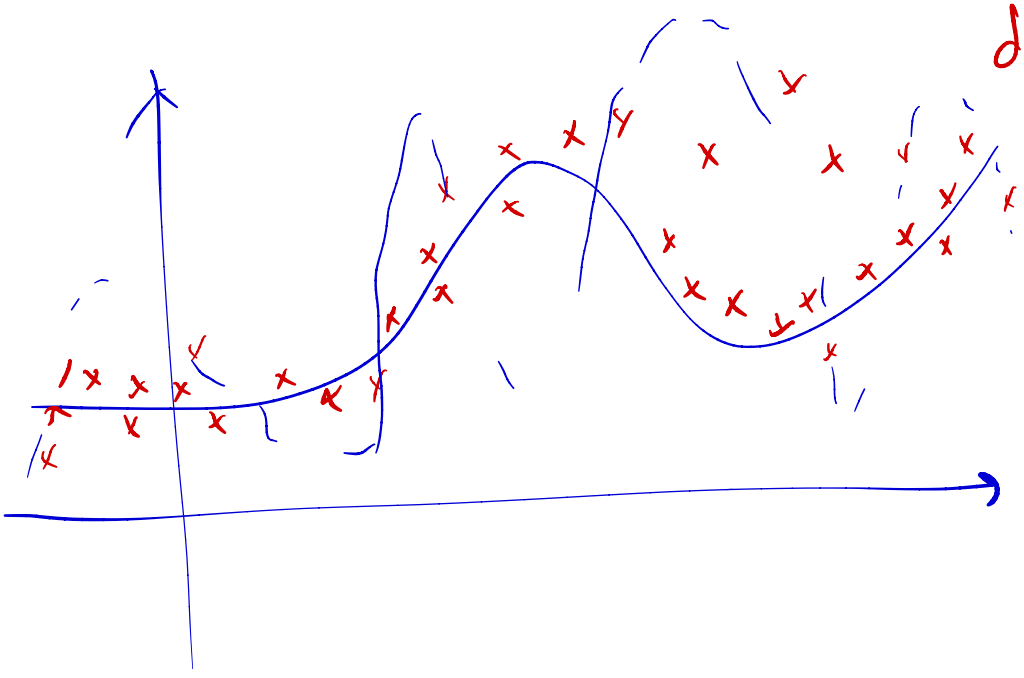
$$\min_{h \in H} L_D(h) \approx \min_{h} L_D(h)$$

$$\min_{h \in H} \underbrace{L_S(h)}_{\text{traing error}} \longrightarrow h_S^* \longrightarrow L_S(h_S^*) \approx 0$$

H: very rich/complex

$$\min_h L_D(h) \ll L_D(h_S^*)$$

true error of what we learned from fitting the function class H to the training data S.



$$L_S(h) \longrightarrow L_D(h)$$

$$\min_{h \in H} L_S(h) \longrightarrow \min_{h \in H} L_D(h)$$

$\Longrightarrow$ to avoid overfitting we should have a sufficient number of training data points.

for every: $h \in H \longrightarrow L_S(h) \approx L_D(h) \pm \varepsilon$

$$\min_{h \in H} L_S(h) \approx \min_{h \in H} L_D(h) \pm \varepsilon$$

It seems that to avoid overfitting the number of training data points should be sufficiently large

↳ - depends on H

- depends on the accuracy ε

PAC learning (PAC = Probably Approximately Correct)

We say that a function class $\mathcal{H}$ is PAC-learnable if for any $\epsilon, \delta > 0$, there exists a number $n_0(\epsilon, \delta)$, such that for any distribution of data, D, the following holds: If $n \geq n_0(\epsilon, \delta)$, then with probability $1-\delta$ we have that:

$$\forall h \in \mathcal{H} : \left| L_D(h) - L_S(h) \right| \leq \epsilon.$$

$$\min_{h \in \mathcal{H}} L_S(h) \approx \min_{h \in \mathcal{H}} L_D(h)$$