Machine Learning Foundations Homework 1

Problem 1 (60 points)**.**

Figure 1: Q1 Screenshot (100 points). Please zoom in to see the details.

Problem 2 (20 points + 10 bonus points)**.** Describe an application of semi-supervised learning in 10 sentences.

Sol. Scoliosis is a medical condition in which the human spine has an abnormal bent, curve, or is otherwise positioned at an unusual angle. There is no known cause for this, though it is most commonly found in females and children going through puberty. While mild scoliosis is generally benign apart from posture abnormalities, more severe forms can cause lung and heart damage, along with chronic back problems. To this end, prognosis is generally determined through X-rays, CT scans or MRI, though identifying the severity of the bent is done by eye and thus is time-consuming (especially if other medical problems are to be identified as well). There is thus an abundance of X-ray scans, but a shortage of well-labelled scans; this shortage is compounded by the fact that X-ray scans provided by different hospitals give varying amounts of personal information on their patients, causing data to become disorganized and hard to generalise. Using semi-supervised learning and computer vision, it is possible to collect large amounts of unlabelled X-ray scans and some labelled X-ray scans and categorize the image by their risk/severity of scoliosis: for example, images may be determined to have "mild", "severe", or "no" symptoms, or group them by course of action (for example, the patient requires "no action", "spinal braces", or "surgery".) □

Problem 3 (20 points)**.**

Sol. Denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ by $\mathcal{Y}^{\mathcal{X}}$.

Let $\mathcal{U}$ be the uniform distribution on

$$\mathcal{H} = \left\{ f \in \mathcal{Y}^{\mathcal{X}} \;\middle|\; f(x_n) = y_n \text{ for all } n = 1, \ldots, N \right\}.$$

We must determine whether or not

$$\mathop{\mathbb{E}}_{f \sim \mathcal{U}} \left[\mathrm{E}_{\mathrm{OTS}}(\mathcal{A}(\mathcal{D}), f) \right]$$

is constant regardless of the algorithm $\mathcal{A} : \mathscr{P}(\mathcal{X} \times \mathcal{Y}) \mapsto \mathcal{Y}^{\mathcal{X}}$, where $\mathscr{P}$ is the powerset. To this end, let $g \in \mathcal{Y}^{\mathcal{X}}$ be the hypothesis that $\mathcal{A}$ outputs.

We may also see that since $f(x_{N+\ell}), g(x_{N+\ell}) \in \{-1, 1\}$, it follows that

$$f^2(x_{N+\ell}) = g^2(x_{N+\ell}) = 1$$

and

$$[\![f(x_{N+\ell}) \neq g(x_{N+\ell})]\!] = \left(\frac{f(x_{N+\ell}) - g(x_{N+\ell})}{2} \right)^2.$$

Then,

$$\mathop{\mathbb{E}}_{f \sim \mathcal{U}} \left[\mathrm{E}_{\mathrm{OTS}}(g, f) \right] = \frac{1}{|\mathcal{H}|} \sum_{f \in \mathcal{H}} \mathrm{E}_{\mathrm{OTS}}(g, f) = \frac{1}{L\,|\mathcal{H}|} \sum_{f \in \mathcal{H}} \sum_{\ell=1}^{L} [\![g(x_{N+\ell}) \neq f(x_{N+\ell})]\!]$$

$$= \frac{1}{L\,|\mathcal{H}|} \sum_{f \in \mathcal{H}} \sum_{\ell=1}^{L} \left(\frac{g(x_{N+\ell}) - f(x_{N+\ell})}{2} \right)^2$$

$$= \frac{1}{4L\,|\mathcal{H}|} \sum_{f \in \mathcal{H}} \sum_{\ell=1}^{L} \left(g^2(x_{N+\ell}) - 2g(x_{N+\ell}) f(x_{N+\ell}) + f^2(x_{N+\ell}) \right)$$

$$= \frac{1}{4L\,|\mathcal{H}|} \sum_{f \in \mathcal{H}} \sum_{\ell=1}^{L} (1 - 2g(x_{N+\ell}) f(x_{N+\ell}) + 1)$$

$$= \frac{1}{2L\,|\mathcal{H}|} \left(L\,|\mathcal{H}| - \sum_{\ell=1}^{L} g(x_{N+\ell}) \sum_{f \in \mathcal{H}} f(x_{N+\ell}) \right)$$

At this point, we observe that applying any function $f \in \mathcal{H}$ on any test input $x_{N+\ell}$ will yield either -1 or $+1$.

If we exhaustively apply all possible functions on every single test input, we obtain a matrix A, where $a_{ij} = f_i(x_j)$ is the evaluation of the ith function $f_i \in \mathcal{H}$ onto the jth test input x_j. we can see that for any particular row a_i, there will be an equal amount of $+1$s and -1s.

To illustrate why this is true, observe that if we fix the results of every function f on every test input x *except* for one test input x', we can see that

for each combination of f and x, there is a combination where $f(x') = +1$ and $f(x') = -1$.

Thus, taking the sum of each row in A will yield 0. In other words, if we sum the result of applying every function f onto a particular test input $x_{N+\ell}$, we have

$$\sum_{f \in \mathcal{H}} f(x_{N+\ell}) = 0.$$

Returning to our original proof, we can obtain

$$\begin{aligned}
&\frac{1}{2L\,|\mathcal{H}|}\left(L\,|\mathcal{H}| - \sum_{\ell=1}^{L} g(x_{N+\ell}) \sum_{f \in \mathcal{H}} f(x_{N+\ell})\right) \\
&= \frac{1}{2} - \sum_{\ell=1}^{L} \frac{g(x_{N+\ell})}{2L\,|\mathcal{H}|} \cdot 0 \\
&= \frac{1}{2}.
\end{aligned}$$

Since any such algorithm outputs a $g \in \mathcal{Y}^{\mathcal{X}}$, the expectation is constant. $\square$

	1	2	3	4	5	6
A						
B						
C						
D						

Table 1: Dice Specification

For simplicity in problems 4 and 5, each digit of each type of die is colored in either orange or green as specified in the table.

Problem 4 (20 points)**.**

Sol. The probability for five dice picked to have digit 1 all colored in green is as follows:

All five dice must be either of type A or type D. Since there are 4 possible dice types, the probability is $(2/4)^5 = 1/32$. □

Problem 5 (20 points)**.**

Sol. We find the probability that some number is purely green below.

First, if any of the following events happen, it is impossible that any number will be purely green:

- At least one dice is of Type A, and at least one dice is of Type B.
- At least one dice is of Type C, and at least one dice is of Type D.

From the two conditions above, we can also see that if at least 3 types of dice appear, it is also impossible that any number will be purely green.

Thus, for a number to be purely green, one of the following must occur:

- All the dice are of a single type.
- Only two types of dice appear, and those two types cannot be A and B, or C and D.

We can define the following probabilities:

- $\mathbb{P}$(all 1 type) as the chance that all dice are of a single type.
- $\mathbb{P}$(exactly A) as the chance that all dice are of type A. $\mathbb{P}$(exactly B) is defined similarly.
- $\mathbb{P}$(exactly 2 types) as the chance that exactly two types of dice appear.
- $\mathbb{P}$(exactly A and B) as the chance that exactly two types of dice appear, those types being A and B. $\mathbb{P}$(exactly C and D) is defined similarly.
- $\mathbb{P}$(exactly A or B) as the chance that all dice are either of type A or type B.

Due to symmetry, the probability is

$$\begin{aligned}
&\mathbb{P}(\text{all 1 type}) + \mathbb{P}(\text{exactly 2 types}) - \mathbb{P}(\text{exactly A and B}) - \mathbb{P}(\text{exactly C and D}) \\
&= 6\mathbb{P}(\text{exactly A and B}) + 4\mathbb{P}(\text{exactly A}) - 2\mathbb{P}(\text{exactly A and B}) \\
&= 4(\mathbb{P}(\text{exactly A or B}) - \mathbb{P}(\text{exactly A}) - \mathbb{P}(\text{exactly B}) + \mathbb{P}(\text{exactly A})) \\
&= 4(\mathbb{P}(\text{exactly A or B}) - \mathbb{P}(\text{exactly A})) = 4(2^5/4^5 - 1/4^5) = 31/256.
\end{aligned}$$

We can then see that the probability for "1" to be purely green (1/32) is much lower than the probability for some colour to be green (31/256). Intuitively, this makes sense, as a dice has more than just one digit.

□

Problem 6 (20 points)**.**

Sol. The environment is as follows:

- The program name is `q6.py`.
- The random seeds are: $5, 6, \ldots, 1130$, with 1126 seeds in total.
- The plotting is done in matplotlib.

The average number of updates before the algoritm halts is 39.735.

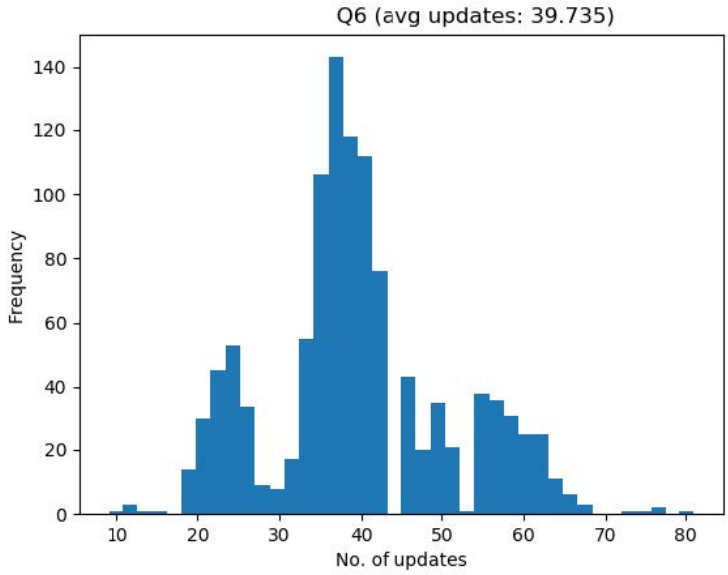

Figure 2: Average number of updates using PLA.

□

Problem 7 (20 points)**.** Coding.

Sol. The environment is as follows:

- The program name is `q7.py`, and the plotting program `plot_q7.py`.
- The random seeds are: $5, 6, \ldots, 1130$, with 1126 seeds in total.
- The plotting is done in matplotlib.

The average error rate is 0.1226.

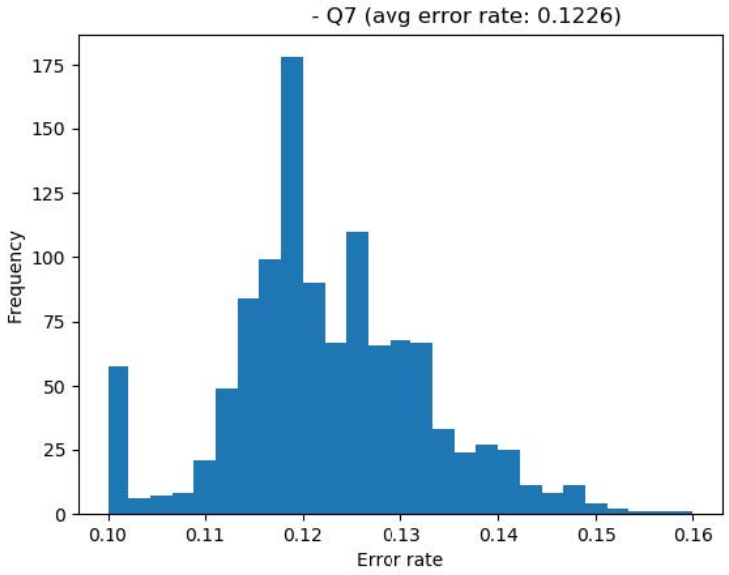

Figure 3: Average error rate using w_{POCKET}.

□

Problem 8 (20 points)**.**

Sol. The environment is as follows:

- The program name is `q8.py`.
- The random seeds are: $5, 6, \ldots, 1130$, with 1126 seeds in total.
- The plotting is done in matplotlib.

The average error rate is 0.33613.

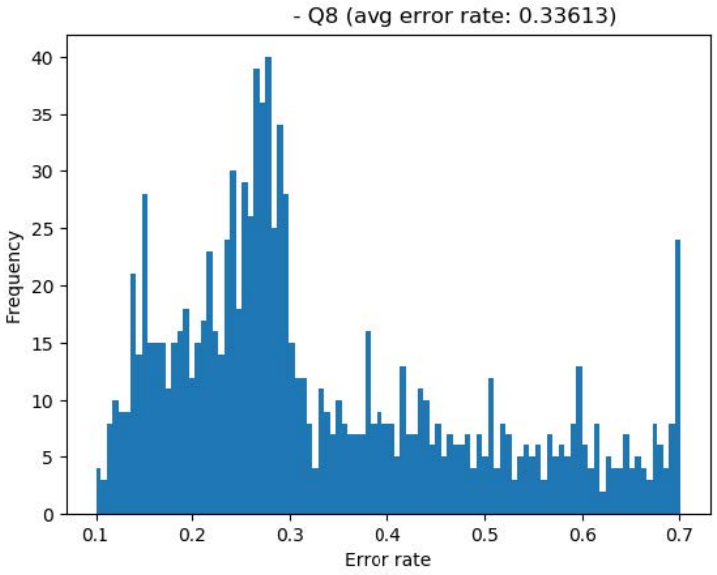

Figure 4: Average error rate using w_{100}.

Intuitively, the error rate of w_{100} should be higher than w_{POCKET}'s, because the given training set has 500 items, so realistically w_{100} will not go through as many iterations as the pocket algorithm's (which runs through the training data 100 times, whereas it is possible that w_{100} does not even finish the first iteration). In addition, the pocket algorithm seems to have a more rigorous updating method, as it guarantees that the returned vector will make the least mistakes out of all its iterations, whereas w_{100} makes no such guarantee.

□

Problem 9 (10 points)**.**

Sol. No, speedup of any kind cannot be guaranteed, as

$$T \leq \frac{R^2}{\rho^2} = \frac{\max_n \|x_n\|^2}{\left(\min_n y_n x_n \cdot (w_f/\|w_f\|^2)\right)^2} = \frac{\max_n \|x_n/10\|^2}{\left(\min_n y_n (x_n/10) \cdot (w_f/\|w_f\|^2)\right)^2}.$$

In other words, the upper bounds of T for both cases (normal and with shrinking) are exactly the same, not reduced by tenfold.

□