Seminar 1

① Consider training set $S = \{(x_i, f(x_i))\}_{i=1}^m \subseteq (\mathbb{R}^d \times \{0,1\})^m$

We consider the classifier from lecture 2, $h_S : \mathbb{R}^d \to \{0,1\}$

$$h_S(x) = \begin{cases} y_i, & \text{if } \exists i \in \{1,\dots,m\} \text{ s.t. } x_i = x \\ 0, & \text{otherwise} \end{cases}$$

We want to show that the classifier h_S can be written as a thresholded polynomial $p_S(x)$, meaning that we want to find a polynom p_S such that

$$h_S(x) = 1 \iff p_S(x) \geq 0.$$

Let's consider the ~~easiest~~ simplest case, $d = 1$ (so x_i is a scalar)

1st try, Consider the polynom $p_S(x) = -\prod_{i=1}^m (x - x_i)$

if $x = x_i$ for some $i \in \{1,\dots,m\} \Rightarrow p_S(x) = p_S(x_0) = 0$. It will not work if the label of the label $y_i = 0$ (in this case $p_S(x_0) = 0 \Rightarrow h_S(x_0) = 1$)

Also if x doesn't appear in the training data you don't know if $p_S(x) \geq 0$ or $p_S(x) < 0$

2nd try Consider the polynom $p_S(x) = -\prod_{i=1}^m (x - x_0)^2$

if $x \neq x_0$ for some $i \in \{1,\dots,m\} \Rightarrow p_S(x) \geq p_S(x_0) = 0 \Rightarrow h_S(x) = 1$.

For points $(x_i, 0) \in S$ it will not work

For all other points it would work fine

3rd try Consider the polynom: $p_S(x) = -\prod_{\substack{i=1 \\ y_i = 1}}^m (x - x_i)^2$

(in this case if all $y_0 = 0$ then $p_S(x) = -1$)

if $x = x_0$ for some $i \in \{1,\dots,m\}$ → if $y_0 = 1 \Rightarrow p_S(x) = 0 \Rightarrow h_S(x) = 1$ ✓

→ if $y_i = 0 \to p_S(x) < 0 \Rightarrow h_S(x) = 0$ ✓

if $x \neq x_0$ for all $i \in \{1,\dots,m\} \Rightarrow p_S(x) < 0 \to h_S(x) = 0$ ✓

Other choices would be: $p_S(x) = -\prod_{i=1}^m (x - x_i)^{2y_i}$

$$p_S(x) = -\prod_{i=1}^m \left[(x - x_i)^2 + 1 - y_0\right]$$

Consider now the general case, d can be > 1

for $d=1$ we have seen that

$$p_S(x) = -\prod_{\substack{i=1 \\ y_i=1}}^{m} (x - x_i)^2 \text{ works fine}$$

In the general case we consider the L_2 distance (Euclidean distance)

$$p_S(x) = -\prod_{\substack{i=1 \\ y_i=1}}^{m} \epsilon_i \|x - x_i\|_2^2$$

This polynomial will work fine.

(2) $\mathcal{H}_{rec}^2 = \{ h_{(a_1,b_1,a_2,b_2)} : \mathbb{R}^2 \to \{0,1\},\ a_1 \le b_1 \text{ and } a_2 \le b_2,$

$$h_{(a_1,b_1,a_2,b_2)}(x_1,x_2) = \begin{cases} 1, & a_1 \le x_1 \le b_1 \text{ and } a_2 \le x_2 \le b_2 \\ 0, & \text{otherwise} \end{cases} \}$$

$\mathcal{H}_{rec}^2$ is an infinite size hypothesis class, it is called the class of all axis aligned rectangles in the plane.

We want to prove that $\mathcal{H}_{rec}^2$ is PAC-learnable.

From the definition of PAC-learnability we know that $\mathcal{H} = \mathcal{H}_{rec}^2$ is PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and exists a learning algorithm A with the following property: for every $\epsilon, \delta > 0$, for every labeling function $f \in \mathcal{H}_{rec}^2$ (realizability case), for every distribution D on $\mathbb{R}^2$ when we run the learning algorithm A on a training set S consisting of $m \ge m_{\mathcal{H}}(\epsilon, \delta)$ examples sampled i.i.d from D and labeled by f the algorithm A returns a hypothesis $h_S \in \mathcal{H}$ such that with probability at least $1-\delta$ (over the choice of examples) the real risk of h_S is smaller than ϵ:

$$P_{S \sim D^m}(L_{f,D}(h_S) \le \epsilon) \ge 1 - \delta \text{ or otherwise said}$$

$$P_{S \sim D^m}(L_{f,D}(h_S) > \epsilon) < \delta$$

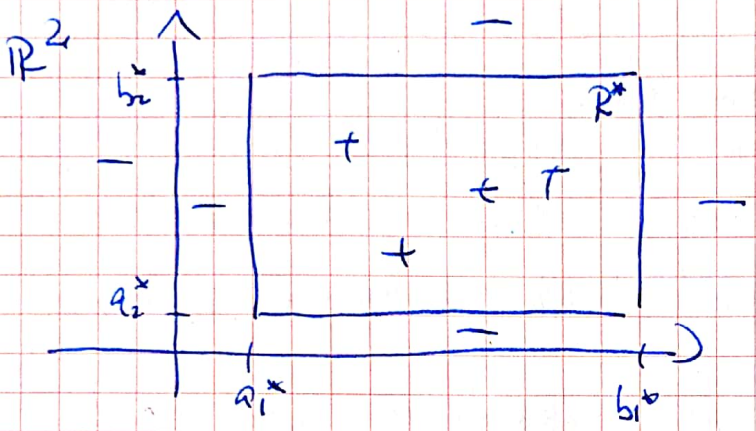
First we need to find the algorithm A.

We are under the realizability assumption, so there exist a labeling function $f \in \mathcal{H}$, $f = h^*_{(a_1^*, b_1^*, a_2^*, b_2^*)}$ that labels the training data.

Consider the training set $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$

$y_i = h^*_{(a_1^*, b_1^*, a_2^*, b_2^*)}(x_i)$, $x_i \in \mathbb{R}^2$

$x_i = (x_{i1}, x_{i2})$ }



h^* labels each point drawn from the rectangle $R^* = [a_1^*, b_1^*] \times [a_2^*, b_2^*]$ with label 1, and all other points with label 0.

So $h^*_{(a_1^*, b_1^*, a_2^*, b_2^*)} = \mathbb{1}_{R^*}$

Consider the following algorithm A, that takes as input the training set S and output h_S.

$h_S = h_{(a_{1S}, b_{1S}, a_{2S}, b_{2S})}$ where

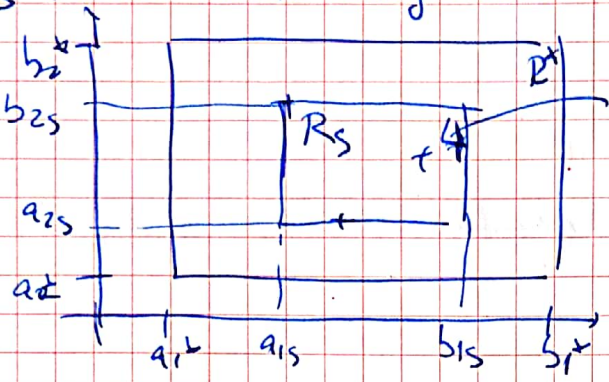
$$a_{1S} = \min_{\substack{i=1,m \\ y_i = 1}} x_{i1} \qquad a_{2S} = \min_{\substack{i=1,m \\ y_i = 1}} x_{i2}$$

$$b_{1S} = \max_{\substack{i=1,m \\ y_i = 1}} x_{i1} \qquad b_{2S} = \max_{\substack{i=1,m \\ y_i = 1}} x_{i2}$$

If all $y_i = 0$ then all points x_i have label 0, then is no positive example. In this case choose $z = (z_1, z_2)$ a point that is not in the training set S and take $a_{1S} = b_{1S} = z_1$, $a_{2S} = b_{2S} = z_2$.

As in the indication $h_S = h_{(a_{1S}, b_{1S}, a_{2S}, b_{2S})} = \mathbb{1}_{[a_{1S}, b_{1S}] \times [a_{2S}, b_{2S}]}$ is the indicator function of the tightest rectangle $R_S = [a_{1S}, b_{1S}] \times [a_{2S}, b_{2S}]$ enclosing all positive examples.

By construction A is an ERM, meaning that $L_{h^*, D}(h_S) = 0$, h_S doesn't make any errors on the training set S.



this is the rectangle R_S implemented by $h_{(a_{1S}, b_{1S}, a_{2S}, b_{2S})}$

Now we want to find the sample complexity $m_H(\varepsilon, \delta)$ such that

$$P_{S \sim D^m}(L_{h^*, D}(h_S) \leq \varepsilon) \geq 1 - \delta \quad \text{when } S \text{ contains } m \geq m_H(\varepsilon, \delta) \text{ examples.}$$

We make the observation that h_S can make errors in region $R^* - R_S$, labeling points that should get label 1 with label 0. All points $\in R_S$ will be labeled correctly, all points outside R^* will be labelled correctly.

Let's fix $\varepsilon > 0$, $\delta > 0$ and consider a distribution D over $\mathbb{R}^2$.

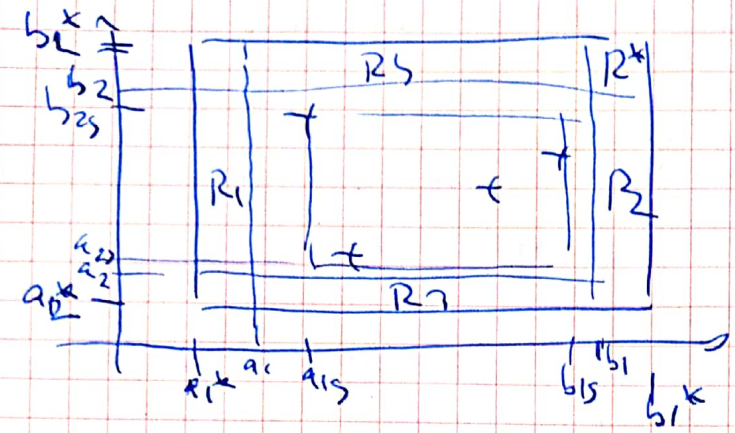
Case 1) if $D(R^*) = P_{x \sim D}(x \in R^*) \leq \varepsilon$ then in this case

$$L_{h^*, D}(h_S) = P_{x \sim D}(h_S(x) \neq h^*(x)) = P_{x \sim D}(x \in R^* - R_S) \leq$$

$$\leq P_{x \sim D}(x \in R^*) \leq \varepsilon \text{ so we have that}$$

$$P_{S \sim D^m}(L_{h^*, D}(h_S) \leq \varepsilon) = 1 \text{ (this happens all the time)}$$

Case 2 $\mathcal{D}(R^*) = \underset{x \in D}{P}(x \in R^*) > \varepsilon$



We construct as in the indicator the 4 rectangles R_1, R_2, R_3, R_4

$R_1 = [a_1^*, a_1] \times [a_2^*, b_2^*)$

$R_2 = [b_{15}, b_1^*] \times [a_2^*, b_2^*)$

$R_3 = [a_1^*, b_1^*] \times [a_2^*, a_2)$

$R_4 = [a_1^*, b_1^*) \times [b_{25}, b_2^*)$

with $\mathcal{D}(R_i) = \underset{x \sim D}{P}(x \in R_i) = \frac{\varepsilon}{4}$.

if $R = [a_{15}, b_{15}) \times [a_{25}, b_{25})$ (the rectangle returned by A, implemented by h_S) returned each R_i, $i = \overline{1,4}$ then

$$L_{h^*,D}(h_S) = \underset{x \sim D}{P}(h^*(x) \neq h_S(x)) = \underset{x \sim D}{P}(x \in R^* \setminus R_S) \leq$$

$$\leq \underset{x \in D}{P}(x \in R_1 \cup R_2 \cup R_3 \cup R_4) \leq \sum_{i=1}^{4} \underset{x \sim D}{P}(x \in R_i)$$

$$= \sum_{i=1}^{4} \mathcal{D}(R_i) = 4 \cdot \frac{\varepsilon}{4} = \varepsilon$$

So, in this case $\underset{S \sim D^m}{P}(L_{h^*,D}(h_S) \leq \varepsilon) = 1$ (this happens always)

In order to have $L_{h^*,D}(h_S) > \varepsilon$ we need that R_S will not intersect at least one rectangle R_i.

We denote with ~~$F_i = \{S \sim D^m \mid R_S \cap R_i = \emptyset\}$~~

$$F_i = \{S \sim D^m \mid R_S \cap R_i = \emptyset\}$$

This leads to the following:

at least one F_i will happen

$$P_{S\sim D^m}\left(L_{h^*,D}(h_S) > \varepsilon\right) = P_{S\sim D^m}\left(F_1 \cup F_2 \cup F_3 \cup F_4\right) \leq$$

$$\leq \sum_{i=1}^{4} P_{S\sim D^m}(F_i)$$

Now, $P_{S\sim D^m}(F_i)$ = what is the probability that R_S will not intersect R_i

= the probability that no point from R_i is sampled in S =

$$= \left(1 - \frac{\varepsilon}{4}\right)^m$$

So $P_{S\sim D^m}\left(L_{h^*,D}(h_S) > \varepsilon\right) \leq \sum_{i=1}^{4} P_{S\sim D^m}(F_i) = 4\cdot\left(1-\frac{\varepsilon}{4}\right)^m$

Now, we know from lecture 2 that $1 - x \leq e^{-x}$

So $1 - \frac{\varepsilon}{4} \leq e^{-\frac{\varepsilon}{4}}$

So $P_{S\sim D^m}\left(L_{h^*,D}(h_S) > \varepsilon\right) \leq 4\cdot\left(1-\frac{\varepsilon}{4}\right)^m \leq 4\cdot e^{-\frac{\varepsilon}{4}m}$

this is the probability to have h_S not ε-accurate $> \varepsilon$

We want to make this probability very small, smaller than δ.

$$4\cdot e^{-\frac{\varepsilon}{4}m} < \delta$$

$$e^{-\frac{\varepsilon}{4}m} < \frac{\delta}{4} \quad / \cdot \log_e$$

$$-\frac{\varepsilon}{4}m < \log\frac{\delta}{4} \quad / \cdot\left(-\frac{4}{\varepsilon}\right)$$

$$m > -\frac{1}{\varepsilon}\log\frac{\delta}{4} = \frac{4}{\varepsilon}\log\frac{4}{\delta}$$

So if you take $m \geq m_H(\varepsilon,\delta) = \frac{4}{\varepsilon}\log\frac{4}{\delta}$ we obtain the desired results.

Repeat the previous question for class of alligned rectangles in $\mathbb{R}^d$.

in $\mathbb{R}^d$ we have $\mathcal{H}^d_{rec} = \{ h_{(a_1,b_1,\dots,a_d,b_d)} : \mathbb{R}^d \to \{0,1\},\ a_i \le b_i \}$

$$h_{(a_1,b_1,\dots,a_d,b_d)} = \mathbb{1}_{[a_1,b_1]\times[a_2,b_2]\times\dots\times[a_d,b_d]}$$

All the arguments used previously work well, the general result will be that $m_{\mathcal{H}}(\varepsilon,\delta) \le \frac{2d}{\varepsilon} \cdot \log\frac{2d}{\delta}$.

For $d=2$ we obtain the previous result.

The runtime of algorithm A is given by taking min, max over each dimension, so this mean $O(m \cdot d)$

m = number of (positive) examples $= O\left(\frac{2d}{\varepsilon} \cdot \log\frac{2d}{\delta}\right)$

d = number of dimensions

So we have the complexity of algorithm A is $O\left(\frac{2d^2}{\varepsilon} \cdot \log\frac{2d}{\delta}\right)$

which is polynomial in $d, \frac{1}{\varepsilon}, \frac{1}{\delta}$.

(3) $\mathcal{H}$ is PAC learnable and $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ is its sample complexity

a) Given $\delta \in (0,1)$ and given $0 \le \varepsilon_1 \le \varepsilon_2 < 1$ we have that

$$m_{\mathcal{H}}(\varepsilon_1,\delta) \ge m_{\mathcal{H}}(\varepsilon_2,\delta)$$

$\mathcal{H}$ is PAC learnable with sample complexity $m_{\mathcal{H}}(\cdot,\cdot)$ means that there exists a learning algorithm A with the property: for every $\varepsilon,\delta > 0$ when given to the algorithm A a sample set S of m examples, $m \ge m_{\mathcal{H}}(\varepsilon,\delta)$ (samples are labelled by $f \in \mathcal{H}$ and i.i.d from a distribution $\mathcal{D}$) we have that $h_S = A(S)$ with the real risk:

$$P_{S\sim\mathcal{D}^m}\left(L_{f,\mathcal{D}}(h_S) \le \varepsilon\right) > 1-\delta$$

We apply this for ε_2 and $\delta \to P_{S\sim\mathcal{D}^m}\left(L_{f,\mathcal{D}}(h_S) \le \varepsilon_1\right) > 1-\delta$ if $m \ge m_{\mathcal{H}}(\varepsilon_1,\delta)$

We know that $\varepsilon_2 \geq \varepsilon_1$ so we have that

$$P_{S\sim D^m}(L_{f,D}(h_S) \leq \varepsilon_2) > 1-\delta \text{ if } m \geq m_{\mathcal{H}}(\varepsilon_1, \delta)$$

But $m_{\mathcal{H}}(\varepsilon, \delta)$ is the smallest number of examples for which the above inequality holds. So, if it holds for $m \geq m_{\mathcal{H}}(\varepsilon_1, \delta)$ we have that $m_{\mathcal{H}}(\varepsilon_1, \delta) \geq m_{\mathcal{H}}(\varepsilon_2, \delta)$.

b) given $\varepsilon \in (0,1)$, $0 < \delta_1 \leq \delta_2 < 1$ we have that $m_{\mathcal{H}}(\varepsilon, \delta_1) \geq m_{\mathcal{H}}(\varepsilon, \delta_2)$

Using the same arguments from a) we have that

$$P_{S\sim D^m}(L_{f,D}(h_S) \leq \varepsilon) > 1-\delta_1 \text{ if } m \geq m_{\mathcal{H}}(\varepsilon, \delta_1)$$

$$\delta_1 \leq \delta_2 \Rightarrow 1-\delta_1 \geq 1-\delta_2 \Rightarrow$$

$$P_{S\sim D^m}(L_{f,D}(h_S) \leq \varepsilon) > 1-\delta_2 \text{ if } m \geq m_{\mathcal{H}}(\varepsilon, \delta_1)$$

But $m_{\mathcal{H}}(\varepsilon, \delta_2)$ is the smallest number of examples for which the above inequality holds (if $m \geq m_{\mathcal{H}}(\varepsilon, \delta_2)$). So, if it holds for $m \geq m_{\mathcal{H}}(\varepsilon, \delta_1)$ we have that $m_{\mathcal{H}}(\varepsilon, \delta_1) \geq m_{\mathcal{H}}(\varepsilon, \delta_2)$

4) $\mathcal{X}$ discrete domain, $\mathcal{H}_{singleton} = \{h_z : z \in \mathcal{X}\} \cup \{h^-\}$

$\forall z \in \mathcal{X}$, $h_z : \mathcal{X} \to \{0,1\}$, $h_z(x) = \begin{cases} 1, & x = z \\ 0, & x \neq z \end{cases}$

$h^- : \mathcal{X} \to \{0,1\}$, $h^-(x) = 0 \ \forall x \in \mathcal{X}$

4.1) Describe an algorithm that implements the ERM rule for learning $\mathcal{H}_{singleton}$ in the realizable setup.

Consider $S = \{(x_i, h^*(x_i)), x_i \text{ iid for a distribution } D \text{ over } \mathcal{X}\}_{i=1}^m$

The algorithm A is the following:

loop over training examples, if there is an $i \in \{1, \dots, m\}$ such that $y_i = 1$ then return hypothesis $h_S = A(S) = h_{x_i}$

Otherwise return h^-.

From construction A is ERM meaning that $L_S(h_S) = 0$.

4.2) Show that $\mathcal{H}_{singleton}$ is PAC-learnable. Provide an upper bound on the sample complexity.

Let $\varepsilon, \delta > 0$ and fix a distribution D over X.

The only case in which algorithm A fails is the case where $h^* = h_z$ and the sample $S = \{(x_i, y_i) | i = 1, \dots, m\}$, x_i sampled i.i.d. from D, doesn't contain any positive example, so all $y_0 = 0$ $\forall i = 1, \dots, m$.

In this case $h_S = A(S) = h^-$ which is different than h^*. However, even if the A fails if $D(\{z\}) \leq \varepsilon$ then everything is ok as we have that:

$$P_{S \sim D^m}(L_{h^*, D}(h_S) \leq \varepsilon) = 1$$

So, we have to upper bound the sample complexity in the case where $D(\{z\}) > \varepsilon$ and there is no positive examples in the set S (actually, for this problem there is just one positive possible training point = z).

So we have that

$$P_{S \sim D^m}(L_{h^*, D}(h_S) > \varepsilon) = \text{probability that each point in } S \text{ is different than } z \text{ (which has probability mass} > \varepsilon) \leq (1-\varepsilon)^m \leq e^{-\varepsilon m}$$

So if we set $e^{-\varepsilon m} < \delta \Leftrightarrow -\varepsilon m < \log \delta$

$$m > -\frac{1}{\varepsilon} \log \delta \Rightarrow m > \frac{1}{\varepsilon} \log \frac{1}{\delta}.$$

So if $m \geq \lceil \frac{1}{\varepsilon} \log \frac{1}{\delta} \rceil$ we have that the $P_{S \sim D^m}(L_{h^*, D}(h_S) > \varepsilon) < \delta$

So the upper bound of $m_{\mathcal{H}}(\varepsilon, \delta)$ is $m_{\mathcal{H}}(\varepsilon, \delta) \leq \lceil \frac{1}{\varepsilon} \log \frac{1}{\delta} \rceil$