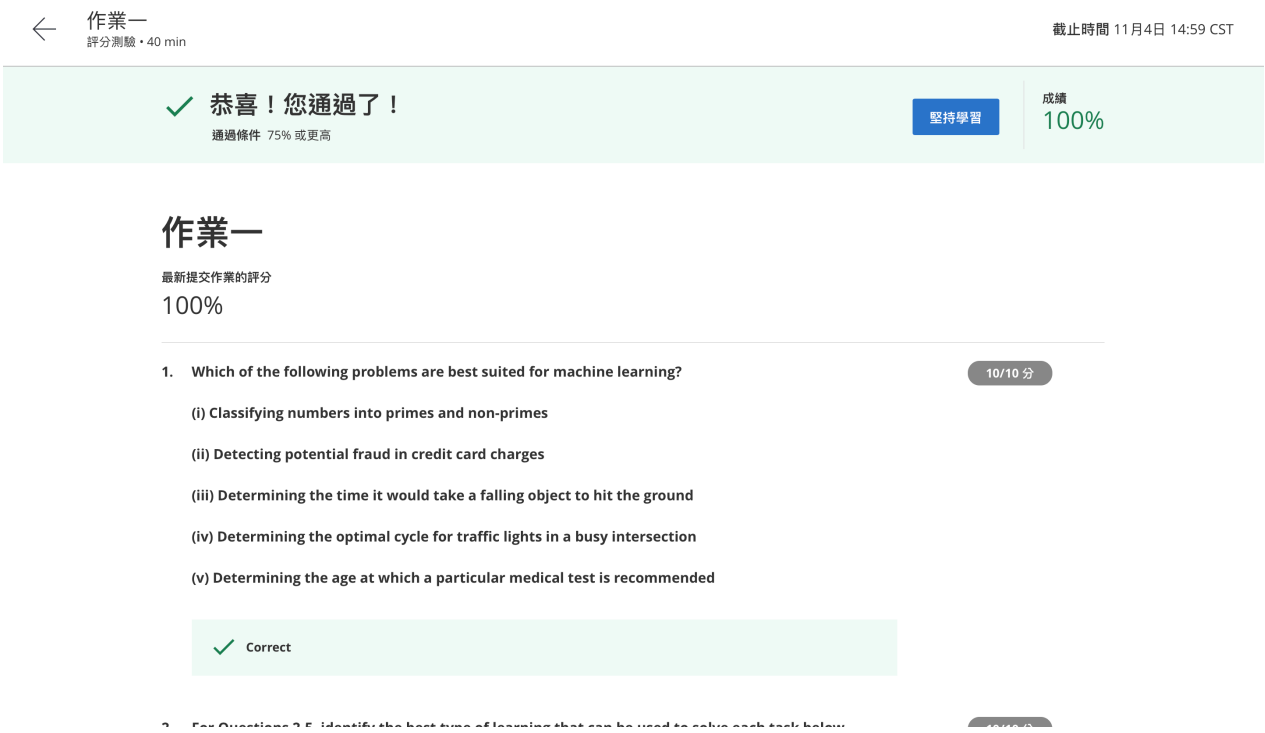
Machine Learning Foundations HW1

B06705028 資管三 朱紹瑜

1.



2.

Consider the subtitles in Youtube. For some of the video clips, users upload the videos along with transcripts, while the subtitles of others are auto-generated. Such a speech recognition feature is an application of semi-supervised learning.

3.

Consider all those f that can "generate" $\mathcal{D}$ in a noiseless setting. Their outputs on $\{\mathbf{x}_n\}_{n=1}^N$ are fixed, while those on $\{\mathbf{x}_{N+l}\}_{l=1}^L$ are arbitrary (-1 or $+1$ for every of the L inputs). Thus, there are 2^L possible f. Each possible f has a probability of $\frac{1}{2^L}$ since all of them are equal likely in probability as stated in the problem description.

The expected value of off-training-set error for any deterministic algoritm $\mathcal{A}$ is as the following.

$$
\begin{aligned}
& \mathbb{E}_f\{E_{OTS}(\mathcal{A}(\mathcal{D}), f)\} \\
= & \sum_{i=1}^{2^L}\left(Pr(f=f_i)\cdot\frac{1}{L}\sum_{\mathrm{l}=1}^{L}[\mathcal{A}(\mathcal{D})(\mathbf{x}_{N+1})\neq f_i(\mathbf{x}_{N+1})]\right) \\
= & \sum_{i=1}^{2^L}\left(\frac{1}{2^L}\cdot\frac{1}{L}\sum_{\mathrm{l}=1}^{L}[\mathcal{A}(\mathcal{D})(\mathbf{x}_{N+1})\neq f_i(\mathbf{x}_{N+1})]\right) \\
= & \frac{1}{2^L}\cdot\frac{1}{L}\sum_{i=1}^{2^L}\sum_{\mathrm{l}=1}^{L}[\mathcal{A}(\mathcal{D})(\mathbf{x}_{N+1})\neq f_i(\mathbf{x}_{N+1})] \\
= & \frac{1}{2^L}\cdot\frac{1}{L}\left(0\cdot C_0^L+1\cdot C_1^L+2\cdot C_2^L+\ldots+(L-1)C_{L-1}^L+L\cdot C_L^L\right) \\
= & \frac{1}{2^L}\cdot\frac{1}{L}\sum_{j=0}^{L} j\cdot C_j^L \\
= & constant
\end{aligned}
$$

Given outputs of $\mathcal{A}(\mathcal{D})$ on the test inputs, it is quaranteed that we can find C_j^L functions in the set of possible f that performs differently from $\mathcal{A}(\mathcal{D})$ on exactly j $\mathbf{x}_n$, where j are integers from 0 to L. Moreover, $\sum_{j=0}^{L} C_j^L = 2^L$, which indicates that by grouping possible f by considering its "distance" to $\mathcal{A}(\mathcal{D})$, all possible f are collected. Thus, given equation (3), equation (4) must hold.

From the equations above, we can see that $\mathbb{E}_f\{E_{OTS}(\mathcal{A}(\mathcal{D}), f)\}$ is a constant regardless of $\mathcal{A}$.

4.

To get all green 1's, only dices of type A and type D are picked. Thus, the probability is $(\frac{1}{2})^5 = \frac{1}{32}$.

5.

Having "some number" to be purely green can be partitioned into 8 subcases.

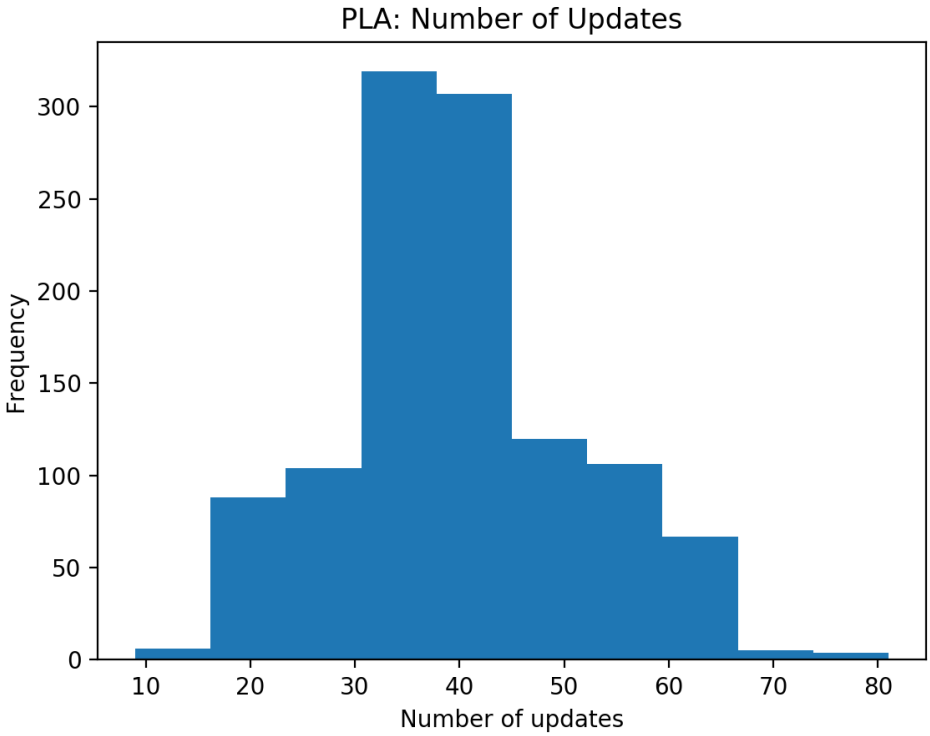
- all of type A
- all of type B
- all of type C
- all of type D
- a mixture of type A and type C
- a mixture of type A and type D
- a mixture of type B and type C
- a mixture of type B and type D

For the first 4 subcases, the probability of each of them is $(\frac{1}{4})^5 = \frac{1}{1024}$. For the last 4 subcases, the probability of each of them is $(C_1^5 + C_2^5 + C_3^5 + C_4^5)(\frac{1}{4})^5 = \frac{30}{1024}$. The total probability of having "some number" to be purely green is $4 \cdot \frac{1}{1024} + 4 \cdot \frac{30}{1024} = \frac{31}{256}$.

The probability of having "some number" purely green is higher than that of having all green 1's, but lower than six times of such probability.

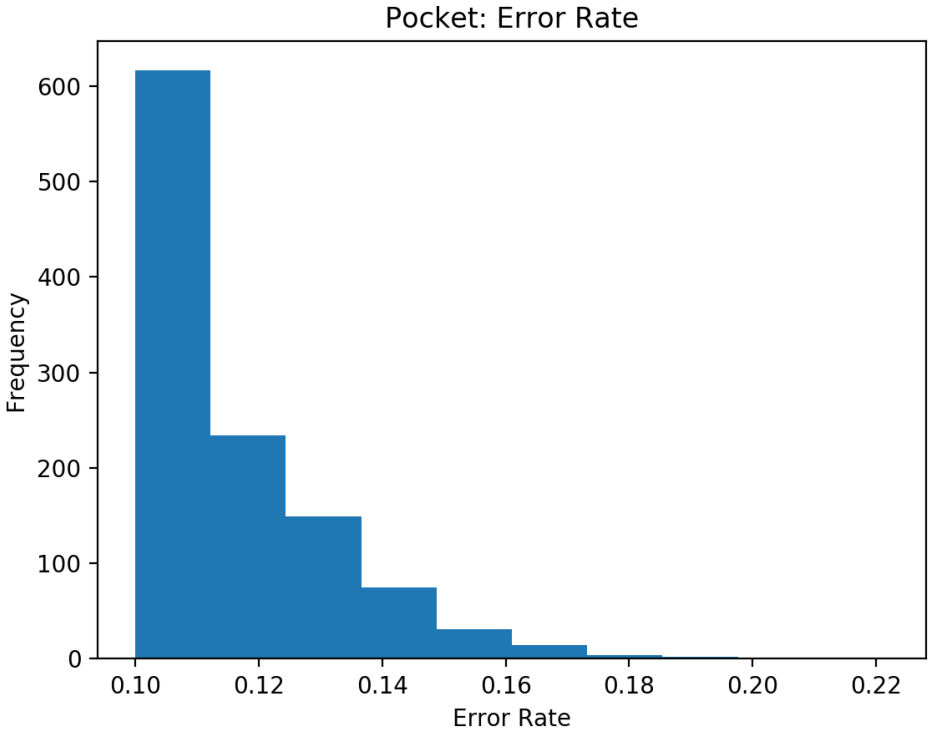
6.

average number of updates before the algorithm halts: 39.8



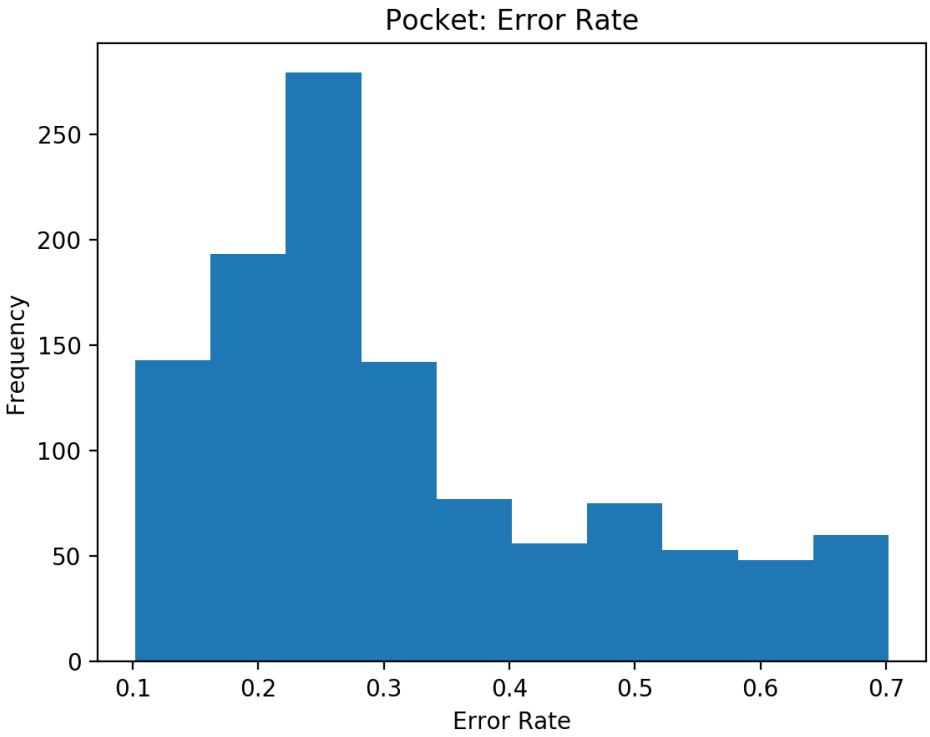
7.

Repeating 1126 times: average $E_{OTS} = 0.116$



8.

Repeating 1126 times: average $E_{OTS} = 0.319$



Comparing the off-training-set errors in problem 7 and 8, we can infer that the average error and the variation of errors are much higher when we return $\mathbf{w}_{100}$ instead of $\mathbf{w}_{pocket}$. With the pocket algorithm being greedy, the performance of $\mathbf{w}_{100}$ is guranteed to be no better than $\mathbf{w}_{pocket}$ on the training set. Along with the assumption that the testing data and the training data are generated from the same target function, we can explain the result of the comparison.

9.

No, the PLA algorithm will not run faster after scaling down all $\mathbf{x}_n$ linearly.

Let the following variables be defined. $T =$ number of mistake corrections before PLA halts
$R = \sqrt{\max_n \|\mathbf{x}_n\|^2}$ $\rho = \min_n y_n \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|}\mathbf{x}_n$

First, we derive an upper bound of T.

$$
\begin{aligned}
\mathbf{w}_f^T \mathbf{w}_t &= \mathbf{w}_f^T \left(\mathbf{w}_{t-1} + y_{n,t-1} \mathbf{x}_{n,t-1} \right) \\
&= \mathbf{w}_f^T \mathbf{w}_{t-1} + y_{n,t-1} \mathbf{w}_f^T \mathbf{x}_{n,t-1} \\
&\geq \mathbf{w}_f^T \mathbf{w}_{t-1} + \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \\
&\geq \left(\mathbf{w}_f^T \mathbf{w}_{t-2} + \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \right) + \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \\
&\geq \mathbf{w}_f^T \mathbf{w}_{t-3} + 3 \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \\
&\geq \mathbf{w}_f^T \mathbf{w}_0 + T \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \\
&= T \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \\
\| \mathbf{w}_t \|^2 &= \| \mathbf{w}_{t-1} + y_{n,t-1} \mathbf{x}_{n,t-1} \|^2 \\
&= \| \mathbf{w}_{t-1} \| + 2 y_{n,t-1} \mathbf{w}_{t-1} \mathbf{x}_{n,t-1} + \| y_{n,t-1} \mathbf{x}_{n,t-1} \|^2 \\
&\leq \| \mathbf{w}_{t-1} \| + \| y_{n,t-1} \mathbf{x}_{n,t-1} \|^2 \\
&\leq \| \mathbf{w}_{t-1} \| + \max_n \| \mathbf{x}_n \|^2 \\
&\leq \left(\| \mathbf{w}_{t-2} \| + \max_n \| \mathbf{x}_n \|^2 \right) + \max_n \| \mathbf{x}_n \|^2 \\
&\leq \| \mathbf{w}_{t-3} \| + 3 \max_n \| \mathbf{x}_n \|^2 \\
&\leq \| \mathbf{w}_0 \| + T \max_n \| \mathbf{x}_n \|^2 \\
&= T \max_n \| \mathbf{x}_n \|^2 \\
1 \geq \frac{\mathbf{w}_f^T}{\| \mathbf{w}_f \|} \frac{\mathbf{w}_t}{\| \mathbf{w}_t \|} &\geq \frac{T \min_n y_n \mathbf{w}_f^T \mathbf{x}_n}{\| \mathbf{w}_f \| \sqrt{T \max_n \| \mathbf{x}_n \|^2}} \geq \frac{\rho \sqrt{T}}{R} \\
T &\leq \frac{R^2}{\rho^2}
\end{aligned}
$$

From the inequalities above, we get an upper bound of T which is $\frac{R^2}{\rho^2}$.

Let $\mathbf{x}'$, T', R' and ρ' denote the associated variables for the modifed $\mathbf{x}$'s, where

$$
\begin{aligned}
\mathbf{x}'_n &= \frac{1}{10} \mathbf{x}_n \\
R' &= \frac{1}{10} R \\
\rho' &= \frac{1}{10} \rho
\end{aligned}
$$

Then we can derive the upper bound of T'.

$$
T' \leq \frac{(R')^2}{(\rho')^2} = \frac{(\frac{1}{10} R)^2}{(\frac{1}{10} \rho)^2} = \frac{R^2}{\rho^2}
$$

From the inequalities above we can see that the upper bound of the number of mistake corrections stays the same after scaling all $\mathbf{x}_n$ down linearly by a factor of 10. Thus, we proved that Dr. Learn's plan is not going to work.