Advanced Machine Learning Seminar 1 - solutions

Exercise 1 Consider the 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: $h_S \colon \mathbb{R}^d \to \{0,1\}$

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

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 polynomial P_S such that $h_S(x) = 1 \Leftrightarrow P_S(x) \geq 0$.

Proof. Let's consider the simpler case, $d = 1$ (so x_i is a scalar).

1st try: Consider the polynomial

$$P_S(x) = -\prod_{i=1}^m (x - x_i)$$

If $x = x_i$ for some $i \in \{1,\dots,m\} \Rightarrow P_S(x) = P_S(x_i) = 0 \Rightarrow h_S(x) = 1$.

It will not work if the label of the point x_i is $y_i = 0$ (in this case, $P_S(x_i) = 0 \Rightarrow h_S(x_i) = 1$).

Also, if x doesn't appear in the training data, we don't know if $P_S(x) \geq 0$ or $P_S(x) < 0$.

2nd try: Consider the polynomial

$$P_S(x) = -\prod_{i=1}^m (x - x_i)^2$$

If $x = x_i$ for some $i \in \{1,\dots,m\} \Rightarrow P_S(x) = P_S(x_i) = 0 \Rightarrow h_S(x) = 1$.

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

For all other points, it will work fine.

3rd try: Consider the polynomial

$$P_S(x) = -\prod_{\substack{i=1 \\ y_i=1}}^m (x - x_i)^2$$

In this case, if all $y_i = 0$, then $P_S(x) = -1$.

$$\text{If } x = x_i, \text{ for some } i \in \{1,\dots,m\}: \begin{array}{lll} \text{if } y_i = 1 & \Rightarrow P_S(x) = 0 & \Rightarrow h_S(x) = 1 \checkmark \\ \text{if } y_i = 0 & \Rightarrow P_S(x) < 0 & \Rightarrow h_S(x) = 0 \checkmark \end{array}$$

If $x \neq x_i$ for all $i \in \{1,\dots,m\} \Rightarrow P_S(x) < 0 \Rightarrow h_S(x) = 0 \checkmark$

Other choices for polynomial P_S could be:

$$P_S(x) = -\prod_{i=1}^m (x - x_i)^{2y_i}$$

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

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):

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

This polynomial will work fine.

□

Exercise 2

$$\mathcal{H}^2_{rec} = \left\{ \begin{array}{l} h_{(a_1,b_1,a_2,b_2)} \colon \mathbb{R}^2 \to \{0,1\}, a_1 \leq b_1 \text{ and } a_2 \leq b_2, \\ h_{(a_1,b_1,a_2,b_2)}(x_1,x_2) = \begin{cases} 1, & a1 \leq x_1 \leq b_1 \text{ and } a_2 \leq x_2 \leq b_2 \\ 0, & \text{otherwise} \end{cases} \end{array} \right\}$$

$\mathcal{H}^2_{rec}$ is an infinite size hypothesis class, it is called the class of all axis aligned rectangles in the plane. We want to prove that $\mathcal{H}^2_{rec}$ is PAC-learnable.

Proof. From the definition of PAC-learnability, we know that $\mathcal{H} = \mathcal{H}^2_{rec}$ is PAC-learnable if there exists a function $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ and there exists a learning algorithm A with the following property: for every $\epsilon, \delta > 0$, for every labeling function $f \in \mathcal{H}^2_{rec}$ (realizability case), for every distribution $\mathcal{D}$ on $\mathbb{R}^2$ when we run the learning algorithm A on a training set S consisting of $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ examples sampled i.i.d. from $\mathcal{D}$ 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:

$$\underset{S \sim \mathcal{D}^m}{P}(L_{f,\mathcal{D}}(h_S) \leq \epsilon) \geq 1 - \delta \text{ or otherwise said}$$
$$\underset{S \sim \mathcal{D}^m}{P}(L_{f,\mathcal{D}}(h_S) > \epsilon) < \delta$$

First, we need to find the algorithm A.

We are under the realizability assumption, so there exists 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 $S = \left\{ (x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m) \;\middle|\; \begin{array}{l} y_i = h^*_{(a_1^*,b_1^*,a_2^*,b_2^*)}(x_i), \\ x_i \in \mathbb{R}^2, x_i = (x_{i1}, x_{i2}) \end{array} \right\}$

As in Figure 1, 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 we have $h^*_{(a_1^*,b_1^*,a_2^*,b_2^*)} = \mathbb{1}_{R^*}$

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

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

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

If all $y_i = 0$, then all points x_i have label 0, so there 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 (see Figure 2).

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

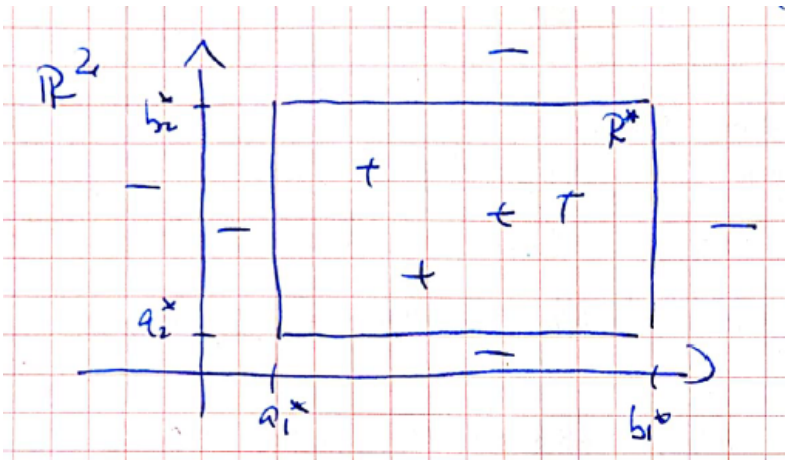

Figure 1: All the points that fall in rectangle R^* will be labeled by h^* with label 1 (+), the other points will be labeled with label 0 (-).

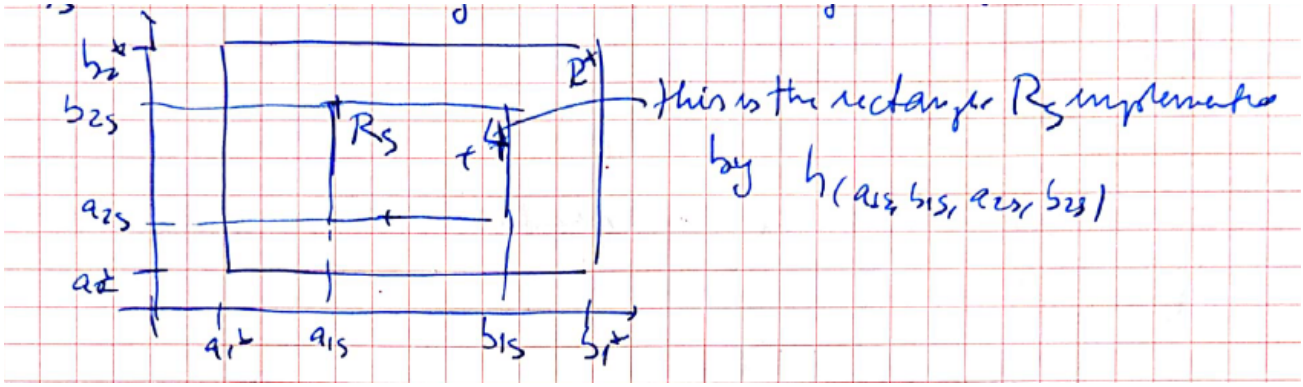

Figure 2: Rectangle R_S is the tightest rectangle enclosing all positive examples.

Now we want to find the sample complexity $m_{\mathcal{H}}(\epsilon, \delta)$ such that

$$\underset{S\sim\mathcal{D}^m}{P}(L_{h^*,\mathcal{D}}(h_S) \leq \epsilon) \geq 1-\delta \text{ where } S \text{ contains } m \geq m_{\mathcal{H}}(\epsilon,\delta) \text{ examples.}$$

We make the observation that h_S makes errors in region $R^* \setminus R_S$, assigning the label 0 to points that should get label 1. All points $\in R_S$ will be labeled correctly (label 1), all points outside R^* will be labeled correctly (label 0).

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

Case 1)

$$\text{If } \mathcal{D}(R^*) = \underset{x\sim\mathcal{D}}{P}(x \in R^*) \leq \epsilon \text{ then in this case}$$
$$L_{h^*,\mathcal{D}}(h_S) = \underset{x\sim\mathcal{D}}{P}(h_S(x) \neq h^*(x)) = \underset{x\sim\mathcal{D}}{P}(x \in R^* \setminus R_S) \leq \underset{x\sim\mathcal{D}}{P}(x \in R^*) \leq \epsilon \text{ so we have that}$$
$$\underset{S\sim\mathcal{D}^m}{P}(L_{h^*,\mathcal{D}}(h_S) \leq \epsilon) = 1 \text{ (this happens all the time)}$$

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

We construct as in the indication the 4 rectangles R_1, R_2, R_3, R_4 (see Figure 3):

$$R_1 = [a_1^*, a_1] \times [a_2^*, b_2^*] \qquad R_2 = [b_1, b_1^*] \times [a_2^*, b_2^*]$$
$$R_3 = [a_1^*, b_1^*] \times [a_2^*, a_2] \qquad R_4 = [a_1^*, b_1^*] \times [b_2, b_2^*] \qquad \text{with } \mathcal{D}(R_i) = \underset{x\sim\mathcal{D}}{P}(x \in R_i) = \frac{\epsilon}{4}$$

If $R_S = [a_{1S}, b_{1S}] \times [a_{2S}, b_{2S}]$ (the rectangle returned by A, implemented by h_S) intersects each $R_i, i = \overline{1,4}$:

$$L_{h^*,\mathcal{D}}(h_S) = \underset{x\sim\mathcal{D}}{P}(h^*(x) \neq h_S(x)) = \underset{x\sim\mathcal{D}}{P}(x \in R^* \setminus R_S) \leq \underset{x\sim\mathcal{D}}{P}(x \in R_1 \cup R_2 \cup R_3 \cup R_4) \leq$$
$$\leq \sum_{i=1}^{4} \underset{x\sim\mathcal{D}}{P}(x \in R_i) = \sum_{i=1}^{4} \mathcal{D}(R_i) = 4 \cdot \frac{\epsilon}{4} = \epsilon$$

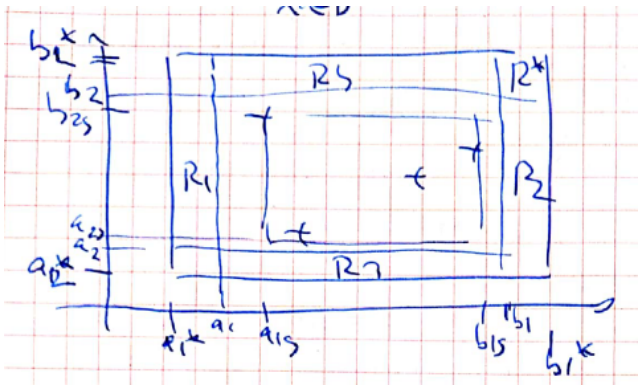

Figure 3: Constructing the rectangles R_1, R_2, R_3 and R_4.

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

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

We denote with F_i this event, so we have $F_i = \{S \sim \mathcal{D}^m \mid R_S \cap R_i = \emptyset\}$. This leads to the following:

$$\underset{S\sim\mathcal{D}^m}{P}(L_{h^*,\mathcal{D}}(h_S) > \epsilon) \leq \underset{S\sim\mathcal{D}^m}{P}(\overset{\text{at least one } F_i \text{ will happen}}{\overset{\nearrow}{F1 \cup F2 \cup F3 \cup F4}}) \leq \sum_{i=1}^{4} \underset{S\sim\mathcal{D}^m}{P}(F_i)$$

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

$$= \text{the probability that no point from } R_i \text{ is sampled in } S$$
$$= \left(1 - \frac{\epsilon}{4}\right)^m$$

So

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

Now, we know from lecture 2 that $1 - x \leq e^{-x}$, so $1 - \frac{\epsilon}{4} \leq e^{-\frac{\epsilon}{4}}$, which means that

$$\underset{S\sim\mathcal{D}^m}{P}(\underset{\substack{\uparrow \\ \text{this is the probability that} \\ h_S \text{ will make an error} > \epsilon}}{L_{h^*,\mathcal{D}}(h_S) > \epsilon}) \leq 4 \cdot \left(1 - \frac{\epsilon}{4}\right)^m \leq 4 \cdot e^{-\frac{\epsilon}{4}m}$$

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

$$\begin{aligned} 4 \cdot e^{-\frac{\epsilon}{4}m} &< \delta \\ e^{-\frac{\epsilon}{4}m} &< \frac{\delta}{4} \quad \Big| \cdot \log_e \\ -\frac{\epsilon}{4} \cdot m &< \log\frac{\delta}{4} \quad \Big| \cdot \left(-\frac{4}{\epsilon}\right) \\ m &> -\frac{4}{\epsilon}\log\frac{\delta}{4} = \frac{4}{\epsilon}\log\frac{4}{\delta} \end{aligned}$$

So, if we take $m \geq m_{\mathcal{H}}(\epsilon, \delta) = \frac{4}{\epsilon} \cdot \log\frac{4}{\delta}$, we obtain the desired results.

Repeat the previous question for the class of aligned rectangles in $\mathbb{R}^d$.
In $\mathbb{R}^d$, we have

$$\mathcal{H}_{rec}^d = \left\{ \begin{array}{l} h_{(a_1,b_1,a_2,b_2,\ldots,a_d,b_d)} \colon \mathbb{R}^d \to \{0,1\} \mid a_i \leq b_i, i = \overline{1,d} \\ h_{(a_1,b_1,a_2,b_2,\ldots,a_d,b_d)} = \mathbb{1}_{[a_1,b_1]\times[a_2,b_2]\times\cdots\times[a_d,b_d]} \end{array} \right\}$$

All the arguments used previously will work, the general result will be that $m_{\mathcal{H}}(\epsilon, \delta) = \frac{2d}{\epsilon} \cdot \log\frac{2d}{\delta}$.
For $d = 2$, we obtain the previous result.

The runtime of algorithm A is given by taking minimum over each dimension, so this means $\mathcal{O}(m * d)$:
m = number of (positive) examples = $\mathcal{O}(\frac{2d}{\epsilon} \cdot \log\frac{2d}{\delta})$
d = number of dimensions.

So we have that the complexity of algorithm A is $\mathcal{O}(\frac{2d^2}{\epsilon} \cdot \frac{2d}{\delta})$, which is polynomial in d, $\frac{1}{\epsilon}$, $\frac{1}{\delta}$.

□

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

a) Given $\delta \in (0,1)$ and given $0 < \epsilon_1 \leq \epsilon_2 < 1$, we have that $m_{\mathcal{H}}(\epsilon_1, \delta) \geq m_{\mathcal{H}}(\epsilon_2, \delta)$.

Proof. $\mathcal{H}$ is PAC-learnable with sample complexity $m_{\mathcal{H}}(\cdot,\cdot)$ means that there exists a learning algorithm A with the following property: for every $\epsilon, \delta > 0$, when we run the algorithm A on a sample set S of m examples, $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ (samples are labeled 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 $\underset{S\sim\mathcal{D}^m}{P}(L_{f,\mathcal{D}}(h_S) \leq \epsilon) > 1 - \delta$.

We apply this for ϵ_1 and δ:

$$\underset{S\sim\mathcal{D}^m}{P}(L_{f,\mathcal{D}}(h_S) \leq \epsilon_1) > 1 - \delta \text{ if } m \geq m_{\mathcal{H}}(\epsilon_1, \delta)$$

We know that $\epsilon_2 \geq \epsilon_1$, so we have that

$$\underset{S\sim\mathcal{D}^m}{P}(L_{f,\mathcal{D}}(h_S) \leq \epsilon_2) > 1 - \delta \text{ if } m \geq m_{\mathcal{H}}(\epsilon_1, \delta)$$

But $m_{\mathcal{H}}(\epsilon_2, \delta)$ is the smallest number of examples for which the above inequality holds. So, if it holds for $m \geq m_{\mathcal{H}}(\epsilon_1, \delta)$, we have that $m_{\mathcal{H}}(\epsilon_1, \delta) \geq m_{\mathcal{H}}(\epsilon_2, \delta)$. □

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

Proof. Using the same arguments from a), we have that

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

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

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

$$\forall z \in \mathcal{X} \quad h_z \colon \mathcal{X} \to \{0,1\}, \quad h_z(x) = \begin{cases} 1, & x = z \\ 0, & x \neq z \end{cases}$$
$$h^- \colon \mathcal{X} \to \{0,1\}, \quad h^-(x) = 0, \ \forall x \in \mathcal{X}$$

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

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

Proof. Let $\epsilon, \delta > 0$ and fix a distribution $\mathcal{D}$ over $\mathcal{X}$.
The only case in which the algorithm A fails is the case where $h^* = h_z$ and the sample $S = \{(x_i, y_i) \mid x_i \text{ sampled i.i.d. from } \mathcal{D}\}$ doesn't contain any positive examples, so all $y_i = 0\ \forall i = \overline{1,m}$. In this case, $h_S = A(S) = h^-$, which is different from h^*. However, even if the algorithm A fails if $\mathcal{D}(\{z\}) \leq \epsilon$, then everything is ok, as we have that:

$$\underset{S \sim \mathcal{D}^m}{P}(L_{h^*,d}(h_S) \leq \epsilon) = 1 \quad \checkmark\checkmark$$

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

$\underset{S \sim \mathcal{D}^m}{P}(L_{h^*,\mathcal{D}}(h_S) > \epsilon) =$ probability that each point in S
is different than z (which has probability mass $> \epsilon$) $\leq (1-\epsilon)^m \leq e^{-\epsilon m}$
So, if we set $e^{-\epsilon m} < \delta \Rightarrow -\epsilon m < \log \delta$

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

If $m \geq \left\lceil \frac{1}{\epsilon} \log \frac{1}{\delta} \right\rceil$ we have that $\underset{S \sim \mathcal{D}^m}{P}(L_{h^*,\mathcal{D}}(h_S) > \epsilon) < \delta$

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

□