Exercise

Your friend has developed a new machine learning algorithm for binary classification (i.e., $y \in \{-1, 1\}$) with 0-1 loss and tells you that it achieves a generalization error of only 0.05. However, when you look at the learning problem he is working on, you find out that $\Pr_{\mathcal{D}}[y = 1] = 0.95$...

- Assume that $\Pr_{\mathcal{D}}[y = \ell] = p_\ell$. Derive the generalization error of the (dumb) hypothesis/model that *always* predicts ℓ.
- Use the result above to decide if your friend's algorithm has learned something or not.

Solution

1) We are considering the hypothesis/model: $h(\vec{x}) = \ell \quad \forall \vec{x} \in \vec{X}$.

Generalization error of h: z

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(\vec{x},y)\sim\mathcal{D}} \left[\ell\left(h, \overbrace{(\vec{x}, y)}^{z}\right) \right]$$

by def. of $L_{\mathcal{D}}(h)$

by def. of $\mathbb{E}[\,]$ $\rightarrow$

$$= 0 \cdot \Pr_{(\vec{x},y)\sim\mathcal{D}} \left[\ell(h, (\vec{x}, y)) = 0 \right] + 1 \cdot \Pr_{(\vec{x},y)\sim\mathcal{D}} \left[\ell(h, (\vec{x}, y)) = 1 \right]$$

$$= Pr_{(\vec{x},y)\sim D}\left[\ell(h_1(\vec{x},y)) = 1\right]$$

by def. of 0-1 loss $\rightarrow = Pr_{(\vec{x},y)\sim D}\left[h(\vec{x}) \neq y\right]$

by def. of h $\rightarrow = Pr_{(\vec{x},y)\sim D}\left[y \neq \ell\right]$

by def. of prob. $\rightarrow = 1 - Pr_{(\vec{x},y)\sim D}\left[y = \ell\right]$

by def. of p_ℓ $\rightarrow = 1 - p_\ell$

2) From point 1) above, the hypothesis $h(\vec{x}) = 1 \; \forall \vec{x} \in X$ has generalization error equal to 0.05. The "dumb" model ($h(\vec{x}) = 1 \; \forall \vec{x} \in X$) has generalization error as good as your friend's algorithm

$\Rightarrow$ your friend's algorithm has not learned a relation between $\vec{x}$ and y, it has "learned" that

$$\Pr_{(\vec{x},y)\sim D}[y=1] = 0.95.$$

(Such probability is trivial to "learn" with enough data)

$\Rightarrow$ your friend's algorithm has not learned something useful (in terms of ML tasks).

Exercise

Assume we have the following training set S, where $\mathbf{x} = [x_1, x_2] \in \mathbb{R}^2$ and $\mathcal{Y} = \{-1, 1\}$:

$S = \{([-3, 4], 1), ([2, -3], -1), ([-3, -4], -1), ([1, 1.5], 1)\}$.

Assume you decide to use $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$ with

$h_1 = sign(-x_1 - x_2)$ $\quad 1 + 1 + 1 + 1 \Longrightarrow L_S(h_1) = 4/4 = 1.$

$h_2 = sign(-x_1 + x_2)$ $\quad 0 + 0 + 0 + 0 \Longrightarrow L_S(h_2) = 0$

$h_3 = sign(x_1 - x_2)$ $\quad 1 + 1 + 1 + 1 \Longrightarrow L_S(h_3) = 1$

$h_4 = sign(x_1 + x_2)$ $\quad 0 + 0 + 0 + 0 \Longrightarrow L_S(h_4) = 0$

Your algorithm uses the ERM rule and the 0-1 loss.

- What model h_S is produced in output by your ML algorithm?
- Assume the realizability assumption holds. What can you say about the generalization error $L_{\mathcal{D}}(h_S)$ of h_S?

Solution

2) compute the empirical/training error for h_1, h_2, h_3, h_4.
Report in output on of the hypotheses minimizing the training error.
Let's compute the training error $L_S(h_i)$, for $i = 1, \dots, 4$

i	$L_S(h_i)$
1	1
2	0
3	1
4	0

$\Rightarrow$ Your machine learning algorithm will report in output one between h_2 and h_4

2) We know $L_S(h_S) = 0$.

From the proof of Corollary 1.3 we know that:

$$\Pr\left[L_D(h_S) > \varepsilon\right] \leq \underbrace{|\mathcal{H}| e^{-\varepsilon m}}_{\delta}$$

, where m = # of samples in the training set S.

$\Rightarrow$ for a given δ, fix $\varepsilon = \frac{1}{m} \ln\left(\frac{|\mathcal{H}|}{\delta}\right)$

$$\Rightarrow \Pr\left[L_D(h_S) \leq \frac{1}{m} \ln\left(\frac{|\mathcal{H}|}{\delta}\right)\right] \geq 1 - \delta$$

For example, let's choose $\delta = 0.1$; we have that with probability ≥ 0.9, the following holds:

$$L_D(h_S) \leq \frac{1}{4} \ln\left(\frac{4}{0.1}\right) \approx 0.75$$

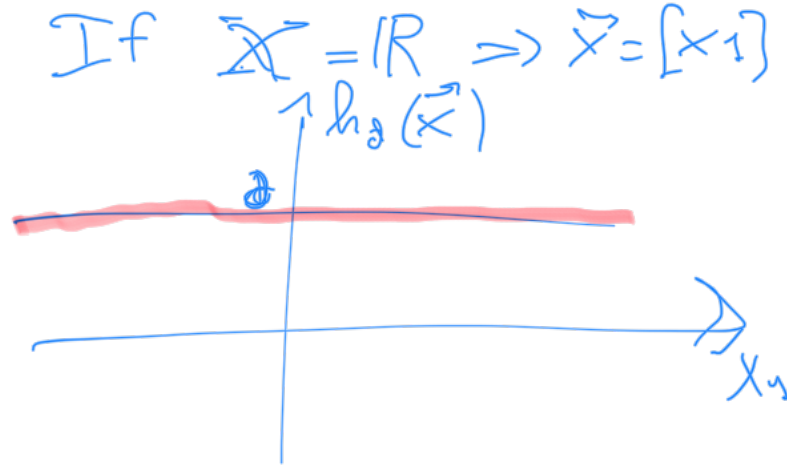
Exercise

Consider a linear regression problem, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$, with mean squared loss. The hypothesis set is the set of *constant* functions, that is $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$, where $h_a(\mathbf{x}) = a$. Let $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ denote the training set.

- Derive the hypothesis $h \in \mathcal{H}$ that minimizes the training error.
- Use the result above to explain why, for a given hypothesis $\hat{h}$ from the set of all linear models, the coefficient of determination $R^2 = 1 - \frac{\sum_{i=1}^{m}(\hat{h}(\mathbf{x}_i) - y_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$ where $\bar{y}$ is the average of the $y_i, i = 1, \ldots, m$ is a measure of how well $\hat{h}$ performs (on the training set).

$\mathcal{H} = \{h_a : a \in \mathbb{R}\}$

$h_a(\vec{x}) = a \quad \forall \vec{x} \in \mathbb{R}^d$



Solution

•) Given $h_a \in \mathcal{H}$, the training error for such hypothesis is:

$$L_S(h_a) = \frac{1}{m} \sum_{i=1}^{m} \left(h_a(\vec{x}_i) - y_i \right)^2 = \frac{1}{m} \sum_{i=1}^{m} \left(a - y_i \right)^2$$

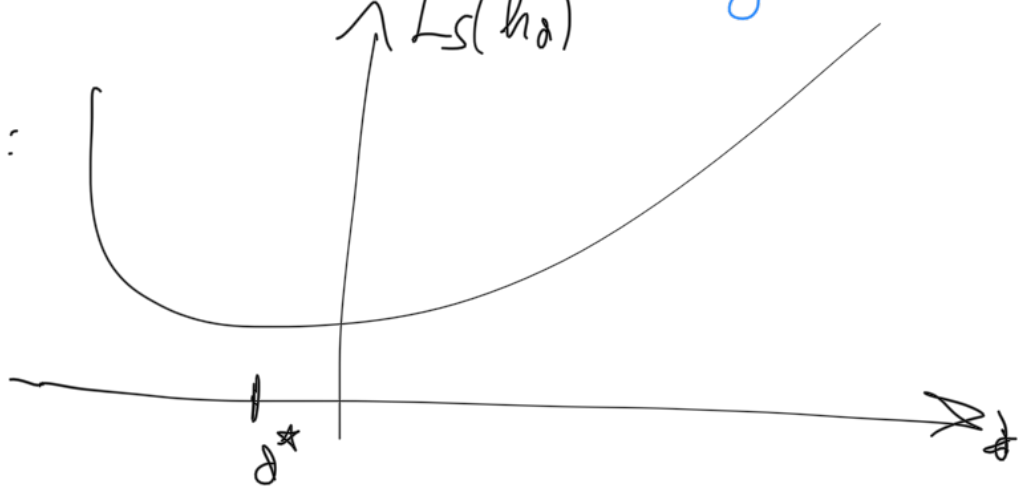
↳ since $h_a(\vec{x}_i) = a \ \forall \vec{x}_i$

Now, finding $h_a \in \mathcal{H}$ that minimizes the training error corresponds to find a that minimizes:

$$L_S(h_a) = \frac{1}{m} \sum_{i=1}^{m} \left(a - y_i \right)^2 = f(a) = (\ )a^2 + (\ )a + (\ )$$

$\vee$ 0

As a function of a:



$\Rightarrow$ compute $\frac{d L_S(h_\theta)}{d\theta}$ and find θ s.t. $\frac{d L_S(h_\theta)}{d\theta} = 0$.

$$\frac{d L_S(h_\theta)}{d\theta} = \frac{d}{d\theta}\left(\frac{1}{m}\cdot\sum_{i=1}^{m}(\theta - y_i)^2\right)$$

$$= \frac{1}{m}\frac{d}{d\theta}\left(\sum_{i=1}^{m}(\theta - y_i)^2\right)$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{d}{d\theta}(\theta - y_i)^2\right)$$

$$= \frac{1}{m}\sum_{i=1}^{m}(2\theta - 2y_i)$$

$$= \frac{2}{m}\sum_{i=1}^{m}(\theta - y_i)$$

$(\theta - y_i)^2 = \theta^2 - 2\theta y_i + y_i^2$

$$\frac{2}{m}\sum_{i=1}^{m}(\theta - y_i) = 0 \iff \sum_{i=1}^{m}(\theta - y_i) = 0$$

$$\iff \left(\sum_{i=1}^{m}\theta\right) - \left(\sum_{i=1}^{m} y_i\right) = 0$$

$$\Leftrightarrow \left(\sum_{i=1}^{m} a\right) = \left(\sum_{i=1}^{m} y_i\right)$$

$$\Leftrightarrow m \cdot a = \left(\sum_{i=1}^{m} y_i\right)$$

$$\Leftrightarrow a = \frac{\sum_{i=1}^{m} y_i}{m} = \bar{y}$$

-) $R^2 = 1 - \left(\sum_{i=1}^{m} \left(\hat{h}(\vec{x}_i) - y_i\right)^2\right) \Big/ \left(\sum_{i=1}^{m} \left(y_i - \bar{y}\right)^2\right)$ (*)

this is the error of $\hat{h}$ (on the training set) relative to the error of the "best" naive predictor (which always predicts a constant value without looking at $\vec{x}$).

$\Rightarrow 1 - (*)$ is a measure of how well $\hat{h}$ performs compared to the best naive predictor.

R^2 can be negative: if the prediction by $\hat{h}$ is worse than the prediction of the naive model $h(\vec{x}) = \bar{y}$