Advanced Machine Learning - Assignment 2

Iordachescu Anca Mihaela

June 2022

1 Exercise 1

$$\mathcal{H} = \{h_{a,b,s} : \mathcal{R} \to \{-1, 1\}, a \leq b, s \in \{-1, 1\}\}$$

where

$$h_{a,b,s}(x) = \begin{cases} s, x \in [a, b] \\ -s, x \notin [a, b] \end{cases}$$

We can say $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ where

$$\mathcal{H}_1 = \{h_{a,b,1} : \mathcal{R} \to \{-1, 1\}, a \leq b \}$$

$$\mathcal{H}_2 = \{h_{a,b,-1} : \mathcal{R} \to \{-1, 1\}, a \leq b \}$$

Let m be an integer number, $m \geq 0$ and C a set of m points, C = $\{c_1, c_2, ..., c_m\}$ with $c_i < c_j$. The shattering coefficient can be computed as follows:

$$\tau_{\mathcal{H}} = |\mathcal{H}_{C1} \cap \mathcal{H}_{C2}| = |\mathcal{H}_{C1}| + |\mathcal{H}_{C2}| - |\mathcal{H}_{C1} \cap \mathcal{H}_{C2}|$$

with

$$\mathcal{H}_{C1} = \{-1^i 1^j -1^k\} | i + j + k = m, 0 \leq i, j, k \leq m\}$$

(using $\mathcal{H}_1$)

$$\mathcal{H}_{C2} = \{1^i -1^j 1^k\} | i + j + k = m, 0 \leq i, j, k \leq m\}$$

(using $\mathcal{H}_2$)

It is easy to see that $|\mathcal{H}_{C1}| = |\mathcal{H}_{C2}|$ because the two sets are symmetrical (by changing 1 to -1 and -1 to 1 in the $\mathcal{H}_{C1}$, we obtain $\mathcal{H}_{C2}$ and vice-versa).

So, the shattering coefficient $\tau_{\mathcal{H}} = |\mathcal{H}_{C1} \cup \mathcal{H}_{C2}| = 2 * |\mathcal{H}_{C1}| - |\mathcal{H}_{C1} \cap \mathcal{H}_{C2}|$.

First of all, we have to compute $|\mathcal{H}_{C1}| = |\{-1^i 1^j -1^k\} | i + j + k = m, 0 \leq i, j, k \leq m\}|$. This number $|\mathcal{H}_{C1}|$ represents how many arrays that contains a contiguous subarray of 1s (having a j size) can be produced to be used for labelling.

If j = 0, this means that the labelling consists only of -1: (-1, -1, ..., -1) - 1 labelling.

If j = 1, this means that there is a subarray of 1s that have the size equal to 1 in a labelling vector of size m: (1, -1, -1, ..., -1), (-1, 1, -1, ..., -1) etc - m labellings.

If j = 2, this means that there is a subarray of 1s that have the size equal to 2 in a labelling vector of size m: (1, 1, -1, ..., -1), (-1, 1, 1, ..., -1) etc - m-1 labellings.

...

If j = m, this means that there is a subarray of 1s that have the size equal to m in a labelling vector of size m: (1, 1, 1, ..., 1) - 1 labelling.

So, $|\mathcal{H}_{C1}| = 1 + \text{m} + (\text{m-1}) + ... + 1 = 1 + \frac{m*(m+1)}{2}$.

The same process can be applied to $|\mathcal{H}_{C2}|$, resulting (as I said above) that $|\mathcal{H}_{C1}| = |\mathcal{H}_{C2}|$.

The next step consists in computing $|\mathcal{H}_{C1} \cap \mathcal{H}_{C2}|$. The intersection of the two sets consists in labelling that looks like $[-1^a, 1^{m-a}]$ or $[1^a, -1^{m-a}]$ with $a \geq 1$.

- $(-1^a, 1^{m-a})$ can be produced by both $h_{a+1,m,1}$ (using $\mathcal{H}_1$) and $h_{1,a,-1}$ (using $\mathcal{H}_2$).
- $(1^a, -1^{m-a})$ can be produced by both $h_{a+1,m,-1}$ (using $\mathcal{H}_2$) and $h_{1,a,1}$ (using $\mathcal{H}_1$).

So, $\mathcal{H}_{C1} \cap \mathcal{H}_{C2} = \{(-1^a, 1^{m-a}), m \geq a > 0\} \cup \{(1^a, -1^{m-a}), m \geq a > 0\}$.

Let A be equal to $\{(-1^a, 1^{m-a}), m \geq a > 0\}$ and B be equal to $\{(1^a, -1^{m-a}), m \geq a > 0\}$.

$|\mathcal{H}_{C1} \cap \mathcal{H}_{C2}| = |A \cup B| = |A| + |B|$ - $|A \cap B| = m + m - 0 = 2m$.

$|A \cap B|$ is equal to 0 because for each a, $m \geq a > 0$ we obtain these labellings and there are no overlaps between them:

$$(-1, -1, -1, -1, \underbrace{1...}_{a}, 1) \in A$$

$$(1, 1, 1, 1, \underbrace{-1...}_{a}, -1) \in B$$

In the end, $\tau_{\mathcal{H}} = |\mathcal{H}_{C1} \cup \mathcal{H}_{C2}| = 2 * |\mathcal{H}_{C1}| - |\mathcal{H}_{C1} \cap \mathcal{H}_{C2}| = 2 + m * (m + 1) - 2m = m^2 - m + 2$.

b. First of all, I am going to prove that VCdim($\mathcal{H}$) = 3.

1. *There exists a set C of size 3 that is shattered by $\mathcal{H}$.*

For instance, I choose $\mathcal{H}^* = \{h_{0,2}, h_{-1,-1,1}, h_{3,3,1}, h_{2,2,1}, h_{1,1,1}, h_{2,3,1}, h_{1,2,1}, h_{2,2,-1}, h_{1,3,1}\}$ and C = {1, 2, 3}. It is obvious that $\mathcal{H}$ shatters C because I can obtain every possible configuration of labelling: [-1, -1, -1], [-1, -1, 1], [-1, 1, -1], [1, -1, -1], [-1, 1, 1], [1, 1, -1], [1, -1, 1], [1, 1, 1].

2. *Every set A of size 4 is not shattered by $\mathcal{H}$.* There is no set that can generate [-1,1,-1,1] because due to the fact that $\mathcal{H}$ is the class of 3-piece classifier, no matter how we choose a, b and s, the labelling contains a subarray that contains only elements equal to s and outside the subarray are elements that are equal to -s, meaning that, as we iterate through the labelling vector, the number of pairs that consists in two consecutive numbers that are not equal

is at most 2. Here, in the labelling [-1,1,-1,1] the number described above is 3 which is greater than 2 and it can't be generated by $\mathcal{H}$.

Now, that we know that VCdim($\mathcal{H}$) = 3, I can use the general upper bound for the growth function: $\tau_{\mathcal{H}} \leq \sum_{n\in\{0,1,..,d\}} C_m^n$ where d = VCdim($\mathcal{H}$) = 3.

Using the lemma, we get the following inequality:

$\tau_{\mathcal{H}} \leq \sum_{n\in\{0,1,2,3\}} C_m^n = 1 + m + \frac{m(m-1)}{2} + \frac{m(m-1)(m-2)}{6} \Leftrightarrow$

$m^2 - m + 2 \leq 1 + m + \frac{m(m-1)}{2} + \frac{m(m-1)(m-2)}{6} \Leftrightarrow$

$m(m-1) \leq (m-1) + \frac{m(m-1)}{2} + \frac{m(m-1)(m-2)}{6} \Leftrightarrow$

$0 \leq (m-1) - \frac{m(m-1)}{2} + \frac{m(m-1)(m-2)}{6} \Leftrightarrow$

$0 \leq 6(m-1) - 3m(m-1) + m(m-1)(m-2) \Leftrightarrow$

$0 \leq (m-1)(6 - 3m + m(m-2)) \Leftrightarrow$

$0 \leq (m-1)(m^2 - 5m + 6)) \Leftrightarrow$

$0 \leq (m-1)(m-2)(m-3)$

At the previous exercise, I got $\tau_{\mathcal{H}} = m^2 - m + 2$ which is different from the right side of the inequality. The inequality is strict for $m > 3$ and the inequality is not strict (equality) for m = 1, m = 2 and m = 3.

For m = 0, however, the general upper bound is less than the shatter coefficient.

c. Let $\mathcal{H} = \{h_a : R \to 0, 1, h_a(x) = 1_{[x<a]}, a \leq 0\}$ be the set of threshold functions over the real line presented in the *Lecture 7*.

First of all, let m be an integer number, $m \geq 0$ and C a set of m points, C = $\{c_1, c_2, ..., c_m\}$ with $c_i < c_j$ in order to compute $\tau_{\mathcal{H}}$.

The only possible labelling than can be obtained using the $\mathcal{H}$ belongs to $\{(1^a, 0^{m-a}),\ a \geq 0\}$. So, $\tau_{\mathcal{H}} = |\{(1^a, 0^{m-a}),\ a \geq 0\}| = m + 1$.

The next step is proving that VCdim($\mathcal{H}$) = 1.

1. *There exists a set C of size 1 that is shattered by $\mathcal{H}$.*

For instance, I choose $\mathcal{H}^* = \{h_3,\ h_{-1}\}$ and C = {1}. It is obvious that $\mathcal{H}$ shatters C because I can obtain every possible configuration of labelling: [1], [0].

2. *Every set A of size 2 is not shattered by $\mathcal{H}$.*

Let A = {a,b} with $a \leq b$. We can obtain the next labelling: [0, 0] (using h_{a-1}), [1,0] (using $h_{\frac{a+b}{2}}$), [1, 1] (using h_{b+1}). We can't obtain [0, 1] because if there is a function h_c^* such that $h_c^*(b) = 1$, due to the fact that $a \leq b$ and $h_c^*(a)$ should also be equal to 1. (contradiction)

Knowing that VCdim($\mathcal{H}$) = 1 and using the general upper bound for the growth function: $\tau_{\mathcal{H}} \leq \sum_{n\in\{0,1,..,d\}} C_m^n$ where d = VCdim($\mathcal{H}$) = 1.

$\tau_{\mathcal{H}} \leq \sum_{n\in\{0,1\}} C_m^n \Leftrightarrow m + 1 \leq C_m^0 + C_m^1 \Leftrightarrow m + 1 \leq 1 + m.$

Looking at the last inequality, it is obvious that the shattering coefficient is equal to the general upper bound.

2 Exercise 2

$$\mathcal{H} = \{h_{a,b,c,d} : \mathcal{R} \to \{-1, 1\}, a \leq b \leq c \leq d\}$$

where

$$h_{a,b,c,d}(x) = \begin{cases} 1, x \in [a,b] \cup [c,d] \\ 0, x \notin [a,b] \cup [c,d] \end{cases}$$

(!) Before writing the algorithms for realizable and agnostic case, I am going to prove that VCdim($\mathcal{H}$) = 4. I prove this in order to use the fundamental theorem of statistical learning for each case (realizable/agnostic).

1. *There exists a set C of size 4 that is shattered by $\mathcal{H}$.*

For instance, I choose and C = {a,b,c,d} with $a \leq b \leq c \leq d$. It is obvious that $\mathcal{H}$ shatters C because I can obtain every possible configuration of labelling ($2^4 = 16$):

- [0,0,0,0] → choose $h_{a-1,a-1,a-1,a-1}$
- [0,0,0,1] → choose $h_{d,d,d,d}$
- [0,0,1,0] → choose $h_{c,c,c,c}$
- [0,1,0,0] → choose $h_{b,b,b,b}$
- [1,0,0,0] → choose $h_{a,a,a,a}$
- [1,1,0,0] → choose $h_{a,b,b,b}$
- [1,0,1,0] → choose $h_{a,a,c,c}$
- [1,0,0,1] → choose $h_{a,a,d,d}$
- [0,1,0,1] → choose $h_{b,b,d,d}$
- [0,0,1,1] → choose $h_{c,d,d,d}$
- [0,1,1,0] → choose $h_{b,c,c,c}$
- [0,1,1,1] → choose $h_{b,d,d,d}$
- [1,0,1,1] → choose $h_{a,a,c,d}$
- [1,1,0,1] → choose $h_{a,b,d,d}$
- [1,1,1,0] → choose $h_{a,c,c,c}$
- [1,1,1,1] → choose $h_{a,d,d,d}$

2. *Every set A of size 5 is not shattered by $\mathcal{H}$.*

Due to the fact that the labelling looks like $0^i1^j0^k1^m0^n$ with i+j+k+m+n=5 (in this case), there is no way we can label a set of 5 points with the label [1,0,1,0,1].

a. realizable case

There exists a function $h_{a^*,b^*,c^*,d^*} \in \mathcal{H}$ that labels the training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, with $y_i = h_{a^*,b^*,c^*,d^*}(x_i)$.

We can have the following possibilities for examples appearing in S:

$$+ + + + + + + + + + + + + +$$

$$- - - - - - - - - - - - - -$$

$$- - - + + + - - + + - - -$$

$$- - - - + + + - - - + + +$$

$$+ + + + - - - - + + + - -$$

$$++++++----++++$$
$$-----++++----$$
$$+++++++------$$
$$------++++++++$$

Consider the following algorithm:

1. The first step is ascending sorting the training set S by x' in order to obtain $S = S' = \{(x_{\sigma_1}, y_{\sigma_1}), ..., (x_{\sigma_m}, y_{\sigma_m})\}$ where $x_{\sigma_1} \leq x_{\sigma_2} \leq ... \leq x_{\sigma_m}$.

2.i. If there are only positive examples, we can compute $a^* = \min_{1 \leq i \leq m, y_i = 1} x_i = x_{\sigma_1}$, $b^* = \max_{1 \leq i \leq m, y_i = 1} x_i = x_{\sigma_m}$, $c^* = d^* = b^*$.

2.ii. If there are only negative examples, we can compute $z = \min_{1 \leq i \leq m, y_i = 0} x_i = x_{\sigma_1}$, $a^* = b^* = c^* = d^* = z - 1$.

2.iii. If there are both positive and negative examples, we first compute $a^* = \min_{1 \leq i \leq m, y_i = 1} x_i = x_{\sigma_{min}}$ and $d^* = \max_{1 \leq i \leq m, y_i = 1} x_i = x_{\sigma_{max}}$ where min and max are the indexes for the minimum and maximum element in the sorted order.

We initialize $b^* = -\infty$ and $c^* = -\infty$.

Iterate through S conditioned by $min < \sigma_i < max$ (because b^* and c^* should be greater than a^* and less than d^*) and check if:

- if the previous element ($x_{\sigma_{i-1}}$) has the label +1 and the current element (x_{σ_i}) has the label 0, then $b^* = x_{\sigma_{i-1}}$.

- if the previous element ($x_{\sigma_{i-1}}$) has the label 0 and the current element (x_{σ_i}) has the label 1, then $c^* = x_{\sigma_i}$.

If b^* and c^* are $-\infty$ after this iteration, this means that in between the elements a^* and d^* there are only elements that have positive labels or there are not any elements between them. In this case, we set $b^* = c^* = d^*$.

3. Return h_{a^*, b^*, c^*, d^*}

Complexity:

1. Sorting - $O(m * log(m))$

2.i. Computing a^*, b^*, c^*, d^* in the case of only positives - $O(m)$

2.ii. Computing a^*, b^*, c^*, d^* in the case of only negatives - $O(m)$

2.iii. Computing a^*, b^*, c^*, d^* in the case of both positives and negatives - $O(m)$

Total Complexity: $O(m * log(m))$

Using the fundamental theorem of statistical learning –quantitative version (presented in *Lecture 9*) and knowing that VCdim($\mathcal{H} = 4$), $\mathcal{H}$ is PAC-learnable with sample complexity $C_1 * \frac{4 + log(\frac{1}{\delta})}{\epsilon} \leq m_{\mathcal{H}(\epsilon, \delta)} \leq C_2 * \frac{4 * log(\frac{1}{\epsilon}) + log(\frac{1}{\delta})}{\epsilon}$, $C_1, C_2 > 0$ (the complexity depends on $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$).

b.agnostic case

Example for agnostic case in *Figure 1*:

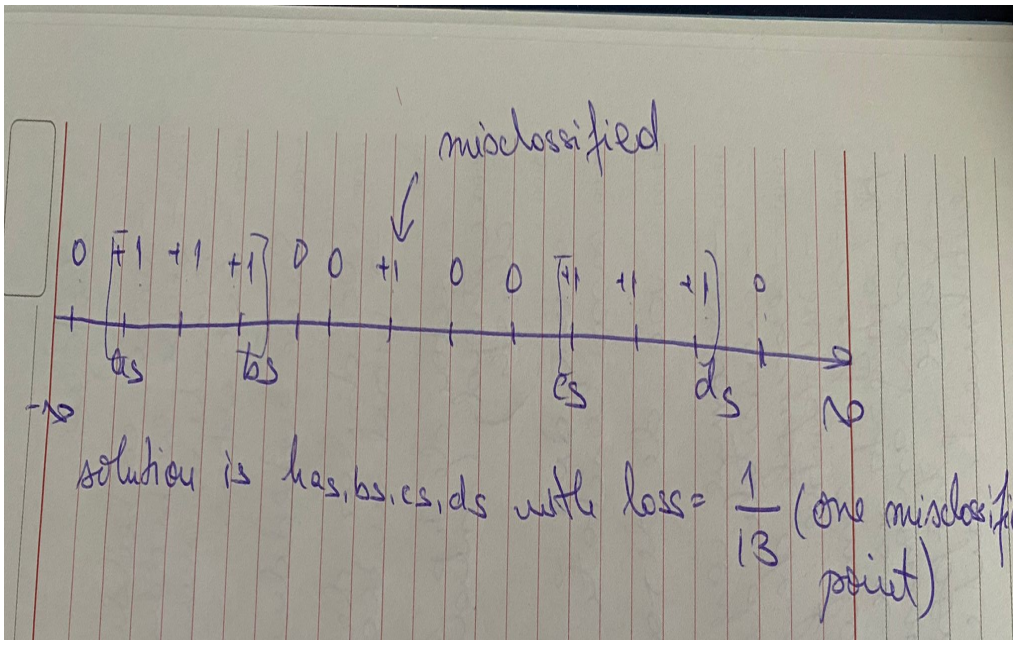

Figure 1. Example for agnostic case: 13 points shattered with loss $\frac{1}{13}$.

Consider the following algorithm:

1. The first step is ascending sorting the training set S by x' in order to obtain $S = S' = \{(x_{\sigma_1}, y_{\sigma_1}), ..., (x_{\sigma_m}, y_{\sigma_m})\}$ where $x_{\sigma_1} \leq x_{\sigma_2} \leq ... \leq x_{\sigma_m}$. Due to the fact that we are in the agnostic case, we can have $x_{\sigma_i} = x_{\sigma_{i+1}}$ and $y_i \neq y_{i+1}$.

2. Construction of the set Z as consisting in unique values of x' from the training set S. $Z = \{z_1, z_2, ..., z_n\}$ where $z_1 = x_{\sigma_1} < z_2 < ... < z_n = x_{\sigma_m}$ and $n \leq m$.

3. Compute P = number of positive examples = $\sum_{i \in 1,..,m} y_i$.

4.i. If there are only negative examples, we can compute we can compute $z = \min_{1 \leq i \leq m, y_i = 0} x_i = x_{\sigma_1} = z_1$, $a^* = b^* = c^* = d^* = z - 1$.

4.ii.1. If not, due to the fact that we are in the agnostic case, we can have $z_j = x_{k_1}, x_{k_2}, ..., x_{k_l}$ (l points) and we have to compute p_j as number of points with label 1 and n_j as number of points with label 0.

For $j = 1..n$

compute p_j = number of points x_i equal to z_i that have label $y_i = 1$

compute n_j = number of points x_i equal to z_i that have label $y_i = 0$

4.ii.2. Consider all possible intervals $Z_{a,b,c,d} = [z_a, z_b] \cup [z_c, z_d]$ with $a, b, c, d \in \{1, 2, ..., n\}$ and determine the $Z^* = Z_{a^*,b^*,c^*,d^*}$ with the smallest empirical risk (meaning $Z_{a^*,b^*,c^*,d^*} = arg \min_{a,b,c,d \in \{1,..,n\}} Loss(Z_{a,b,c,d})$.

We can compute fast $Z_{a,b,c,d}$ by using dynamic programming.

$Loss(Z_{a,b,c,d}) = \frac{\#\text{negative points inside } Z_{a,b,c,d} + \#\text{positive points outside } Z_{a,b,c,d}}{m}$

$Loss(Z_{a,b,c,d}) = \frac{\#\text{negative points inside } [z_a, z_b] and [z_c, z_d] + \#\text{positive points outside } [z_a, z_b] and [z_c, z_d]}{m}$

So, in order to compute $Loss(Z_{a,b,c,d})$ we have to know how many negative points are inside $[z_a, z_b]$, how many negative points are inside $[z_c, z_d]$, how many positive points are outside $[z_a, z_b] \cup [z_c, z_d]$. A naive method to find out these numbers of occurrences would be to iterate all z_i and counter those that appear in the intervals described above. This method has a linear complexity, making the algorithm complexity bigger, but I found a better method that has constant complexity.

Having p and n vectors, I can use them in order to compute partial sums (the sum of all elements of a vector from the beginning until a index). In this way, it is easier to compute, for example, the number of negative points inside $[z_a, z_b]$ because, having the partial sums, the number would be the partial sum of negative points of z_b - the partial sum of negative points of z_{a-1}. The same stands for the other intervals.

For $j = 2..n$

compute $p_j = p_{j-1} + p_j$

compute $n_j = n_{j-1} + n_j$

4.ii.3. Iterate through all indexes $a, b, c, d \in \{1, 2, .., n\}$ to find out the minimum loss.

$min.error$ = 1, a^* = -1, b^* = -1, c^* = -1, d^* = -1

for a $\in$ {1,2,..n}

  for b $\in$ {1,2,..n}

    for c $\in$ {1,2,..n}

      for d $\in$ {1,2,..n}

        partial.loss.neg = $n_b - n_{a-1} + n_d - n_{c-1}$

        partial.loss.poss = $p_b - p_{a-1} + p_d - p_{c-1}$

        actual.loss = partial.loss.neg + P - partial.loss.poss

        if actual.loss < min.error

          min.error = actual.loss

          $a^* = a$

          $b^* = b$

          $c^* = c$

          $d^* = d$

5. Return h_{a^*,b^*,c^*,d^*}

Complexity:

1. Sorting - $O(m * log(m))$

2 Construction of Z - $O(m)$

3. Computing P - $O(m)$.

4.i. Computing a^*, b^*, c^*, d^* in the case of only negatives - $O(m)$

4.ii.1. Compute p_i and n_i - $O(m)$

4.ii.2 Compute partial sums - $O(m)$, calculate $Loss(Z_{a,b,c,d})$ - constant time, calculate min.error and find out best a, b, c, d - $O(m^4)$

Total Complexity: $O(m^4)$

Using the fundamental theorem of statistical learning –quantitative version (presented in *Lecture 9*) and knowing that VCdim($\mathcal{H} = 4$), $\mathcal{H}$ is agnostic PAC-

learnable with sample complexity $C_1 * \frac{4+log(\frac{1}{\delta})}{\epsilon^2} \leq m_{\mathcal{H}(\epsilon,\delta)} \leq C_2 * \frac{4+log(\frac{1}{\delta})}{\epsilon^2}$, $C_1, C_2 > 0$ (the complexity depends on $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$).

3 Exercise 3

For resolving this exercise, I am going to use all relation written in the *Lecture 11, slides 36-38*.

a. I am going to demonstrate that the probability of the classifier h_1 to be chosen in the second round is equal to 0.

Before starting resolving the exercise, I am going to enumerate some relations that I used during computations:

1. $\sum_{h_1(x_i)\neq y_i} D^{(1)}(i) = e_1$

2. $\sum_{h_1(x_i)=y_i} D^{(1)}(i) = 1 - \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) = 1 - e_1$

3. $Z_2 = \sum_{i=1}^{n} D^{(1)}(i) * e^{-w_1 h_1(x_i) y_i} = \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) * e^{-w_1 h_1(x_i) y_i} + \sum_{h_1(x_i)=y_i} D^{(1)}(i) * e^{-w_1 h_1(x_i) y_i} = \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) * \sqrt{\frac{1-\epsilon_1}{\epsilon_1}} + \sum_{h_1(x_i)=y_i} D^{(1)}(i) * \sqrt{\frac{\epsilon_1}{1-\epsilon_1}} = \sqrt{\frac{1-\epsilon_1}{\epsilon_1}} * \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) + \sqrt{\frac{\epsilon_1}{1-\epsilon_1}} * \sum_{h_1(x_i)=y_i} D^{(1)}(i) = \sqrt{\frac{1-\epsilon_1}{\epsilon_1}} * e_1 + \sqrt{\frac{\epsilon_1}{1-\epsilon_1}} * (1 - e_1) = \sqrt{\epsilon_1 * (1-\epsilon_1)} + \sqrt{\epsilon_1 * (1-\epsilon_1)} = 2 * \sqrt{\epsilon_1 * (1-\epsilon_1)}$

In order to prove this, I am going to compute the error e_2 obtained by the weak classifier h_2 which is equal to the weak classifier h_1.

$e_2 = Pr_{i\sim D^{(2)}}[h_2(x_i) \neq y_i] = Pr_{i\sim D^{(2)}}[h_1(x_i) \neq y_i] = \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) * \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} * \frac{1}{Z_2} = \frac{1}{Z_2} * \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} * \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) \overset{\text{rel.1+3}}{=} \frac{1}{2*\sqrt{\epsilon_1*(1-\epsilon_1)}} * \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} * \epsilon_1 = \frac{\epsilon_1}{2*\sqrt{\epsilon_1}*\sqrt{\epsilon_1}} = \frac{1}{2}$.

The error achieved by weak classifier h_1 used as weak classifier h_2 is $\frac{1}{2}$. However, according to *Lecture 7*, in weak learnability we have to output a classifier whose error rate is at most $\frac{1}{2} - \gamma, \gamma > 0$. In the second round, $e_2 \leq \frac{1}{2} - \gamma_2, \gamma_2 > 0 \Leftrightarrow \frac{1}{2} \leq \frac{1}{2} - \gamma_2, \gamma_2 > 0 \Leftrightarrow 0 \leq \gamma_2, \gamma_2 > 0$ which is clearly a contradiction.

This means that the weak classifier h_1 can't be selected in the second round, the probability of selecting h_1 in the second round being equal to 0.

b. In order to resolve this exercise, I have found in the book *Boosting: Foundations and Algorithms by Robert E. Schapire, Yoav Freund*, chapter *Using AdaBoost to Minimize Training Error* some hints: first, compute the probabilities notated below as $pr_1, pr_2, pr_3, pr_4, pr_5, pr_6$, then show that training error is at most $3 * e_{max}^2 - 3 * e_{max}^3, e_{max} = max(e_1, e_2, e_3)$, and then show that training error is at most $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3$.

Before starting resolving the exercise, I am going to enumerate some relations that I used during computations (plus the one in the course):

1. $w_t = \frac{1}{2} * ln(\frac{1}{e_t} - 1)$

2. Z normalization factor such that $D^{(3)}$ is a probability distribution $\rightarrow Z = Pr_{i\sim D_1}[h_1(x_i) = y_i \text{ and } h_2(x_i) \neq y_i] + Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) = y_i]$

3. $\sum_{h_1(x_i)\neq y_i} D^{(1)}(i) = e_1$

4. $\sum_{h_1(x_i)=y_i} D^{(1)}(i) = 1 - \sum_{h_1(x_i)\neq y_i} D^{(1)}(i) = 1 - e_1$

5. $Pr_{i\sim D_1}[h_1(x_i) \neq y_i] = Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) = y_i] + Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i]$

6. Compute $D^{(2)}(i) = \frac{D^{(1)}(i) * e^{w_1 h_1(x_i) y_i}}{Z_2}$

Knowing that $Z_2 = 2*\sqrt{e_1 * (1 - e_1)}$ and $w_1 = \frac{1}{2} * ln(\frac{1}{e_1} - 1) = ln(\sqrt{\frac{1}{e_1} - 1}) = ln(\sqrt{\frac{1-e_1}{e_1}})$, the above relations becomes:

$D^{(2)}(i) = \frac{1}{2} * \frac{1}{\sqrt{e_1 * (1-e_1)}} * D^{(1)}(i) * e^{-1 * ln(\sqrt{\frac{1-e_1}{e_1}}) * h_1(x_i) y_i} = \frac{1}{2} * \frac{1}{\sqrt{e_1 * (1-e_1)}} * D^{(1)}(i) * \sqrt{\frac{1-e_1}{e_1}}^{-1 * h_1(x_i) y_i} = D^{(1)}(i) * \frac{1}{2} * \frac{1}{\sqrt{e_1 * (1-e_1)}} * \sqrt{\frac{e_1}{1-e_1}}^{h_1(x_i) y_i} = D^{(1)}(i) * \frac{1}{2} * \frac{1}{\sqrt{e_1^{1-h_1(x_i) y_i}}} * \frac{1}{\sqrt{1-e_1^{1+h_1(x_i) y_i}}}$

We know that $h_1(x_i)$ can be either 1 or -1 and y_i can be either 1 or -1 $\Leftrightarrow$ $h_1(x_i) * y_i$ can be either 1 or -1.

If $h_1(x_i) * y_i = 1 \Leftrightarrow D^{(2)}(i) = D^{(1)}(i) * \frac{1}{2} * \frac{1}{1-e_1} \Leftrightarrow D^{(1)}(i) = 2 * (1-e_1) * D^{(2)}(i)$

If $h_1(x_i) * y_i =$ -1 $\Leftrightarrow D^{(2)}(i) = D^{(1)}(i) * \frac{1}{2} * \frac{1}{e_1} \Leftrightarrow D^{(1)}(i) = 2 * e_1 * D^{(2)}(i)$

Next, I am going to compute the next probabilities that are presented in the book *Boosting: Foundations and Algorithms* cited above. The main reason that I do that is because the last probability (probability that H is incorrect with regard to D_1) is the training error using H and I want to show that this value is at most $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3$. Below it can be seen that I am going to write all the probabilities $pr_2, pr_3, pr_4, pr_5, pr_6$ depending on pr_1 which represents the probability that both h_1 and h_2 are incorrect with regard to D_2.

Let pr_1 be the probability that both h_1 and h_2 are incorrect with regard to D_2, defining $pr_1 = Pr_{i\sim D_2}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i] = \sum_{h_1(x_i)\neq y_i \& h_2(x_i)\neq y_i} D^{(2)}(i)$

i. pr_2 - probability that both h_1 and h_2 are incorrect with regard to D_1, defining $pr_2 = Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i]$

$pr_2 = Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i] = \sum_{i\in\{1,..,m\}} D^{(1)}(i) * [1_{h_1(x_i)\neq y_i \& h_2(x_i)\neq y_i}] = \sum_{h_1(x_i)\neq y_i \& h_2(x_i)\neq y_i} D^{(1)}(i) = \sum_{h_1(x_i)\neq y_i \& h_2(x_i)\neq y_i} 2 * e_1 * D^{(2)}(i) = 2 * e_1 * \sum_{h_1(x_i)\neq y_i \& h_2(x_i)\neq y_i} D^{(2)}(i) = 2 * e_1 * pr_1$

Due to the fact that $pr_2 \geq 0$ (because it is a probability), this means that: $e_1 \geq 0$ and $pr_1 \geq 0$.

ii. pr_3 - probability that h_1 is incorrect and h_2 is correct with regard to D_1, defining $pr_3 = Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) = y_i]$

$pr_3 = Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) = y_i] = Pr_{i\sim D_1}[h_1(x_i) \neq y_i] - Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i] = e_1 - pr_2 = e_1 - 2*e_1*pr_1 = 1(e_1 - 2*pr_1)$

Due to the fact that $pr_3 \geq 0$ (because it is a probability), this means that: $e_1 \geq 2 * pr_1$.

iii. pr_4 - probability that h_1 is correct and h_2 is incorrect with regard to D_1, defining $pr_4 = Pr_{i\sim D_1}[h_1(x_i) = y_i \text{ and } h_2(x_i) \neq y_i]$

$pr_4 = Pr_{i\sim D_1}[h_1(x_i) = y_i \text{ and } h_2(x_i) \neq y_i] = \sum_{i\in\{1,..,m\}} D^{(1)}(i) * [1_{h_1(x_i)=y_i \& h_2(x_i)\neq y_i}] = \sum_{h_1(x_i)=y_i \& h_2(x_i)\neq y_i} D^{(1)}(i) = \sum_{h_1(x_i)=y_i \& h_2(x_i)\neq y_i} 2 * (1 - e_1) * D^{(2)}(i) = 2 * (1 - e_1) * \sum_{h_1(x_i)=y_i \& h_2(x_i)\neq y_i} D^{(2)}(i) = 2 * (1 - e_1) * (Pr_{i\sim D_2}[h_2(x_i) \neq y_i] - Pr_{i\sim D_2}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i]) = 2 * (1 - e_1) * (e_2 - pr_1)$

Due to the fact that $pr_4 \geq 0$ (because it is a probability), this means that: $e_2 \geq pr_1$ and $1 \geq e_1$.

iv.pr_5 - probability that h_1 and h_2 give different results and h_3 is incorrect with regard to D_1, defining $pr_5 = Pr_{i\sim D_1}[h_1(x_i) \neq h_2(x_i) \text{ and } h_3(x_i) \neq y_i]$

$\sum_{h_1(x_i)\neq h_2(x_i) \& h_3(x_i)\neq y_i} \frac{D^{(1)}}{Z} = Pr_{i\sim D_3}[h_3(x_i) \neq y_i]$ - $\sum_{h_1(x_i)=h_2(x_i) \& h_3(x_i)\neq y_i} 0 = Pr_{i\sim D_3}[h_3(x_i) \neq y_i]$ - $0 = Pr_{i\sim D_3}[h_3(x_i) \neq y_i] = e_3 \Leftrightarrow$

$\frac{1}{Z} * \sum_{h_1(x_i)\neq h_2(x_i) \& h_3(x_i)\neq y_i} D^{(1)} = e_3 \Leftrightarrow \frac{1}{Z} * pr_5 = e_3$

Z (normalization factor) $= Pr_{i\sim D_1}[h_1(x_i) = y_i \text{ and } h_2(x_i) \neq y_i] + Pr_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) = y_i] = e_1 - 2 * e_1 * pr_1 + 2 * (1 - e_1) * (e_2 - pr_1) = e_1 - 2 * pr_1 * e_1 + 2 * e_2 - 2 * pr_1 - 2 * e_1 * e_2 + 2 * e_1 * pr_1 = e_1 + 2 * e_2 - 2 * e_1 * e_2 - 2 * pr_1$

$pr_5 = Z * e_3 = e_3 * (e_1 + 2 * e_2 - 2 * e_1 * e_2 - 2 * pr_1) = e_1 * e_3 + 2 * e_2 * e_3 - 2 * pr_1 * e_3 - 2 * e_1 * e_2 * e_3$.

v.pr_6 - probability that and H is incorrect with regard to D_1, defining $pr_6 = Pr_{i\sim D_1}[H(x_i) \neq y_i]$

$pr_6 = Pr_{i\sim D_1}[H(x_i) \neq y_i] = \text{Pr}_{i\sim D_1}[h_1(x_i) \neq y_i \text{ and } h_2(x_i) \neq y_i] + \text{Pr}_{i\sim D_1}[h_1(x_i) \neq h_2(x_i) \text{ and } h_3(x_i) \neq y_i] = pr_2 + pr_5 = 2 * e_1 * pr_1 + e_1 * e_3 + 2 * e_2 * e_3 - 2 * pr_1 * e_3 - 2 * e_1 * e_2 * e_3$

Now, an idea that leads to the inequality we want to achieve is to prove that the final classifier H obtain a training error that is at most $3 * e_{max}^2 - 2 * e_{max}^3$ with $e_{max} = max(e_i), i \in \{1, 2, 3\}$

$pr_6 = -2 * e_1 * e_2 * e_3 + 2 * e_2 * e_3 + e_1 * e_3 + pr_1(2 * e_1 - 2 * e_3)$ and we can have two following cases depending on $\text{pr}_1(2 * e_1 - 2 * e_3)$:

- $e_1 - e_3 \leq 0$ and $e_2 \geq pr_1 \geq 0 \Rightarrow pr_1(2 * e_1 - 2 * e_3) \leq 0 \Rightarrow pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_2 * e_3$

  This means that we have to show that $pr_1 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_2 * e_3 \leq 3 * e_{max}^2 - 2 * e_{max}^3$

  $-2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_2 * e_3 \leq 3 * e_{max}^2 - 2 * e_{max}^3 \Leftrightarrow$

  $e_3(-2 * e_1 * e_2 + e_1 + 2 * e_2) \leq e_{max}(3 * e_{max} - 2 * e_{max}^2) \Leftrightarrow$

$-2 * e_1 * e_2 + e_1 + 2 * e_2 \leq 3 * e_{max} - 2 * e_{max}^2$ and $e_{max} \geq e_3$

The second inequality is true and the first one has to be proven. Due to the fact that the above inequality relation has 4 unknown variables. Due to the fact that $e_{max} = max(e_i), i \in \{1, 2, 3\}$, we can write $e_1 + a = e_{max}, e_2 + b = e_{max}, e_3 + c = e_{max}$ with$a, b, c \geq 0$.

$-2 * e_1 * e_2 + e_1 + 2 * e_2 = -2 * (e_{max} - a)(e_{max} - b) + e_{max} - a + 2(e_{max} - b) = -2 * e_{max} + 2 * (a + b) * e_{max} - 2 * ab + 3e_{max} - a - 2b = 3 * e_{max} - 2 * e_{max}^2 - 2ab - a - 2b + 2 * (a + b) * e_{max} \leq$(since $e_{max} < \frac{1}{2}$) $3 * e_{max} - 2 * e_{max}^2 - 2ab - a - 2b + 2 * (a + b) * \frac{1}{2} = 3 * e_{max} - 2 * e_{max}^2 - 2ab - a - 2b + a + b = 3 * e_{max} - 2 * e_{max}^2 - 2ab - b \leq 3 * e_{max} - 2 * e_{max}^2$ (since $2ab + b \geq 0$).

Now, that we proved that $pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_2 * e_3 \leq 3 * e_{max}^2 - 2 * e_{max}^3$, I have to prove that the final classifier H obtain a training error that is at most $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + \gamma_{min}^3$.

- $e_1 - e_3 > 0$ and $e_2 \geq pr_1 \geq 0 \rightarrow pr_1(2 * e_1 - 2 * e_3) > 0 \Rightarrow pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + e_2(2 * e_1 - 2 * e_3) + 2 * e_2 * e_3 \Leftrightarrow pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_1 * e_2 - 2 * e_3 * e_2 + 2 * e_2 * e_3 \Leftrightarrow pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_1 * e_2$

  This means that we have to show that $pr_1 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_1 * e_2 \leq 3 * e_{max}^2 - 2 * e_{max}^3$

  $-2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_1 * e_2 \leq 3 * e_{max}^2 - 2 * e_{max}^3 \Leftrightarrow$

  $e_1(-2 * e_3 * e_2 + e_3 + 2 * e_2) \leq e_{max}(3 * e_{max} - 2 * e_{max}^2) \Leftrightarrow$

  $-2 * e_3 * e_2 + e_3 + 2 * e_2 \leq 3 * e_{max} - 2 * e_{max}^2$and $e_{max} \geq e_1$

  The second inequality is true and the first one has to be proven. Due to the fact that the above inequality relation has 4 unknown variables. Due to the fact that $e_{max} = max(e_i), i \in \{1, 2, 3\}$, we can write $e_1 + a = e_{max}, e_2 + b = e_{max}, e_3 + c = e_{max}$ with $a, b, c \geq 0$.

  $-2 * e_3 * e_2 + e_3 + 2 * e_2 = -2 * (e_{max} - c)(e_{max} - b) + e_{max} - c + 2(e_{max} - b) = -2 * e_{max} + 2 * (c + b) * e_{max} - 2 * bc + 3e_{max} - c - 2b = 3 * e_{max} - 2 * e_{max}^2 - 2bc - c - 2b + 2 * (c + b) * e_{max} \leq$(since $e_{max} < \frac{1}{2}$) $3 * e_{max} - 2 * e_{max}^2 - 2bc - c - 2b + 2 * (c + b) * \frac{1}{2} = 3 * e_{max} - 2 * e_{max}^2 - 2bc - c - 2b + c + b = 3 * e_{max} - 2 * e_{max}^2 - 2cb - b \leq 3 * e_{max} - 2 * e_{max}^2$ (since $2bc + b \geq 0$).

  Now, that we proved that $pr_6 \leq -2 * e_1 * e_2 * e_3 + e_1 * e_3 + 2 * e_1 * e_2 \leq 3 * e_{max}^2 - 2 * e_{max}^3$, I have to prove that the final classifier H obtain a training error that is at most $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + \gamma_{min}^3$.

We know that for each $i \in \{1, 2, 3\}$, $e_i \leq \frac{1}{2} - \gamma_i, \gamma_i > 0$. That means that, using $\gamma_{min} = min(\gamma_i)$, each $e_i \leq \frac{1}{2} - \gamma_{min}$. That leads to $e_{max} \leq \frac{1}{2} - \gamma_{min}$.

Looking at the *Figure 2* that shows the graph of function $f : R \rightarrow R, f(x) = 3 * x^2 - 2 * x^3$, it is obvious that f(x) is a monotone increasing function on interval

$[0, \frac{1}{2}]$. This means that whatever $x_1, x_2 \in [0, \frac{1}{2}], x_1 \leq x_2 \rightarrow f(x_1) \leq f(x_2)$ and I can write that $pr_6 \leq 3 * e_{max}^2 - 2 * e_{max}^3 \leq f(e_{max}) \leq f(\frac{1}{2} - \gamma_{min}) = 3 * (\frac{1}{2} - \gamma_{min})^2 - 2 * (\frac{1}{2} - \gamma_{min})^3 = 3 * (\frac{1}{4} - \gamma_{min} + \gamma_{min}^2) - 2(\frac{1}{8} - 3 * \gamma_{min} * \frac{1}{4} + 3 * \gamma_{min}^2 * \frac{1}{2} - \gamma_{min}^3) = \frac{3}{4} - 3 * \gamma_{min} + 3 * \gamma_{min}^2 - \frac{1}{4} + 3 * \gamma_{min} * \frac{1}{2} - 3 * \gamma_{min}^2 + 2 * \gamma_{min}^3 = \frac{1}{2} - 3 * \gamma_{min} + \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3 = \frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3$. This means that the training error on the final classifier H is at most $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3$.

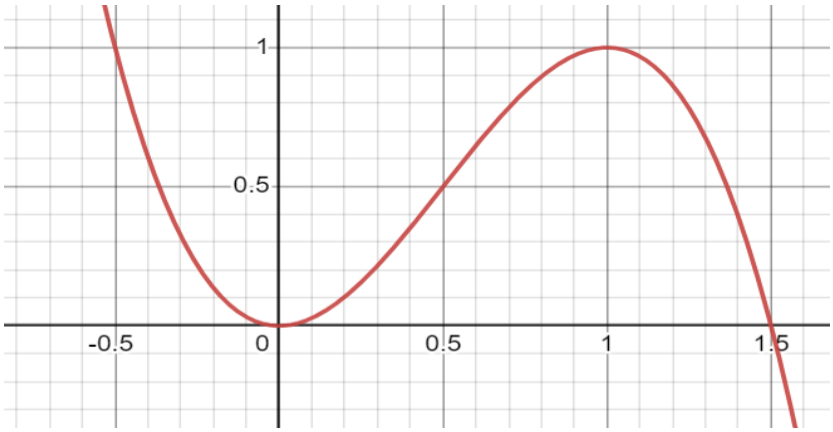

Figure 2. Graph of the function $f(x) = 3 * x^2 - 2 * x^3$.

The next exercise is to prove that $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3 < \frac{1}{2} - \gamma_{min}$.
$\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3 < \frac{1}{2} - \gamma_{min} \Leftrightarrow$
$\frac{3}{2} * \gamma_{min} - 2 * \gamma_{min}^3 - \gamma_{min} > 0 \Leftrightarrow$
$\frac{1}{2} * \gamma_{min} - 2 * \gamma_{min}^3 > 0 \Leftrightarrow$
$\gamma_{min}(\frac{1}{2} * -2 * \gamma_{min}^2) > 0$

We already know that $\gamma_{min} > 0$ and also want to prove that $0 < \frac{1}{2} - 2 * \gamma_{min}^2$ by resolving $0 < \gamma_{min} < \frac{1}{2} \Leftrightarrow 0 < \gamma_{min}^2 < \frac{1}{4} \Leftrightarrow -\frac{1}{4} < -\gamma_{min}^2 < 0 \Leftrightarrow -\frac{1}{2} < -2 * \gamma_{min}^2 < 0 \Leftrightarrow 0 < \frac{1}{2} - 2 * \gamma_{min}^2 < \frac{1}{2}$. Because $\gamma_{min} > 0$ and $0 < \frac{1}{2} - 2 * \gamma_{min}^2$ are true, I proved that $\frac{1}{2} - \frac{3}{2} * \gamma_{min} + 2 * \gamma_{min}^3 < \frac{1}{2} - \gamma_{min}$.

4 Exercise 4

According to *Lecture 10-slide 11*, we can convert a 2-term DNF to 2-term CNF by applying the distribution rule.

This means that I can write that $A_1 \vee A_2 = \bigwedge_{a \in A_1, b \in A_2} (a \vee b) = \bigwedge_{a \in A_1, b \in A_2} y_{a,b}$ with A_1 and A_2 being Boolean conjunctions of literals in $\mathcal{H}_{2DNF}^d$. Now, we can treat 2-term DNF as a conjunction of $(2n)^2$ variables.

Let's consider the pair that is within the conjunction of $(2n)^2$ variables that is formed of the first literal that appears in A_1 (named x) and the first literal that appears in A_2 (named w). The pair $(x \vee w) = y_{x,w} \in \bigwedge_{a \in A_1, b \in A_2} y_{a,b}$, being

one of the $(2n)^2$ variables that constructs the conjunction. $(x \vee w) = y_{x,w}$ can take two values, $y_{x,w} \in \{0, 1\}$ since $x, w \in \{0, 1\}$.

In order to achieve an algorithm that is γ-weak-learner, called A, I am going to use the conjunction that I constructed above with an exception: the disjunction between x and w = $(x \vee w) = y_{x,w}$ is always going to be 1 ($y_{x,w} = 1$).

We know that $x, w \in \{0, 1\}$.
- x = 0 and w = 0 $\rightarrow (x \vee w) = y_{x,w} = 0$
- x = 0 and w = 1 $\rightarrow (x \vee w) = y_{x,w} = 1$
- x = 1 and w = 0 $\rightarrow (x \vee w) = y_{x,w} = 1$
- x = 1 and w = 1 $\rightarrow (x \vee w) = y_{x,w} = 1$

If we compute the disjunction between u and w, we can see that in 3 cases the disjunction is equal to 1 and in one case the disjunction is equal to 0. This means that the chance of error (how many times it made a mistake divided by the number of all cases) is equal to $1 - \frac{3}{4} = \frac{1}{4}$.

According to *Lecture 10-slide 30*, $H^d_{2DNF} \subseteq H^{(2n)^2}_{conj}$ and we can learn efficiently PAC-learn $H^{(2n)^2}_{conj}$ with sample complexity:

m$(\delta, \epsilon) = \frac{1}{\epsilon} * [-log(\delta) + (2n)^2 * log(3)]$.

Now, looking at the example that can be found in *Lecture 11-slide 29*, I am going to compute $L_D(A(S)) \leq \frac{1}{4} + \epsilon$. Take ϵ such that $\frac{1}{4} + \epsilon < \frac{1}{2} \Leftrightarrow \epsilon < \frac{1}{4}$. Then ERM_A is a γ-weak-learner for $H^{(2n)^2}_{conj}$ with $\epsilon = \gamma$.

Now, I am going to use the definition of weak learnability presented in *Lecture 11-slide 24* in order to prove that A is a γ-weak-learner :

A is a γ-weak-learner for the class $\mathcal{H}^d_{2DNF}$ because

- for every $\gamma > 0$
- for every labelling function $f \in \mathcal{H}^d_{2DNF}, f : X \rightarrow \{0, 1\}$
- for every distribution D over X

there is a function m$(\delta, \epsilon) = \frac{1}{\epsilon} * [-log(\delta) + (2n)^2 * log(3)]$ such that when we run the learning algorithm A on a training set that has $m \geq m(\delta, \epsilon)$ examples sampled from D and labeled by f, the algorithm A returns a hypothesis h such that $L_{D,f}(h) < \frac{1}{2} - \gamma$,with $\gamma < \frac{1}{4}$, probability at least $1 - \delta$ and $\epsilon = \gamma$.