Advanced Machine Learning

Assignment – 2 **Tomeh Lara** **June 2022**

Question – 1

Consider $\mathcal{H}$ the class of 3-piece classifiers (signed intervals) ….

Part a.

we first show that for every distribution that is consistent with $\mathcal{H}$, there exists a decision stump with $L_{\mathcal{D}}(h) \leq 1/3$. Indeed, just note that every classifier in $\mathcal{H}$ consists of three regions (two unbounded rays and a center interval) with alternate labels. For any pair of such regions, there exists a decision stump that agrees with the labeling of these two components. Note that for every distribution $\mathcal{D}$ over $\mathbb{R}$ and every partitioning of the line into three such regions, one of these regions must have $\mathcal{D}$ -weight of at most $1/3$. Let $h \in \mathcal{H}$ be a zero-error hypothesis. A decision stump that disagrees with h only on such a region has an error of at most $1/3$. Finally, since the VC-dimension of decision stumps is 2, if the sample size is greater than $\Omega(\log(1/\delta)/\epsilon^2)$, then with probability of at least $1-\delta$, the ERM_B rule returns a hypothesis with an error of at most $1/3+\epsilon$. Setting $\epsilon = 1/12$ we obtain that the error of ERM_B is at most $1/3 + 1/12 = 1/2 - 1/12$.
We see that ERM_B is a γ-weak learner for $\mathcal{H}$.
The set family H contains all the real intervals, i.e., all sets of the form $\{x \in [x_0, x_1] | x \in \mathbb{R}\}$ for some $x_0, x_1 \in \mathbb{R}$. For any set C of m real numbers, the intersection $H \cap C$ contains all runs of between 0 and m consecutive elements of C. The number of such runs is $\binom{m+1}{2} + 1$, so $\text{Growth}(H, m) = \binom{m+1}{2} + 1$.

Fix $m \in \mathbb{N}$ and let $S = \{x_1 < x_2 < \cdots < x_m\}$ a set of samples of length m.

The only labels that can be obtained using any $h \in H$ from S are of the form $A = \{0^k 1^{m-k} | k \in \{0, \dots, m\}\}$ or $B = \{1^k 0^{m-k} | k \in \{0, \dots, m\}\}$.

Since$|A| = m + 1, |B| = m + 1, |A| \cap |B| = |\{0^m, 1^m\}| = 2$. And $|A \cup B| = |A| + |B| - |A \cap B| = m + 1 + m + 1 - 2 = 2m$. We know that A and B are the only possible labelings due to the fact that the behavior of a function $h \in H$ when restricted to S changes only in the region of the points from S.

Part b.

Any set of 2 points $x_1 < x_2$ in $\mathbb{R}$ can be shattered by $\mathcal{H}$: consider the functions in $\mathcal{H}$ corresponding to the intervals $[x_1 - 2, x_1 - 1], [x_1 - 1, \frac{x_1+x_2}{2}], [\frac{x_1+x_2}{2}, x_2 + 1], and\ [x_1 - 1, x_2 + 1]$; these functions realize all possible binary labelings of the 2 points. Moreover, no set of 3 points $x_1 < x_2 < x_3$ in $\mathbb{R}$ can be shattered by $\mathcal{H}$: no function in $\mathcal{H}$ can label x_2 as negative and x_1, x_3 as positive. Therefore VCdim($\mathcal{H}$) = 2.

Question – 2

Consider de concept class C_2 formed by the union of two closed intervals ….

Part a. Realizable case.

In this case there is a function $h_{a^*,\,b^*,c^*,d^*}(x) = (1_{[a^*,b^*]} \cup 1_{[c^*,d^*]})(x)$ which label the training points

$$S = \{(x_1, y_1), (x_2, y_2), \ldots . (x_m, y_m)\} \qquad y_i = h_{a^*,\,b^*,c^*,d^*}(x_i)$$

we have to sort the points according to $x's$, finding:

$$S = \{(x_{\sigma(1)}, y_{\sigma(1)}), (x_{\sigma(2)}, y_{\sigma(2)}), \ldots . (x_{\sigma(m)}, y_{\sigma(m)})\}, x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(m)}$$

Taking into account the ERM algorithm for determining a, b, c, d after sorting the training set S.

1) If there are only positive examples return $h_{a,b,c,d}$ where $a = x_{\sigma(1)}, b = c = d = x_{\sigma(m)}$.
2) *If there are no positive labels return:* $h_{a,b,c,d}$ *where* $a = x_{\sigma(1)} - 1, b = x_{\sigma(1)} - 1,\ c = x_{\sigma(m)} + 1,\ d = x_{\sigma(m)} + 1.$
3) If $\exists (x_i, y_i) \in S$ such that $y_i == 1, then$
   $a = b = min_i = \overline{1, m}\,, y_i = +1^{x_i}$
   $c = d = max_i = \overline{1, m}\,, y_i = +1^{x_i}$
4) The final values for at least a and d are found, now we have to adjust the values for b and c.. For $i = \overline{2, m}$

- If $y_{\sigma(i)} == -1\ and\ y_{\sigma(i-1)} == +1$, then $b = x_{\sigma(i-1)}$
- If $y_{\sigma(i)} == +1\ and\ y_{\sigma(i-1)} == -1$, then $c = x_{\sigma(i)}$

  Return $h_{a,b,c,d}$

We have to determine the complexity of this algorithm…

- Sorting $\mathcal{O}(m \cdot log\ m)$.
- Determining there are no positive labels $\mathcal{O}(m)$
- Determining there are only positive labels $\mathcal{O}(m)$
- Final iteration to adjust b and c , $\mathcal{O}(m)$

  Total complexity $\mathcal{O}(m \cdot log\ m)$.

Part b. Agnostic case.

In this case we have various labels for the same point, thus we are dealing with a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$. In the same way to the realizable case, we start by sorting the training set S according to the $x's$, gaining:

$S = \{(x_{\sigma(1)}, y_{\sigma(1)}), (x_{\sigma(2),}y_{\sigma(2)})(x_{\sigma(m),}y_{\sigma(m)})\}$ with $x_{\sigma(1)} \leq\ x_{\sigma(2)} \leq \cdots \leq\ x_{\sigma(m)}$

Consider the set Z containing the values of x' with no repetition

$Z = \{z_1, z_2, \dots, z_n\}$

$z_1 = x_{\sigma(1)} <\ z_2\ < \cdots <\ z_n =\ x_{\sigma(m)} \quad n \leq m$

1. If all $y_i = 0$, in order to return an ERM algorithm we can pick two intervals outside the training set S, for example $a_s = b_s = c_s = d_s =\ x_{\sigma(1)} - 1$.
2. Consider all possible two intervals reunions $Z_{i,j,k,l} = [z_i, z_j] \cup [z_k, z_l], i = \overline{1,n}\ , j = \overline{i,n}, k = \overline{j,n}, l = \overline{k,n}$.

For the ERM algorithm, we have to determine the solution $Z^* = Z_{i^*,j^*,k^*,l^*}$ with the smallest empirical risk. We calculate this as:

$$Loss(Z_{i,j,k,l})$$
$$= \frac{\#\ negative\ points\ inside\ Z_{i,j,k,l} + \#\ positive\ points\ outside\ Z_{i,j,k,l}}{m}$$

First, we pre-compute the total number of positive (pos_prefix_i) and negative points (neg_prefix_i) with value $x \leq x_{\sigma(i)}$ using a dynamic programming approach of prefix-sums. Because we can have multiple points with the same value x, we need the auxiliary pos_i and neg_i, the number of points with positive, respectively negative labels and value $x = x_{\sigma(i)}$. Considering the base case $pos\_prefix_0 = neg\_prefix_0 = 0$, we have the recurrence, for $i = \overline{1,n}$:

- $pos\_prefix_i = pos\_prefix_{i-1} + pos_i$
- $neg\_prefix_i = neg\_prefix_{i-1} + neg_i$

Now, we fix the limits i, j, k, l and find the ones that minimize Loss ($Z_{i,j,k,l}$). An efficient ERM algorithm for this is:

1. Sort S and find $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(m)}$. Build set Z containing value x without repetition: $Z = \{z_1, z_2, \ldots, z_n\}$, $z_1 = x_{\sigma(1)} < z_2 < \cdots < z_n = x_{\sigma(m)}$
2. Check if all y_i $i = \overline{1,m}$ have value 0. If so, return h_{a_s,b_s,c_s,d_s}, where $a_s = b_s = c_s = d_s = z_1 - 1$
3. For $j = \overline{1,n}$, calculate values

$$pos_j = \#\ points\ x_i = z_j\ with\ label\ y_i = 1$$

$$neg_j = \#\ \ points\ x_i = z_j\ with\ label\ y_i = 0$$

4. For $i = \overline{1,n}$

$$pos\_prefix_i = pos\_prefix_{i-1} + pos_i$$
$$neg\_prefix_i = neg\_prefix_{i-1} + neg_i$$

5. min_error $= \frac{m}{m} = 1, i^* = [\,]\,,\ \ j^* = [\,], k^* = [\,], l^* = [\,]$

   for $i = \overline{1,n}$

   for $j = \overline{i,n}$

   for $k = \overline{j,n}$

   for $l = \overline{k,n}$

$$\text{Loss}(Z_{i,j,k,l}) = \frac{(neg\_prefix_j - neg\_prefix_{i-1}) + (neg\_prefix_l - neg\_prefix_{k-1})}{m} + \frac{pos\_prefix_n - (pos\_prefix_j - pos\_prefix_{i-1}) - (pos\_prefix_l - pos\_prefix_{k-1})}{m}$$

   If $\text{Loss}(Z_{i,j,k,l}) < \text{min\_}error$

   min_error = $\text{Loss}(Z_{i,j,k,l})$

   $i^* = i$

   $j^* = j$

   $k^* = k$

   $l^* = l$

6. return h_{a_s,b_s,c_s,d_s}, where $a_s = z_{i^*}, b_s = z_{j^*}, c_s = z_{k^*}, d_s = z_{l^*}$

   Let us calculate the complexity of this algorithm:

   - Sorting $\mathcal{O}(m \cdot log\ m)$.
   - Linear check $\mathcal{O}(m)$
   - Auxiliary counts pre-compute $\mathcal{O}(m)$
   - Prefix-sums DP pre-compute $\mathcal{O}(m)$
   - Finding best i, j, k, l combination $\mathcal{O}(m^4)$ (constant time for Loss using pre-computed prefix-sums)

     Total complexity: $\mathcal{O}(m^4)$.

Question – 3

Consider a modified version of the AdaBoost algorithm ….

Part a.

We will prove that under the constraint $e_i \leq \frac{1}{2} - \gamma_i, \gamma_i > 0, i \in \{1,2,3\}$

There is 0 probability of choosing h_1 in the second round, and we have the following relationship between distributions $\mathcal{D}_{t+1}$ and $\mathcal{D}_t$:

- $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i) * e^{-w_t h_t(x_i) * y_i}}{Z_{t+1}}$
- Z_{t+1} is a normalizing factor
- $w_t = \frac{1}{2} * \log\left(\frac{1}{\epsilon_t} - 1\right)$
- $\epsilon_t = P_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{h_t(x_i) \neq y_i} D_t(i)$
- If $h_t(x_i) = y_i \longrightarrow D_{t+1}(i) = \frac{D_t(i) * e^{-w_t}}{Z_{t+1}} = \frac{D_t(i) * e^{-\frac{1}{2}\ln\left(\frac{1}{\epsilon_t} - 1\right)}}{Z_{t+1}} = \frac{D_t(i) * \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_{t+1}}$
- If $h_t(x_i) \neq y_i \longrightarrow D_{t+1}(i) = \frac{D_t(i) * e^{w_t}}{Z_{t+1}} = \frac{D_t(i) * e^{\frac{1}{2}\ln\left(\frac{1}{\epsilon_t} - 1\right)}}{Z_{t+1}} = \frac{D_t(i) * \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_{t+1}}$
- $Z_{t+1} = \sum_{h_t(x_i) = y_i} D_t(i) \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \sum_{h_t(x_i) \neq y_i} D_t(i) \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = (1 - \epsilon_t) * \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t * \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = 2 * \sqrt{\epsilon_t * (1 - \epsilon_t)}$

Assume that $h_2 = h_1$, then, we have that $\epsilon_2 = \sum_{h_2(x_i) \neq y_i} D_2(i) = \sum_{h_1(x_i) \neq y_i} D_2(i) = \sum_{h_1(x_i) \neq y_i} \frac{D_1(i) * \sqrt{\frac{1-\epsilon_1}{\epsilon_1}}}{Z_2} = \frac{\sum_{h_1(x_i) \neq y_i} D_1(i) * \sqrt{\frac{1-\epsilon_1}{\epsilon_1}}}{2 * \sqrt{\epsilon_1 * (1-\epsilon_1)}} = \frac{\sqrt{\frac{1-\epsilon_1}{\epsilon_1}} * \sum_{h_1(x_i) \neq y_i} D_1(i)}{2 * \sqrt{\epsilon_1 * (1-\epsilon_1)}} = \frac{\sqrt{\frac{1-\epsilon_1}{\epsilon_1}} * \epsilon_1}{2 * \sqrt{\epsilon_1 * (1-\epsilon_1)}} = \frac{1}{2}$

But from the hypothesis we have that $\epsilon_2 = \frac{1}{2} \leq \frac{1}{2} - \gamma_2 \longrightarrow \gamma_2 \leq 0$, a contradiction with $\gamma_i > 0, i \in \{1,2,3\}$.

Part b.

Firstly, we will prove that: $D_{t+1}(i) = \frac{D_t(i)}{1+y_i h_t(x_i)*(1-2*\epsilon_t)}$. This can be proved by considering $D_{t+1}(i) = \frac{D_t(i)* e^{-w_t h_t(x_i)*y_i}}{Z_{t+1}} = \frac{D_t(i)* e^{-w_t h_t(x_i)*y_i}}{2*\sqrt{\epsilon_t*(1-\epsilon_t)}}$.

But $w_t = \frac{1}{2} * \log\left(\frac{1}{\epsilon_t} - 1\right) \rightarrow e^{-\omega t} = \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}$, therefore, it follows that

$$D_{t+1}(i) = \frac{D_t(i) * e^{-w_t h_t(x_i)*y_i}}{2 * \sqrt{\epsilon_t * (1-\epsilon_t)}} = \frac{D_t(i) * \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}^{h_t(x_i)*y_i}}{2 * \sqrt{\epsilon_t * (1-\epsilon_t)}}.$$

It is obvious that for $h_t(x_i) * y_i \in \{-1,1\}$ "which are the only possible combinations" we get the same results in the two relations; thus, they are indeed equals.

Let $a = P_{i\sim D_2}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i]$.

We will compute the probabilities:

a. $A = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i]$.
b. $B = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) = y_i]$.
c. $C = P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i]$.
d. $D = P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) = y_i]$.
e. $E = P_{i\sim D_1}[h_1(x_i) \neq h_2(x_i) \; and \; h_3(x_i) \neq y_i]$.
f. $F = P_{i\sim D_1}[H(x_i) \neq y_i]$.

The evidence:

a. $A = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i] = \sum_{h_1(x_i)\neq y_i \; and \; h_2(x_i)\neq y_i} D_1(i) = \sum_{h_1(x_i)\neq y_i \; and \; h_2(x_i)\neq y_i} D_2(i)(1 + y_i h_1(x_i) * (1 - 2 * \epsilon_1)) = \sum_{h_1(x_i)\neq y_i \; and \; h_2(x_i)\neq y_i} D_2(i)(1 + (-1) * (1 - 2 * \epsilon_1)) = \sum_{h_1(x_i)\neq y_i \; and \; h_2(x_i)\neq y_i} D_2(i) * 2 * \epsilon_1 = 2 * \epsilon_1 * \sum_{h_1(x_i)\neq y_i \; and \; h_2(x_i)\neq y_i} D_2(i) = 2 * \epsilon_1 * P_{i\sim D_2}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i] = 2 * a * \epsilon_1.$

b. Since $\epsilon_1 = P_{i\sim D_1}[h_1(x_i) \neq y_i] = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i] + P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) = y_i] = A + B \rightarrow B = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) = y_i] = \epsilon_1 - A = \epsilon_1 * (1 - 2 * a).$

c. $C = P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i] = \sum_{h_1(x_i)=y_i \; and \; h_2(x_i)\neq y_i} D_1(i) = \sum_{h_1(x_i)=y_i \; and \; h_2(x_i)\neq y_i} D_2(i)(1 + y_i h_1(x_i)) * (1 - 2 * \epsilon_1) = \sum_{h_1(x_i)=y_i \; and \; h_2(x_i)\neq y_i} D_2(i)(1 + 1 * (1 - 2 * \epsilon_1)) = \sum_{h_1(x_i)=y_i \; and \; h_2(x_i)\neq y_i} D_2(i) * 2 * (1 - \epsilon_1) = 2 * (1 - \epsilon_1) * \sum_{h_1(x_i)=y_i \; and \; h_2(x_i)\neq y_i} D_2(i) = 2 * (1 - \epsilon_1) * P_{i\sim D_2}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i]$. But we know that $\epsilon_2 = P_{i\sim D_2}[h_2(x_i) \neq y_i] = P_{i\sim D_2}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i] + P_{i\sim D_2}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i]$, so it follows that $C = 2 * (1 - \epsilon_1) * (\epsilon_2 - a)$.

d. We know that $1 - \epsilon_1 = P_{i\sim D_1}[h_1(x_i) = y_i] = P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i] + P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) = y_i] = C + D \rightarrow D = 1 - \epsilon_1 - C.$

e. In addition, from $\epsilon_3 = P_{i\sim D_3}[h_3(x_i) \neq y_i] = \sum_{h_3(x_i)\neq y_i} D_3(i) =$ $\sum_{h_3(x_i)\neq y_i \; and \; h_1(x_i)\neq h_2(x_i)} \frac{D_1(i)}{Z} +$ $\sum_{h_3(x_i)\neq y_i \; and \; h_1(x_i)= h_2(x_i)} 0 = \sum_{h_3(x_i)\neq y_i \; and \; h_1(x_i)\neq h_2(x_i)} \frac{D_1(i)}{Z} = \frac{E}{Z} \rightarrow E = Z * \epsilon_3$. Now, $Z = \sum_{h_1(x_i)\neq h_2(x_i)} D_1(i) = P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) = y_i] + P_{i\sim D_1}[h_1(x_i) = y_i \; and \; h_2(x_i) \neq y_i] = B + C$. This implies that $E = \epsilon_3 * (B + C) = -2 * a * \epsilon_3 + \epsilon_1 * \epsilon_3 + 2 * \epsilon_2 * \epsilon_3 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3$.

f. $F = P_{i\sim D_1}[H(x_i) \neq y_i] = P_{i\sim D_1}[h_1(x_i) \neq h_2(x_i) \; and \; h_3(x_i) \neq y_i] +$ $P_{i\sim D_1}[h_1(x_i) \neq y_i \; and \; h_2(x_i) \neq y_i] = A + E = 2 * a * \epsilon_1 +$ $\epsilon_3[\epsilon_1 * (1 - 2 * a) + 2 * (1 - \epsilon_1) * (\epsilon_2 - a)] = 2 * a * (\epsilon_1 - \epsilon_3) + \epsilon_1 * \epsilon_3 + 2 * \epsilon_2 * \epsilon_3 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3$. From c. we have that $\epsilon_2 \geq a$ (otherwise $C < 0$, which would be absurd for a probability).

If we prove that $F \leq 3 * \epsilon_{max}^2 - 2 * \epsilon_{max}^3$, where $\epsilon_{max} = \max(\epsilon_1, \epsilon_2, \epsilon_3)$, then we can use the fact that $\epsilon_{max} \leq \frac{1}{2} - \gamma_{min}$(since $\epsilon_i \leq \frac{1}{2} - \gamma_i \leq \frac{1}{2} - \gamma_{min}$, $i \in \{1,2,3\}$) to get the conclusion. Depending on the sign of $\epsilon_1 - \epsilon_3$, we have that either:

- $F \leq \epsilon_1 * \epsilon_3 + 2 * \epsilon_2 * \epsilon_3 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3$ (when $\epsilon_1 \leq \epsilon_3$).
- $F \leq 2 * \epsilon_2 * (\epsilon_1 - \epsilon_3) + \epsilon_1 * \epsilon_3 + 2 * \epsilon_2 * \epsilon_3 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3 = \epsilon_1 * \epsilon_3 + 2 * \epsilon_1 * \epsilon_2 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3$ (when $\epsilon_1 > \epsilon_3$).

These two cases are similar, so, we have to prove that $\epsilon_1 * \epsilon_3 + 2 * \epsilon_2 * \epsilon_3 - 2 * \epsilon_1 * \epsilon_2 * \epsilon_3 = \epsilon_3 * (\epsilon_1 + 2 * \epsilon_2 - 2\epsilon_1 * \epsilon_2) \leq 3 * \epsilon_{max}^2 - 2 * \epsilon_{max}^3 = \epsilon_{max} * (3 * \epsilon_{max} - 2 * \epsilon_{max}^2)$. Of course, since $\epsilon_i \leq \epsilon_{max}$, $i \in \{1,2,3\}$ obviously holds, and the quantity $(\epsilon_1 + 2 * \epsilon_2 - 2\epsilon_1 * \epsilon_2)$ is obviously positive (considering that $0 \leq \epsilon_i < \frac{1}{2}, i \in \{1,2,3\}$ and notice

that $2 * \epsilon_2 \geq 2 * \epsilon_1 * \epsilon_2$), it would be enough to prove that $(\epsilon_1 + 2 * \epsilon_2 - 2\epsilon_1 * \epsilon_2) \leq 3 * \epsilon_{max} - 2 * \epsilon_{max}^2$. Since $\epsilon_{max} = \max(\epsilon_1, \epsilon_2, \epsilon_3)$, there exist $r_i \geq 0, i \in \{1,2,3\}$ such that $\epsilon_i = \epsilon_{max} - r_i, i \in \{1,2,3\}$. The left expression be converted into: $\epsilon_1 + 2 * \epsilon_2 - 2\epsilon_1 * \epsilon_2 = (\epsilon_{max} - r_1) + 2 * (\epsilon_{max} - r_2) - 2 * (\epsilon_{max} - r_1) * (\epsilon_{max} - r_2) = 3 * \epsilon_{max} - 2 * \epsilon_{max}^2 - X$, where $X = r_1 + 2 * r_2 - 2 * \epsilon_{max} * (r_1 + r_2) + 2 * r_1 * r_2$. But $\epsilon_{max} < \frac{1}{2}$, so $X \geq r_1 + 2 * r_2 - 2 * \frac{1}{2} * (r_1 + r_2) + 2 * r_1 * r_2 = r_2 + 2 * r_1 * r_2 \geq 0$, so the desired inequality is proved. Now we proved that $P_{i \sim D_1}[H(x_i) \neq y_i] \leq 3 * \epsilon_{max}^2 - 2 * \epsilon_{max}^3$. But the function $f(x) = 3 * x^2 - 2 * x^3$, which has $f'(x) = 6 * x - 6 * x^2 = 6 * x * (1 - x) > 0$ for $x \in \left(0, \frac{1}{2}\right)$ is strictly increasing, so we have that $P_{i \sim D_1}[H(x_i) \neq y_i] \leq 3 * \epsilon_{max}^2 - 2 * \epsilon_{max}^3 = f(\epsilon_{max}) \leq f\left(\frac{1}{2} - \gamma_{min}\right) = \frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3$, as required to prove that $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3 < \frac{1}{2} - \gamma_{min}$, $\Rightarrow \frac{1}{2} * \gamma_{min} > 2 * \gamma_{min}^3$, or $\gamma_{min} < \frac{1}{2}$, which is obvious from $\left(0 < \epsilon_i \leq \frac{1}{2} - \gamma_i \leq \frac{1}{2} - \gamma_{min} \rightarrow \gamma_{min} \in \left(0, \frac{1}{2}\right)\right)$.

So, since $\gamma_i > 0$, $i \in \{1,2,3\} \rightarrow \gamma_{min} = min_{i \in \{1,2,3\}} \gamma_i > 0$.

Question – 4

Consider $\mathcal{H}_{2DNF}^{d}$the class of 2-term disjunctive normal form ….

Starting from the definition, a learning algorithm A is a γ-weak-learner for a class $\mathcal{H}$ if there exist a function $m_{\mathcal{H}}: (0,1) \longrightarrow \mathbb{N}$ such that: for every $\delta > 0$, for every labeling function $f \in \mathcal{H}, f: \mathcal{X} \to \{-1, +1\}$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, when we run the learning algorithm A on a training set, consisting of $m > m_{\mathcal{H}}(\delta)$ examples sampled i.i.d from $\mathcal{D}$ and labeled by f, the algorithm A return a hypothesis h (h might not be from $\mathcal{H}$- improper learning) such that, with probability at least $1 - \delta$ (over the choice of examples), $L_{\mathcal{D},f} \leq 1/2 - \gamma$.

Using the distribution rule, we can transform the 2-turm DNF formula to a 2-CNF formula: $A_1 \vee A_2 = \bigwedge_{u \in A_1, v \in A_2} (u \vee v) = \bigwedge_{u \in A_1, v \in A_2} (y_{u,v})$

Now, we gain a conjunction of $(2n)^2$ variables, each of them being a disjunction of 2 literals from the original problem. This conjunction can be efficiently PAC learned.

Now we want a learning algorithm A_{weak} that drops one variable from the conjunction without losing generality, $y_{u,v}$, and learns to predict the conjunction of the remaining y type variables. We have to prove that this is a γ-weak-learner for a class $\mathcal{H}_{2DNF}^{d}$.

By removing the variable $y_{u,v} = (u \vee v)$ from the conjunction, it is equivalent with saying that we assume its value is $+1$. The chance to be incorrect (both u and v is -1) is $1/4$ ($1/2 \times 1/2$, both literals having 50% chance to be positive).

Thus, we return the answer of the conjunction problem for the remaining of the $(2n)^2 - 1$ variables, having introduced an additional 1/4 chance of error to the accuracy of this predictor. We know that C_n, the concept class of conjunctions of at

most n Boolean literals is PAC learnable with sample complexity $m_{\mathcal{H}}(\epsilon, \delta) = \left\lceil \frac{1}{\epsilon}(nlog(3) - \log(\delta)) \right\rceil$.

This results in, $L_{\mathcal{D}}(A_{weak}(S)) \leq 1/4$ (introduced by the variable elimination) $+ \epsilon$.

We want $L_{\mathcal{D}}(A_{weak}(S)) < 1/2 - \gamma$.

Take ϵ such that $1/4 + \epsilon < 1/2 \implies \epsilon < 1/4$. Then, the ERM algorithm for solving the resulted conjunction of Booleans (whit one dropped variable) is a γ-weak-learner for a class $\mathcal{H}_{2DNF}^{d}$, where $\gamma = \epsilon$.

Going back to the definition, we proved that for every $\delta > 0$, for every labeling function $f \in \mathcal{H}, f: \mathcal{X} \longrightarrow \{-1, +1\}$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, we have the function $m_{\mathcal{H}}(\gamma, \delta) = \left\lceil \frac{1}{\gamma}((2n)^2 log(3) - \log(\delta)) \right\rceil$ such that when we run the algorithm A_{weak} on $m > m_{\mathcal{H}}(\delta)$ examples sampled i.i.d from $\mathcal{D}$ and labeled by f, the algorithm A_{weak} return a hypothesis h (h might not be from $\mathcal{H}$- improper learning) such that, with probability at least $1 - \delta$ (over the choice of examples), $L_{\mathcal{D},f} \leq 1/2 - \gamma$. (for γ values $< 1/4$). Therefore, our A_{weak} algorithm is a γ-weak-learner algorithm for learning the class $\mathcal{H}_{2DNF}^{d}$.