# Advanced Machine Learning - Assignment 2

Bouruc Petru-Liviu

14/06/2023

## 1 Exercise 1

*a*) To compute the growth function, we first have to find what is the VCdim of $\mathcal{H}$.

Consider C = $\{c\}$. Then, $\mathcal{H}_C$ has two elements, depending if $c$ is inside the interval ($-a \leq c \leq a$) or outside the interval. So, we can have both labelings (0 and 1), thus $|\mathcal{H}_C| = 2$ and VCdim($\mathcal{H}$) is at least 1.

Consider C = $\{c_1, c_2\}$. We have 3 possibilities:

- $|c_1| < |c_2|$: we cannot obtain the labeling (0, 1)
- $|c_1| = |c_2|$: we cannot obtain the labeling (0, 1) or (1, 0)
- $|c_1| > |c_2|$: we cannot obtain the labeling (1, 0)

In this case, $\mathcal{H}$ does not shatter $C$. So, VCdim($\mathcal{H}$) = 1.

We want to find out $\tau_{\mathcal{H}}(m) = \max_{C \subseteq X : |C| = m} |H_C|$. We know from Lecture 8 that if VCdim($\mathcal{H}$) = d, for any m $\leq$ d we have $\tau_{\mathcal{H}} = 2^m$.

Now let's consider C = $\{c_1, c_2, ..., c_m\}$ a set of m points, with $|c_i| < |c_j|, \forall i, j \leq m$. As in Lecture 8, $\mathcal{H}_C$ can have at most m+1 functions: we can take $|a_1| < |c_1| < |a_2| < |c_2| < ... < |a_m| < |c_m| < |a_{m+1}|$. Then $|\mathcal{H}_C| = \{h_{a_1}, h_{a_1}, ..., h_{a_{m+1}}\} = m + 1$ as:

- $h_{a_1}$ labels points $c_1, c_2, ..., c_m$ with labels (0,0,..,0,0)
- $h_{a_2}$ labels points $c_1, c_2, ..., c_m$ with labels (1,0,..,0.0)
- $h_{a_3}$ labels points $c_1, c_2, ..., c_m$ with labels (1,1,..,0,0)
- .....
- $h_{a_m}$ labels points $c_1, c_2, ..., c_m$ with labels (1,1,..,1,0)
- $h_{a_{m+1}}$ labels points $c_1, c_2, ..., c_m$ with labels (1,1,..,1,1)

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

*b*) We know from Sauer's Lemma that if we have $\text{VCdim}(\mathcal{H}) = \text{d} < \infty$, then $\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} C_m^i$.

$$\sum_{i=0}^{d} C_m^i = C_m^0 + C_m^1 = 1 + m$$

Corroborating with *a*), we have that $\tau_{\mathcal{H}}(m)$ is equal to the general upper bound given by the Sauer's lemma.

# 2 Exercise 2

For this exercise, we will take as guidance the problems 1 and 2 from seminar 5.

*a*) Realizable case.

There exists a function $h_{a^*}$ from the hypothesis class $\mathcal{H}$ that labels correctly the training points:

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

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

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

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

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

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

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

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

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

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

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

We consider the following algorithm:

1. Compute

$$st_+ = \min_{i=1..m, y_i=1} x_i$$

$$dr_+ = \max_{i=1..m, y_i=1} x_i$$

* if there are no +'s, we will return a value for $a$ such that the interval does not contain any points from the training set (for example $a = z_1 - 10$).

This can be done in $O(m)$.

2. We first have to take into consideration when all the positive values are in one interval. To check this, we can simply iterate one more time to see if there is an $x_i$ between $st_+$ and $dr_+$ with $y_i = 0$ ($O(m)$). If there is only one interval of +'s, we shall see if it fits into $[a, a+1]$ or in $[a+2, a+4]$ (using the values $st_+$ and $dr_+$) and set the $a$ correspondingly.

3. Now we are in the situation where we have the intervals separated from - values. Set $a = st_+$. We have to check if there exists an $x_i$ which is in between $[a, a+1]$ or $[a+2, a+4]$ and has $y_i = 0$. If it is, shift to the left the value of $a$ by the difference between $x_i$ and $a+1$ (or $a+4$), so that our negative point is not contained in the new interval. This can be done in $O(m)$, and done only once because there is necessary one iteration until we find the negative sample, knowing that we are in the realizable case.

4. return $a$

Total time complexity: $O(m)$

*b*) Agnostic case.

We will first sort the training set S in ascending order of $x$. We obtain the set $S = \{(x_{\sigma(1)}, y_{\sigma(1)}), (x_{\sigma(2)}, y_{\sigma(2)}), ..., (x_{\sigma(m)}, y_{\sigma(m)})\}$, with $x_{\sigma(1)} \leq x_{\sigma(2)} \leq ... \leq x_{\sigma(m)}$. Consider $Z$ be the set containing values of $x$ with no repetition.

$Z = \{z_1, z_2, ..., z_n\}, n \leq m$

We want to find an ERM algorithm that has the smallest empirical risk.

Observation: if all the labels are 0, then we can return an interval that does not contain any points from the training set (for example, $a = z_1 - 10$).

Considering the notations from above, we have to find the interval $Z_a = [z_a, z_{a+1}] \cup [z_{a+2}, z_{a+4}]$ with the smallest empirical risk. We can define the loss for our concept class as as:

$$Loss(Z_a) = \frac{\#negative\ points\ inside\ Z_a + \#positive\ points\ outside\ Z_a}{m}$$

We have to take into account that, because we are in the agnostic case, we can have the same point with different label. So, we need to compute for each $z_i$ two values, $z_i.pos$ (number of samples $x_i = z_i$ which have $y_i = 1$) and $z_i.neg$ (number of samples $x_i = z_i$ which have $y_i = 0$).

So, we have the following algorithm:

1. Sort the training set $O(m \cdot logm)$

2. Compute for each $z_i$ the *pos* and *neg* attributes. We can do this by iterating through each element from the training set in $O(m)$. In the same loop, we can compute an array $ps[i]$ representing the number of positive samples from the training set $S$ in the interval $z_1$ to $z_i$. Analogically, we compute $ns[i]$.

3. for $i = 1..n$: $O(m)$

   $Loss(Z_i) = \frac{ns[i+1]-ns[i-1]+ns[i+4]-ns[i+1]}{m} + \frac{ps[i-1]+ps[i+2]-ps[i+1]+ps[n]-ps[i+4]}{m}$

   if indexes are out of bounds, just take the beginning or the end of the set of points.

   through our iteration, make $a = i$, where $i$ is the index which gives the minimum loss.

4. return $a$

Total time complexity: $O(m \cdot logm)$

# 3 Exercise 3

*a*) We first start by writing down the equations from Lecture 11:

$$D^{(2)}(i) = \frac{D^{(1)}(i) \cdot e^{-w_1 h_1(x_i) y_i}}{Z_2}$$

$$Z_2 = \sum_{i=1}^{m} D^{(1)}(i) \cdot e^{-w_1 h_1(x_i) y_i}$$

$$w_1 = \frac{1}{2} ln(\frac{1}{\varepsilon_1} - 1) = ln(\sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}})$$

Using the last 2 equations, we have that:

$$Z_2 = \sum_{i=1}^{m} D^{(1)}(i) \cdot e^{-w_1 h_1(x_i) y_i} \cdot 1_{[h_1(i) \neq y_i]} + \sum_{i=1}^{m} D^{(1)}(i) \cdot e^{-w_1 h_1(x_i) y_i} \cdot 1_{[h_1(i) = y_i]}$$

$$Z_2 = e^{w_1} \sum_{i=1}^{m} D^{(1)}(i) \cdot 1_{[h_1(i) \neq y_i]} + e^{-w_1} \sum_{i=1}^{m} D^{(1)}(i) \cdot 1_{[h_1(i) = y_i]}$$

$$Z_2 = \sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}} \varepsilon_1 + \sqrt{\frac{\varepsilon_1}{1-\varepsilon_1}} (1 - \varepsilon_1)$$

$$Z_2 = \sqrt{\varepsilon_1(1-\varepsilon_1)} + \sqrt{\varepsilon_1(1-\varepsilon_1)} = 2\sqrt{\varepsilon_1(1-\varepsilon_1)}$$

To see the probability that classifier $h_1$ will be selected again, we have to find the it's error $\varepsilon_2$:

$$\varepsilon_2 = Pr_{i \sim D^{(2)}}[h_2(i) \neq y_i] = \sum_{i=1}^{m} D^{(2)}(i) \cdot 1_{[h_2(i) \neq y_i]}$$

If $x_i$ is misclassified then $h_1(x_i) \neq y_i$ so $D^{(2)}(i) = \frac{D^{(1)} \cdot e^{w_1}}{Z_2}$
For $h_1$ to be selected means: $Pr_{i \sim D^{(2)}}[h_2(i) \neq y_i] = Pr_{i \sim D^{(2)}}[h_1(i) \neq y_i]$
So,

$$\varepsilon_2 = \frac{\sum_{i=1}^{m} D^{(1)}(i) \cdot 1_{[h_1(i) \neq y_i]} \cdot e^{w_1}}{Z_2} = \frac{e^{w_1}}{Z_2} \cdot \sum_{i=1}^{m} D^{(1)}(i) \cdot 1_{[h_1(i) \neq y_i]}$$

$$\varepsilon_2 = \frac{\sqrt{\frac{1-\varepsilon_1}{\varepsilon_1}}}{2\sqrt{\varepsilon_1(1-\varepsilon_1)}} \cdot \varepsilon_1 = \frac{\sqrt{1-\varepsilon_1}}{2\sqrt{1-\varepsilon_1}} = \frac{1}{2}$$

This is a contradiction with the definition of weak learnability, such that we must have $\varepsilon_2 \leq \frac{1}{2} - \gamma_2$, with $\gamma_2 > 0$, which is not the case because $\varepsilon_2 = \frac{1}{2}$. So, the probability that the classifier $h_1$ will be selected again in round 2 is 0.

*b*) If we would know that the training error of the final classifier $h_{final}$ is at most $\frac{1}{2} - \frac{1}{3}\gamma + 2\gamma^3$, then it would be very easy to prove that it is strictly smaller $\frac{1}{2} - \gamma$:

$$\frac{1}{2} - \frac{1}{3}\gamma + 2\gamma^3 < \frac{1}{2} - \gamma$$

$$-\frac{1}{2}\gamma + 2\gamma^3 < 0 \ (\gamma > 0 \textit{ so we can devide without changing the sign})$$

$$2\gamma^2 < \frac{1}{2}$$

$$\gamma < \sqrt{\frac{1}{4}}$$

$$\gamma < \frac{1}{2}$$

We know that for every round, the weak learner return a weak classifier for which the error $\epsilon < \frac{1}{2} - \gamma, \gamma > 0$. Because, $\epsilon > 0$ and we proved that $\gamma < \frac{1}{2}$, the relation holds.

# 4 Exercise 4

According to Lecture 11, an algorithm A is $\gamma$-weak-learner algorithm for a class $H^d_{2DNF}$ if there exists a function $m_H : (0,1) \to N$ such that:

- for every $\delta > 0$
- for every labeling function $f \in H^d_{2DNF}, f : X \to \{0,1\}$
- for every distribution $D$ over $X$

when we run the learning algorithm A on a training set, consisting of $m \geq m_{\mathcal{H}}(\delta)$ examples sampled i.i.d. from D and labeled by f, the algorithm A returns a hypothesis h such that, with probability at least $1 - \delta$ (over the choice of examples), $L_{D,f}(h) \leq \frac{1}{2} - \gamma$.

We can use the transformation in Lecture 11:

$$A_1 \vee A_2 = \bigwedge_{u \in A_1, v \in A_2} (u \vee v) = \bigwedge_{u \in A_1, v \in A_2} y_{u,v}, \ \textit{each } u, v \textit{ can take } 2d \textit{ values in } \{x_1, \overline{x_1}, ..., x_d, \overline{x_d}\}$$

So we have that a 2-term DNF can be viewed as a conjunction of $(2d)^2$ variables: $H^d_{2DNF} \subseteq H^{(2d)^2}_{conj}$, which can efficiently PAC-learn the new class of conjunctions.

We can observe the fact that our disjunction will almost always be 1, except the case in which $u$ and $v$ are both 0, in which case we will have that $y_{u,v} = 0$. We can define an algorithm that labels only with 1, and when our $h(x) = 0$ is when $u$ and $v$ are 0. So, it has an error of $\frac{1}{4}$.

As said in the beginning, we want the loss to be $L_{D,f}(h) \leq \frac{1}{2} - \gamma$. We know that our algorithm has a loss of $\frac{1}{4}$ + some error $\epsilon$. We can take $\epsilon$ such that $\frac{1}{4} + \epsilon < \frac{1}{2} \Rightarrow \epsilon < \frac{1}{4}$.

If we make $\epsilon = \gamma$, we can say that we found an algorithm A that on a training set consisting of $m \geq m_{\mathcal{H}}(\delta)$ examples sampled i.i.d. from D and labeled by f, the algorithm A returns a hypothesis h such that, with probability at least $1-\delta$ (over the choice of examples), $L_{D,f}(h) \leq \frac{1}{2} - \gamma$, with $\gamma = \epsilon < \frac{1}{4}$.