Exercise 1 (Sample Size Bound for Decision Tree Learning) 3+2=5 points

We consider a Boolean classification problem with $a = \mathcal{A}$ features, where each feature can attain at most v possible values. Our hypothesis class $\mathcal{H}$ is the class of Boolean functions represented by decision trees over this feature set.

a) Define a suitable description scheme Δ for the hypothesis class, that is, for decision trees with this feature set.

Suppose $h \in \mathcal{H}$ is a hypothesis that can be represented by a decision tree with n nodes. In terms of n, a and v, give an upper bound for $|h|_\Delta$.

Hint: Your bound needs not be as sharp as possible. You are allowed to make slightly rougher estimates in order to obtain a nicer expression.

b) Consider again the Boolean classification problem from chapter 1 of the lecture whose objective was to decide under which circumstances students attend lectures (slide 1.32). The input features, and their possible values are as follows:

Feature	Possible Values
weather	sunny, cloudy, rainy, snowy
day of week	Monday, Tuesday, Wednesday, Thursday, Friday
party the night before	yes, no
topic	DB, ML, Alg, Logic
lecturer	Codd, Karp, Rabin, Valiant

Assume that the unknown target function can be represented by a decision tree with 10 nodes. Use your description scheme from the first part of the exercise, and give a lower bound m on the number of training examples such that any hypothesis that is consistent with m randomly drawn examples has a generalisation error of at most 5 % with probability greater than 99 %. Justify your answer.

Solution:

a) We use the alphabet $\Sigma = \{0, 1, \$\}$.

Node labels are either features or 0/1, so every node label can be represented as a binary number with $\lceil \log(a + 2) \rceil \leq \lceil \log(a) \rceil + 1$ bits.

Every attribute (per feature) can be represented as a binary number with $\lceil \log v \rceil$ bits.

Suppose the nodes of the decision tree are $\{1, \dots, n\}$, and let A_i be the label of node i.

We represent the nodes and their features with the string

$$w = \text{bin}(A_1)\$\,\text{bin}(A_2)\$ \cdots \$\,\text{bin}(A_n)\$$$

Then

$$\begin{aligned} |w| &\leq (n-1)\cdot(\lceil\log(a)\rceil+1)+n \\ &\leq n\lceil\log(a)\rceil+2n. \end{aligned}$$

The edges from node i to its children $j_1, \dots, j_k$ can be represented by a string

$$w_i = \mathrm{bin}(j_1)\$\,\mathrm{bin}(V_{ij_1})\$\cdots\$\,\mathrm{bin}(j_k)\$\,\mathrm{bin}(V_{ij_k})\$$$

where V_{ij} is the attribute value the edge is labelled with. Then

$$|w_1\cdots w_n| \leq (n-1)\cdot\lceil\log(n)\rceil+(n-1)\lceil\log(v)\rceil+2n-1 \quad \leq n\lceil\log(n)\rceil+n\lceil\log(v)\rceil+2n.$$

Let T be a shortest description of $h \in \mathcal{H}$. The full tree T is described by

$$w_T = w\$w_1\$\cdots\$w_n.$$

(Note that the individual parts are always separated by two consecutive $s.)

Then

$$\begin{aligned} |h|_\Delta = |w_T| \leq n\lceil\log(a)\rceil+2n+ \\ n\lceil\log(n)\rceil+n\lceil\log(v)\rceil+2n+ \\ n \\ \leq 5n+n\big(\lceil\log(a)\rceil+\lceil\log(n)\rceil+\lceil\log(v)\rceil\big) \end{aligned}$$

b) Using the bound from the first part of the exercise, and plugging in $n = 10$, $a = 5$ and $v = 6$, we get

$$|h|_\Delta \leq 5\cdot 10+10(3+4+3)=150.$$

By Theorem 3.8 from the lecture, the desired guarantee holds if $m \geq \frac{1}{\varepsilon}\cdot\big(n\ln|\Sigma|+\ln\left(\frac{2}{\delta}\right)\big)$ where n is an upper bound on the description length. Plugging in $n = 150$, $\varepsilon = 0.05$ and $\delta = 1-0.99 = 0.01$, this yields

$$m \geq \tfrac{1}{0.05}\cdot\big(150\cdot\ln 3+\ln\tfrac{2}{0.01}\big) = 20\big(150\ln 3+\ln 200\big) \approx 3401.80.$$

Thus, 3402 training examples are sufficient.

Exercise 2 (VC Dimension) 4 points

Determine the VC dimension $\mathrm{VC}(\mathcal{H})$ of the following class of functions, and prove that your claim is correct.
Let $\mathbb{X} = \mathbb{R}^2$ and let $\mathcal{H}$ be the class of all functions $h_{a,b} \colon \mathbb{R}^2 \to \{0,1\}$ with

$$h_{a,b}(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 = a \text{ or } x_2 = b \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

where $a, b \in \mathbb{R}$.

Hint: It may be more convenient to view $\{0,1\}$-valued functions h over $\mathbb{X}$ as subsets S_h of $\mathbb{X}$, where $S_h = \{x \in \mathbb{X} \mid h(x) = 1\}$. Then a set $Y \subseteq \mathbb{X}$ is shattered by $\mathcal{H}$ if (and only if) for every subset Y' of Y there exists $h \in \mathcal{H}$ such that $Y' = S_h \cap Y$.

Solution:

It holds that $\mathrm{VC}(\mathcal{H}) = 3$.

- We claim $\mathrm{VC}(\mathcal{H}) \geq 3$. Consider the set of points $Y = \left\{\binom{1}{1}, \binom{1}{2}, \binom{2}{3}\right\}$. Then Y is shattered by $\mathcal{H}$: (We represent functions $g \colon Y \to \{0,1\}$ as triples $\left(g(\binom{1}{1}), g(\binom{1}{2}), g(\binom{2}{3})\right)$.)

function	restriction of
$(0,0,0)$	$h_{0,0}$
$(0,0,1)$	$h_{0,3}$
$(0,1,0)$	$h_{0,2}$
$(1,0,0)$	$h_{0,1}$

function	restriction of
$(0,1,1)$	$h_{2,2}$
$(1,1,0)$	$h_{1,1}$
$(1,0,1)$	$h_{1,2}$
$(1,1,1)$	$h_{2,3}$

- We claim $\mathrm{VC}(\mathcal{H}) < 4$. Assume otherwise and let Y be any set of four distinct points in $\mathbb{R}^2$ that is shattered by $\mathcal{H}$. Then the 4 points must lie on cross ($Y = S_{h_{a,b}}$ for some a and b). If 3 of them lie on the same vertical or horizontal line (either $x = a$ or $y = b$), then there is no function $h_{a',b'} \in \mathcal{H}$ that is equal to 0 on exactly one of these three points, and 1 on all other points. The only other possibility is that exactly two of the points lie on $x = a$, and the other two lie on $y = b$. But then there exists no function $h_{a',b'}$ in $\mathcal{H}$ that is equal to 1 on exactly 3 of the 4 points. Thus, Y is not shattered by $\mathcal{H}$ in contradiction to the assumption.

Exercise 3 (Bandit Learning) 4+1=5 points

Consider the bandit learning scenario with three slot machines (or actions) $a \in \{1,2,3\}$. We want to execute the Exp3 algorithm over 3 rounds using the parameter $\gamma = \frac{1}{2}$.

a) In order to reenact the execution algorithm, for this exercise we assume that the sequence of actions that gets drawn is

$$\mathbf{a} = (a^{(1)}, a^{(2)}, a^{(3)}) = (1,2,3)$$

(regardless of the probability distribution). Also assume that the rewards of these actions in the respective rounds are

$$q_1^{(1)} = 3\ln(2) \quad q_2^{(2)} = 5\ln(2) \quad q_3^{(3)} = 3\ln(2).$$

Give the weight vectors $(w_1^{(s)}, w_2^{(s)}, w_3^{(s)})$ and the probabilities $(p_1^{(s)}, p_2^{(s)}, p_3^{(s)})$ for $s = 1,2,3,4$ as computed by the Exp3 algorithm. Give numeric results for $s = 4$ that are rounded off to 2 decimal places.

b) In the given action sequence, both action 1 and 3 yielded the same reward when they were chosen. Yet, $w_1^{(4)}$ and $w_3^{(4)}$ are different. Discuss why.

Solution:

a)

$$\begin{array}{cccc}
\mathbf{w}^{(1)} & \mathbf{w}^{(2)} & \mathbf{w}^{(3)} & \mathbf{w}^{(4)} \\
\hline
\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 2.82842712 \\ 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 2.82842712 \\ 8.47907147 \\ 1 \end{pmatrix} & \begin{pmatrix} 2.82842712 \\ 8.47907147 \\ 5.32231117 \end{pmatrix} \approx \begin{pmatrix} 2.83 \\ 8.48 \\ 5.32 \end{pmatrix}
\end{array}$$

$$\begin{array}{cccc}
\mathbf{p}^{(1)} & \mathbf{p}^{(2)} & \mathbf{p}^{(3)} & \mathbf{p}^{(4)} \\
\hline
\begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 0.33333333 \\ 0.33333333 \\ 0.33333333 \end{pmatrix} & \begin{pmatrix} 0.45955989 \\ 0.27022006 \\ 0.27022006 \end{pmatrix} & \begin{pmatrix} 0.28157333 \\ 0.51113437 \\ 0.20729231 \end{pmatrix} & \begin{pmatrix} 0.25170754 \\ 0.42160258 \\ 0.32668988 \end{pmatrix} \approx \begin{pmatrix} 0.25 \\ 0.42 \\ 0.33 \end{pmatrix}
\end{array}$$

b) The weight of action 3 is higher: at the point where it is chosen, its probability is lower than that of action 1 (because the weight of action 1 has been increased in the first round); thus, its reward has a greater impact on the new weight (compared to the first round of the algorithm).

Exercise 4 (MWU with Payoffs) 6 points

We consider the multiplicative weight update algorithm. In some situations it is easier to model the problem using *payoffs* (also called *gains* or *rewards*) instead of costs. For this we use a *payoff matrix* ($n \times t$, with entries $r_i^{(s)} \in [0,1]$) and the weight update rule

$$w_i^{(t+1)} := w_i^{(t)} \left(1 + \alpha \cdot r_i^{(t)}\right)$$

where $r_i^{(t)}$ is the *reward* of following expert i in round t. The choice which expert to follow is done randomly according to

$$p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}}.$$

Show that, if one extends the MWU algorithm as described above, the following bound on the *expected payoff in round* t holds:

$$\sum_{s=1}^{t} \sum_{i=1}^{n} r_i^{(s)} p_i^{(s)} \geq -\frac{\ln n}{\alpha} + (1-\alpha) \sum_{s=1}^{t} r_j^{(s)}$$

for all $t \geq 1$ and all $j \in [n]$.

Hint: You can follow the general idea of the proof of Theorem 4.2 in the lecture. The following inequalities may be useful for achieving the desired bound (you may use them without showing them to hold):

$$1 + \alpha x \geq (1+\alpha)^x \qquad \text{for all } x \in [0,1] \text{ and } \alpha > -1. \tag{1}$$

$$\ln(1+\alpha) \geq \alpha - \alpha^2 \qquad \text{for all } \alpha \geq 0. \tag{2}$$

$$1 + x \leq e^x \qquad \text{for all } x \in \mathbb{R}. \tag{3}$$

Solution: ______________________________

For all $i \in [n]$ we have $w_i^{(1)} = 1$ and

$$w_i^{(t+1)} = \prod_{s=1}^{t} \left(1 + \alpha r_i^{(s)}\right) \overset{(1)}{\geq} \prod_{s=1}^{t} (1+\alpha)^{r_i^{(s)}} = (1+\alpha)^{\sum_{s=1}^t r_i^{(s)}}. \tag{4}$$

Define the potential function

$$\Phi^{(t)} := \sum_{i=1}^{n} w_i^{(t)}.$$

Using (4) it follows that for all $i \in [n]$,

$$\Phi^{(t+1)} \geq \sum_{i=1}^{n} (1+\alpha)^{\sum_{s=1}^t r_i^{(s)}} \geq (1+\alpha)^{\sum_{s=1}^t r_i^{(s)}}. \tag{5}$$

On the other hand

$$\Phi^{(t+1)} = \sum_{i=1}^{n} w_i^{(t+1)} = \sum_{i=1}^{n} w_i^{(t)}\left(1 + \alpha r_i^{(t)}\right) = \sum_{i=1}^{n} w_i^{(t)} + \alpha \cdot \sum_{i=1}^{n} w_i^{(t)} r_i^{(t)}.$$

Now define

$$r^{(t)} := \sum_{i=1}^{n} r_i^{(t)} p_i^{(t)} = \frac{1}{\Phi^{(t)}} \sum_{i=1}^{n} r_i^{(t)} w_i^{(t)}$$

to be the *expected payoff* in round t. Then we have

$$\Phi^{(t+1)} = \Phi^{(t)} + \alpha \cdot \Phi^{(t)} r^{(t)} = \Phi^{(t)}\left(1 + \alpha r^{(t)}\right).$$

By induction on t (note that $\Phi^{(1)} = n$),

$$\Phi^{(t+1)} = n \cdot \prod_{s=1}^{t} \left(1 + \alpha r^{(s)}\right) \overset{(3)}{\leq} n \cdot \prod_{s=1}^{t} e^{\alpha r^{(s)}} = n \cdot \exp\left(\alpha \textstyle\sum_{s=1}^{t} r^{(s)}\right). \tag{6}$$

From (5) and (6) we get

$$(1+\alpha)^{\sum_{s=1}^{t} r_i^{(s)}} \leq n \cdot \exp(\alpha \textstyle\sum_{s=1}^{t} r^{(s)}).$$

Taking logarithms on both sides then yields

$$\begin{aligned} &\ln(1+\alpha) \cdot \sum_{s=1}^{t} r_i^{(s)} \leq \ln n + \alpha \sum_{s=1}^{t} r^{(s)} \\ \overset{(2)}{\Leftrightarrow}\ &(\alpha - \alpha^2) \cdot \sum_{i=1}^{t} r_i^{(s)} \leq \ln n + \alpha \sum_{i=1}^{t} r^{(s)} \\ \Leftrightarrow\ &-\frac{\ln n}{\alpha} + (1-\alpha) \sum_{s=1}^{t} r_i^{(s)} \leq \sum_{s=1}^{t} r^{(s)}. \end{aligned}$$