Assigment II

Advance Machine Learning

Dăscălescu Dana

May 31, 2023

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

$$\mathcal{H} = \{h_{a,b,s} : \mathbb{R} \to \{-1, 1\} \mid a \leq b, s \in \{-1, 1\}\}, \text{ where } h_{a,b,s}(x) = \begin{cases} s, & x \in [a, b] \\ -s, & x \notin [a, b] \end{cases}$$

a. Compute the shattering coefficient $\tau_H(m)$ of the growth function for $m \geq 0$ for hypothesis class $\mathcal{H}$.

b. Compare your result with the general upper bound for the growth functions and show that $\tau_H(m)$ obtained at previous point a is not equal with the upper bound.

c. Does there exist a hypothesis class $\mathcal{H}$ for which the shattering coefficient $\tau_H(m)$ of the growth function for $m \geq 0$ is equal to the general upper bound (over $\mathbb{R}$ or another domain $\mathcal{X}$)? If your answer is yes please provide an example, if your answer is no please provide a justification.

Solution.

a. The growth function of a hypothesis class $\mathcal{H}$, denoted by $\tau_{\mathcal{H}}$, is defined as the maximum number of distinct functions from a set C of size m to $\{0, 1\}$, that can be obtained by restricting $\mathcal{H}$ to C. (*Chapter 6 in* [1])

$$\tau_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}, \qquad \tau_{\mathcal{H}} = \max_{C \subseteq \mathcal{X}:|C|=m} |\mathcal{H}_C|$$

Let $C \subset \mathbb{R}$ be a set of m points, $C = \{c_1, c_2, \ldots, c_m\}$. Without loss of generality, we will consider $c_1 < c_2 < \cdots < c_m$.

In general, the concept class assigns two different types of labels based on the s parameter (sequence of either 1 or -1, respectively, surrounded by two sequences of -1 or 1, respectively):

$$(-1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1)$$
$$(1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1)$$

Now, we will count how many possibilities of labeling we have for each case:

$\boxed{s = 1:}$ We will count the number of labelings for each sequence of i positive points, $i = \overline{0, m}$.

$i = 0$ (for the base case we have no positive label) : $(-1, -1, -1, \ldots, -1, -1, -1) \Rightarrow 1$ label
$i = 1 : (1, -1, -1, \ldots, -1, -1, -1), \ldots, (-1, -1, -1, \ldots, -1, -1, 1) \Rightarrow m$ labels
$i = 2 : (1, 1, -1, \ldots, -1, -1, -1), \ldots, (-1, -1, -1, \ldots, -1, 1, 1) \Rightarrow (m-1)$ labels
$\vdots$
$i = m-1 : (1, 1, 1, \ldots, 1, 1, -1),\ (-1, 1, 1, \ldots, 1, 1, 1) \Rightarrow 2$ labels
$i = m : (1, 1, 1, \ldots, 1, 1, 1) \Rightarrow 1$ label

In total, we can obtain $1 + m + (m-1) + \cdots + 2 + 1 = 1 + \dfrac{m(m+1)}{2}$ functions in the first case.

$\boxed{s = \text{-}1:}$ To count the possible labels that do not appear in the first case, we will count how many labelings we have for each sequence of i negative points surrounded by positive points on both sides, $i = \overline{1, m-2}$. In other words, we want to obtain the reunion of the all labelings sets for each case, minus their intersection.

$i = 1 : (1, -1, 1, \ldots, 1, 1, 1), (1, 1, -1, \ldots, 1, 1, 1), \ldots, (1, 1, 1, \ldots, 1, -1, 1) \Rightarrow (m-2)$ labels
$i = 2 : (1, -1, -1, \ldots, 1, 1, 1), \ldots, (1, 1, 1, \ldots, -1, -1, 1) \Rightarrow (m-3)$ labels
$\vdots$
$i = m-2 : (1, -1, -1, \ldots, -1, -1, 1) \Rightarrow 1$ label

In total, we can obtain $(m-2) + (m-3) + \cdots + 1 = \dfrac{(m-2)(m-1)}{2}$ functions in the second case.

Thus, $\tau_m = 1 + \frac{m(m+1)}{2} + \frac{(m-2)(m-1)}{2} = m^2 - m + 2$.

b. According to the Sauer-Shelah-Perles Lemma, for a hypothesis class $\mathcal{H}$ with $VCdim(\mathcal{H}) < d$ and all m, we have that $\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$. (*Chapter 6 in* [1])

First, we will calculate the Vapnik–Chervonenkis dimension of our hypothesis class $\mathcal{H}$ (*as in Seminar 3*).

Let's consider $\mathcal{C} = \{c_1, c_2, c_3\}$ a set of 3 distinct points with $c_1 < c_2 < c_3$.

$$\begin{aligned}
&\text{for label } (-1,-1,-1),\ \text{take } a = c_1 - 1,\ b = c_3 + 1,\ \text{and } s = -1 \\
&\text{for label } (-1,-1,1),\ \text{take } a = c_1,\ b = \frac{c_2 + c_3}{2},\ \text{and } s = -1 \\
&\text{for label } (-1,1,-1),\ \text{take } a = \frac{c_1 + c_2}{2},\ b = \frac{c_2 + c_3}{2},\ \text{and } s = 1 \\
&\text{for label } (1,-1,-1),\ \text{take } a = c_1,\ b = \frac{c_1 + c_2}{2},\ \text{and } s = 1 \\
&\text{for label } (-1,1,1),\ \text{take } a = \frac{c_1 + c_2}{2},\ b = c_3 + 1,\ \text{and } s = 1 \\
&\text{for label } (1,-1,1),\ \text{take } a = \frac{c_1 + c_2}{2},\ b = \frac{c_2 + c_3}{2},\ \text{and } s = -1 \\
&\text{for label } (1,1,-1),\ \text{take } a = c_1,\ b = \frac{c_2 + c_3}{2},\ \text{and } s = 1 \\
&\text{for label } (-1,-1,1),\ \text{take } a = c_1,\ b = \frac{c_2 + c_3}{2},\ \text{and } s = -1
\end{aligned}$$

$\mathcal{H}$ shatters C, so $VCdim(\mathcal{H}) \geq 3$. (*)

Now, we will show that $VCdim(\mathcal{H}) < 4$. Take a set of points $C = \{c_1, c_2, c_3, c_4\}$, and consider, with no loss of generality, $c_1 < c_2 < c_3 < c_4$. $\mathcal{H}$ cannot shatter the following labelling: $(-1, 1, -1, 1)$ (or we could also take $(1, -1, 1, -1)$). This holds for any C, $|C| = 4$. So $VCdim(\mathcal{H}) < 4$. (**)

$$(*)(**) \implies VCdim(\mathcal{H}) = 3$$

In our case, $VCdim(\mathcal{H}) = 3$, thus the general upper bound is:

$$C_m^0 + C_m^1 + C_m^2 + C_m^3 = 1 + m + \frac{m(m-1)}{2} + \frac{m(m-1)(m-2)}{6} = \frac{m^3 + 5m + 6}{6}$$

Therefore we have that $m^2 - m + 2 \leq \dfrac{m^3 + 5m + 6}{6},\ \forall m \in \mathbb{N} \Leftrightarrow$

$$\Leftrightarrow \frac{m^3 + 5m + 6}{6} - m^2 + m - 2 \geq 0 \,\Big|\cdot 6,\ \forall m \in \mathbb{N} \ \Leftrightarrow\ m^3 + 5m + 6 - 6m^2 + 6m - 12 \geq 0,\ \forall m \in \mathbb{N}$$

$$\Leftrightarrow m^3 - 6m^2 + 11m - 6 \geq 0,\ \forall m \in \mathbb{N} \ \overset{\text{observe that } m=1}{\underset{\text{is a solution}}{\Leftrightarrow}}\ (m-1)(m^2 - 5m + 6) \geq 0,\ \forall m \in \mathbb{N}$$

$$\Leftrightarrow (m-1)(m-2)(m-3) \geq 0,\ \forall m \in \mathbb{N}$$

m	0 1 2 3 ∞
m-1	- - - - - - 0 + + + + + + + + + + + + + + + + + +
m-2	- - - - - - - - - - - - - - 0 + + + + + + + + + + + + + +
m-3	- 0 + + + + + + + + + +
(m-1)(m-2)(m-3)	- - - - - - - 0 + + + 0 - - - - - - 0 + + + + + + + + + +

The solution of the inequality $m^2 - m + 2 \leq \dfrac{m^3 + 5m + 6}{6}$ is $[1, 2] \cup [3, \infty)$. So, we have that the general upper bound is greater or equal to the shatter coefficient found for subpoint a, for all

$m \in \mathbb{N}, m \in [1,2] \cup [3,\infty] \implies m \in \{1,2,3,4,\dots\}$, and it is less than the shatter coefficient $\tau_{\mathcal{H}}$ of the growth function for $m = 0$.

c. Lets consider $\mathcal{H}$ the class of threshold functions over the real line:

$$\mathcal{H}_{thresholds} = \{h_\theta : \mathbb{R} \to \{0,1\},\ h_\theta(x) = \mathbb{1}_{[x<\theta]}(x),\ \theta \in \mathbb{R}\},\ \text{where}$$

$$\mathbb{1}_{[x<\theta]} = \begin{cases} 1, & x \in (-\infty, \theta] \\ 0, & otherwise \end{cases}$$

First, we will calculate the Vapnik–Chervonenkis dimension for our hypothesis class.

Consider $C = \{c_1\}$. Then $\mathcal{H}_C = \{h : C \to \{0,1\} \mid h \in \mathcal{H}\}$ has two elements $\{h_a, h_b\}$ with $a < c_1$ and $b \geq c_1$, so $\mathcal{H}$ shatters $C \implies VCdim(\mathcal{H}) \geq 1$. (1)

Let $C = \{c_1, c_2\}$ and without the loss of generality, we will consider $c_1 < c_2$. $\mathcal{H}_C$ does not shatter C, since we cannot label simultaneously the point c_1 negatively and c_2 positively (if c_1 has label 0, it means that $c_1 > \theta$ so no point $c > c_1$ can have label 1), so there is no function that realizes the labelling $(0,1) \implies VCdim(\mathcal{H}) < 2$. (2)

$$(1),(2) \implies VCdim(\mathcal{H}) = 1$$

Now, lets calculate the shattering coefficient of the growth function.

Let $C = \{c_1, c_2, \dots, c_m\}$ be a set of m points. Without loosing generality, consider $c_1 < c_2 < \dots < c_m$. Our hypothesis class can only output a sequence of ones followed by a sequence of zeros. Now, for $i = \overline{0,m}$, we count the number of labellings with a sequence of length i of positive points (because there are m elements, the length of the sequences of ones is between 0 and m):

$$\begin{aligned}
&i = 0 \text{ (we have no positive label)} : (0,0,0,\dots,0,0,0) \Rightarrow 1 \text{ label} \\
&i = 1 : (1,0,0,\dots,0,0,0) \Rightarrow 1 \text{ label} \\
&i = 2 : (1,1,0,\dots,0,0,0) \Rightarrow 1 \text{ label} \\
&\vdots \\
&i = m-1 : (1,1,1,\dots,1,1,0) \Rightarrow 1 \text{ label} \\
&i = m : (1,1,1,\dots,1,1,1) \Rightarrow 1 \text{ label}
\end{aligned}$$

$$\implies \tau_{\mathcal{H}(m)} = m + 1$$

Using the Sauer's lemma, we get that the general upper bound for our hypothesis class $\mathcal{H}$ with $VCdim(\mathcal{H}) = 1$ is $\sum_{i=0}^{1} C_m^i = C_m^0 + C_m^1 = \frac{m!}{0 \cdot m!} + \frac{m!}{1! \cdot m!} = 1 + m = \tau_{\mathcal{H}}(m)$. Thus, we showed that there exists a class H for which the shattering coefficient $\tau_{\mathcal{H}}(m)$ of the growth function is equal to the general upper bound.

2. Consider the concept class $\mathcal{C}_2$ formed by the union of two closed intervals $[a,b]\cup[c,d]$, where $a,b,c,d\in\mathbb{R},\ a\le b\le c\le d$. Give an efficient ERM algorithm for learning the concept class $\mathcal{C}_2$ and compute its complexity for each of the following cases:

a. realizability case
b. agnostic case

Solution.

$$\mathcal{H}=\left\{h_{a,b,c,d}:\mathbb{R}\to\{0,1\},h_{a,b,c,d}=\mathbb{1}_{[a,b]\cup[c,d]},h_{a,b,c,d}(x)=\left\{\begin{array}{ll}1 & if\ x\in[a,b]\cup[c,d]\\ 0 & otherwise\end{array}\right\}\right\}$$

Consider a training set S of size m:

$$S=\left\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\,\middle|\,x_i\in\mathbb{R},\ y_i\in\{0,1\},\ i=\overline{1,m}\right\}$$

In the following, we will present an ERM$_{\mathcal{H}}$ learning rule implementation in both the realizability and agnostic scenarios (similar to the approach took in the Seminar class 5), which is equivalent to finding the hypothesis h_{a_S,b_S,c_S,d_S} with the smallest empirical risk.

a. In the realizable case, we assume that there exists a function $h^*=h_{a^*,b^*,c^*,d^*}$ taken from the hypothesis class $\mathcal{H}$, which is known to the learner and that labels all the training points (i.e. $y_i=h^*(x_i),\ i=\overline{1,m}$).

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

$$\begin{array}{ll}+++++++++++++ & \text{(only positive examples)}\\ ------------- & \text{(only negative examples)}\\ ++++++------- & \\ -----++++++++ & \\ ++++-----++++ & \\ --+++--+++-- & \end{array}$$

Consider the following algorithm:

1. *Initialization:* $\quad a_S=b_S=-\infty,\quad c_S=d_S=\infty$
2. Sort S and obtain $S=\left\{(x_{\sigma(1)},y_{\sigma(1)}),(x_{\sigma(2)},y_{\sigma(2)}),\ldots,(x_{\sigma(m)},y_{\sigma(m)})\right\}$
3. If there are only positive points, return:

$$h_{a_S,b_S,c_S,d_S},\ \text{where } a_S=x_{\sigma(1)},\ b_S=c_S=d_S=x_{\sigma(m)}$$

If there are no positive examples, return:

$$h_{a_S,b_S,c_S,d_S},\ \text{where } a_S=x_{\sigma(1)}-2,\ d_S=x_{\sigma(1)}-1,\ b_S=a_S+e^{-5},\ c_S=d_S-e^{-5}$$

4.

$$a_S,\ left = \min_{\substack{i=\overline{1,m} \\ y_{\sigma(i)}=1}} x_{\sigma(i)},\ \operatorname*{argmin}_{\substack{i=\overline{1,m} \\ y_{\sigma(i)}=1}} x_{\sigma(i)}$$

$$d_S,\ right = \max_{\substack{i=\overline{1,m} \\ y_{\sigma(i)}=1}} x_{\sigma(i)},\ \operatorname*{argmax}_{\substack{i=\overline{1,m} \\ y_{\sigma(i)}=1}} x_{\sigma(i)}$$

$$b_S = a_S + e^{-5},\ c_S = d_S - e^{-5}$$

5. for $i = \overline{left+1, right}$

    if $y_{\sigma(i-1)} == 1$ and $y_{\sigma(i)} == 0$ then

        $b_S = x_{\sigma(i-1)}$

    if $y_{\sigma(i-1)} == 0$ and $y_{\sigma(i)} == 1$ then

        $c_S = x_{\sigma(i)}$

    return: h_{a_S,b_S,c_S,d_S}

Complexity:

1. Sorting the training set: $\mathcal{O}(m \cdot \log_2 m)$
2. Determining that there are either only positive labels or negative labels: $\mathcal{O}(m)$
3. Determining the ranges of the intervals: $\mathcal{O}(m)$
4. Adjusting the ends of the intervals (b and c values): $\mathcal{O}(m)$

Total: $\mathcal{O}(m \cdot \log_2 m)$

b. In this case, the realizability assumption is waived, which means that we no longer assume that all labels are generated by some $h^* \in \mathcal{H}$, but rather that we are dealing with a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ (the same point might have different labels), in our case $\mathcal{X} = \mathbb{R}$.

Idea of the implementation of $\text{ERM}_{\mathcal{H}}$

We begin by sorting the training set S in ascending order of x's, as we did in the realizable case. We obtain $S = \{(x_{\sigma(1)}, y_{\sigma(1)}), (x_{\sigma(2)}, x_{\sigma(2)}), \dots, (x_{\sigma(m)}, y_{\sigma(m)}) \mid x_{\sigma(1)} \leq \dots \leq x_{\sigma(m)}\}$.

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

$$Z = \{z_1, z_2 \dots, z_n\}, \qquad z_1 = x_{\sigma(1)} < z_2 < \dots < z_n = x_{\sigma(m)}, \qquad n \leq m$$

If all initial x values are different, then $z_1 = x_{\sigma(1)}, \dots, z_n = x_{\sigma(m)},\ n = m$.

In the following, we should consider the following cases:

1. If all $y_i = 0$, return an interval not containing any point x from the training set. Let's take as example: $a_S = z_1 - 2, d_S = z_1 - 1, b_S = a_S + e^{-5}, c_S = d_S - e^{-5}$.

2. Consider all possible reunions of intervals $Z_{i,j,k,l} = [z_i, z_j] \cup [z_k, z_l], i = \overline{1,n},\ j = \overline{i,n},\ k = \overline{j,n},\ l = \overline{k,n}$.

There are $n^2 + \cdots + 2 + 1 = \dfrac{n^2(n^2+1)}{2}$ such intervals.

An ERM algorithm will have to determine the intervals $Z^* = Z_{i^*,j^*,k^*,l^*}$ with the smallest empirical risk. $Z_{i^*,j^*,k^*,l^*} = \underset{\substack{i=\overline{1,m},\ j=\overline{1,m} \\ k=\overline{1,m},\ l=\overline{1,m}}}{\operatorname{argmin}} Loss(Z_{i,j,k,l})$, where

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

A dynamic programming approach will be used to efficiently compute $Loss(Z_{i,j,k,l})$. As a prerequisite, we must calculate the total number of positive points less than or equal to a given point z_i, as well as the total number of negative points less than or equal to a given point z_i. We will use two prefix-sum arrays for this (linear runtime complexity)[2]. We will use this partial results to calculate in constant time the number of positives or negative examples on any given interval $[i, j]$ with $i = \overline{1,m}, j = \overline{1,m}$. We will also need to pre-compute for each z_i the number of positive and negative points $x_\sigma(j)$, $i = \overline{1,n}$ (*as we did in Seminar class 5*), since we are in the agnostic case and can have $x_{\sigma(i)} = x_{\sigma(i+1)}$ and $y_{\sigma(i)}$.

$positive\_prefixSums[0] = 0$

for $i = \overline{1,m}$:

$positive\_prefixSums[i] = positive\_prefixSums[i-1] + \#$ points $z_i = x_j$ with label $y_i = 1$

$negative\_prefixSums[0] = 0$

for $i = \overline{1,m}$:

$negative\_prefixSums[i] = negative\_prefixSums[i-1] + \#$ points $z_i = x_j$ with label $y_i = 0$

For each selected reunion of intervals $[i, j] \cup [k, l],\ i = \overline{1,n},\ j = \overline{i,n},\ k = \overline{j,n},\ l = \overline{k,n}$, the loss function is calculated as follows:

$$Loss(Z_{i,j,k,l}) = \frac{positive\_prefixSums[n]}{m} - \frac{positive\_prefixSums[j] - positive\_prefixSums[i]}{m}$$

$$- \frac{positive\_prefixSums[l] - positive\_prefixSums[k]}{m}$$

$$+ \frac{negative\_prefixSums[j] - negative\_prefixSums[i]}{m}$$

Consider the following implementation of the $ERM_{\mathcal{H}}$ rule for $\mathcal{C}_2$:

1. Sort S and obtain $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(m)}$. 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 $a_S = z_1 - 2, d_S = z_1 - 1, b_S = a_S + e^{-5}, c_S = d_S - e^{-5}$

3. positive_prefixSums[0] = 0, negative_prefixSums[0] = 0

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

   if $y_i = 1$ then

   $positive\_prefixSums[i] = positive\_prefixSums[i{-}1] +$ # points $z_i = x_j$ with label

   if $y_i = 0$ then

   $negative\_prefixSums[i] = negative\_prefixSums[i{-}1] +$# points $z_i = x_j$ with label

4. $min\_error = 1,\ a_S = None,\ b_S = None,\ c_S = None,\ d_S = None$

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

   for $j = \overline{1, n}$:

   for $k = \overline{1, n}$:

   for $l = \overline{1, n}$:

$$
\begin{aligned}
Loss(Z_{i,j,k,l}) = & \frac{positive\_prefixSums[n]}{m} - \frac{positive\_prefixSums[j]}{m} \\
& + \frac{positive\_prefixSums[i]}{m} - \frac{positive\_prefixSums[l]}{m} \\
& + \frac{positive\_prefixSums[k]}{m} + \frac{negative\_prefixSums[j]}{m} \\
& - \frac{negatives\_prefixSums[i]}{m}
\end{aligned}
$$

   if $Loss(Z_{i,j,k,l}) < min\_error$ then

   $min\_error = Loss(Z_{i,j,k,l})$, $a_S = i$, $b_S = j$, $c_S = k$, $d_S = l$

   return a_S, b_S, c_S, d_S

Complexity:

1. Sorting the training set: $\mathcal{O}(m \cdot \log_2 m)$

2. Determining that there are only negative labels: $\mathcal{O}(m)$

3. Computing the pre-fix sums vectors: $O(m)$

4. Determining the ranges of the intervals: $\mathcal{O}(m^4)$

   - Constant time for computing the value of the loss function

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

3. Consider a modified version of the AdaBoost algorithm that runs exactly three rounds as follows:

- the first two rounds run exactly as in AdaBoost (at round 1 we obtain distribution $\mathbf{D}^{(1)}$, weak classifier h_1 with error ϵ_1; at round 2 we obtain distribution $\mathbf{D}^{(2)}$, weak classifier h_2 with error ϵ_2)
- in the third round we compute for each $i = 1, 2, \ldots, m$:

$$\mathbf{D}^{(3)}(i) = \begin{cases} \frac{D^{(1)}(i)}{Z}, & \text{if } h_1(x_i) \neq h_2(x_i) \\ 0, & \text{otherwise} \end{cases}$$

  where Z is a normalization factor such that $\mathbf{D}^{(3)}$ is a probability distribution.
- obtain weak classifier h_3 with error ϵ_3.
- output the final classifier $h_{final}(x) = sign(h_1(x) + h_2(x) + h_3(x))$.

Assume that at each round $t = 1, 2, 3$ the weak learner returns a weak classifier h_t for which the error ϵ_t satisfies $\epsilon_t \leq \frac{1}{2} - \gamma_t, \gamma_t > 0$.

a. What is the probability that the classifier h_1 (selected at round 1) will be selected again at round 2? Justify your answer.

b. Consider $\gamma = min\{\gamma_1, \gamma_2, \gamma_3\}$. Show that the training error of the final classifier h_{final} is at most $\frac{1}{2} - \frac{3}{2}\gamma + 2\gamma^3$ and show that this is strictly smaller than $\frac{1}{2} - \gamma$.

Solution. a. Assume that the classifier h_1 (selected in round 1) is chosen again in round 2. The error of the classifier h_1 with respect to the distribution $\mathbf{D}^{(2)}$ is:

$$\varepsilon_2' = \Pr_{i \sim \mathbf{D}^{(2)}}[h_1(x_i) \neq y_i] = \sum_{i=1}^{m} \mathbf{D}^{(2)}(i) \cdot \mathbb{1}_{[h_1(x) \neq y_i]} \tag{1}$$

where:

- $\mathbf{D}^{(2)}(i) = \dfrac{\mathbf{D}^{(1)}(i) \cdot e^{-w_1 \cdot h_1(x_i) \cdot y_i}}{Z_2}$
- $w_1 = \frac{1}{2} \cdot ln\left(\frac{1}{\varepsilon_1} - 1\right) = ln\left(\sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}}\right)$
- $Z_2 = \sum_{i=1}^{m} \mathbf{D}^{(1)}(i) \cdot e^{-w_t \cdot h_1(x_i) \cdot y_i}$

- for $h_1(x_i) = y_i \implies \mathbf{D}^{(2)}(i) = \dfrac{\mathbf{D}^{(1)}(i) \cdot e^{-w_1}}{Z_2} = \dfrac{\mathbf{D}^{(1)}(i) \cdot e^{-ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right)}}{Z_2} = \dfrac{\mathbf{D}^{(1)}(i) \cdot \sqrt{\dfrac{\varepsilon_1}{1-\varepsilon_1}}}{Z_2}$

- for $h_1(x_i) \neq y_i \implies \mathbf{D}^{(2)}(i) = \dfrac{\mathbf{D}^{(1)}(i) \cdot e^{w_1}}{Z_2} = \dfrac{\mathbf{D}^{(1)}(i) \cdot e^{ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right)}}{Z_2} = \dfrac{\mathbf{D}^{(1)}(i) \cdot \sqrt{\dfrac{1-\varepsilon_1}{\varepsilon_1}}}{Z_2}$

From the definition of Z_2 we obtain:

$$Z_2 = \sum_{i=1}^{m} \mathbf{D}^{(1)}(i) \cdot e^{-w_t \cdot h_1(x_i) \cdot y_i} = \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{m} \mathbf{D}^{(1)}(i) \cdot e^{-w_t \cdot h_1(x_i) \cdot y_i} + \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{m} \mathbf{D}^{(1)}(i) \cdot e^{-w_t \cdot h_1(x_i) \cdot y_i}$$

$$= (1-\varepsilon_1) \cdot \sqrt{\frac{\varepsilon_1}{1-\varepsilon_1}} + \varepsilon_1 \cdot \sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}} = 2\sqrt{\varepsilon_1(1-\varepsilon_1)}$$

Replacing these relations in equation 1 we obtain:

$$\varepsilon_2' = \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{m} \frac{\mathbf{D}^{(1)}(i) \cdot \sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}}}{2 \cdot \sqrt{\varepsilon_1 \cdot (1-\varepsilon)}} = \frac{\sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}}}{2 \cdot \sqrt{\varepsilon_1 \cdot (1-\varepsilon_1)}} \cdot \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{m} \mathbf{D}^{(1)}(i) = \frac{\sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}}}{2 \cdot \sqrt{\varepsilon_1 \cdot (1-\varepsilon_1)}} \cdot \varepsilon_1 =$$

$$= \frac{\sqrt{\varepsilon_1 \cdot (1-\varepsilon_1)}}{2 \cdot \sqrt{\varepsilon_1 \cdot (1-\varepsilon_1)}} = \frac{1}{2}$$

$\implies$ *contradiction with the hypothesis of the problem* $\varepsilon_2 \leq \dfrac{1}{2} - \gamma_2, \gamma_2 > 0$

$\implies$ classifier h_1 will always we replaced at round 2

Therefore, the probability of the classifier h_1 to be selected again at round 2 is 0.

In general, we can demonstrate that at each iteration j of the AdaBoost algorithm, the error of the classifier h_j with respect to the distribution $\mathbf{D}^{(j+1)}(j)$ equals $\frac{1}{2}$, implying that the probability of the classifier h_j being selected again at the next iteration is 0.

References

[1] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014. ISBN: 1107057132.

[2] Rohit Thapliyal. *Prefix Sum Array - Implementation and Applications in Competitive Programming - GeeksforGeeks. Available at:* `https://www.geeksforgeeks.org/prefix-sum-array-implementation-applications-competitive-programming/` . [Accessed 15 June 2022].