CM146, Fall 2019

Problem Set 4: Learning Theory, Boosting, Multi-class Classification

Due December 8, 2019, 11:59pm

Name: Yongqian Li

Student Id #: 004997466

Submission instructions

- Submit your solutions electronically on the course Gradescope site as PDF files.
- If you plan to typeset your solutions, please use the LaTeX solution template. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app.

1 PAC Learning [25 pts]

(a). Iterate all examples in Strain, for each data point (x_i, y_i) in the m examples

If its label $y_i = 1$, the hypothesis is h_{x_i}. ~~If none of the~~ If after all iterations no $y_i = 1$ is found, the hypothesis is h^-. The training error of this algorithm is always 0 since the prediction is consistent with ~~the~~ Strain.

(b). The hypothesis A(Strain) is always consistent with Strain, and it makes at most 1 mistake in D since any $h^* \in H_{Singleton}$ is either h^- or has only one $x \in X$ s.t. $h^*(x) = 1$, and mistakes are only made when A(Strain) is h^- with $h^* \neq h^-$ (Otherwise we would have predicted h^* correctly).

$\therefore$ Loss is either 0 or $\frac{1}{N}$.

If the true h^* is h_{x^*}, we will make mistakes only if x^* does not appear in Strain. $\therefore P(loss > 0) = (1 - \frac{1}{N})^m$

$\because P(loss > 0) \geq P(loss > \varepsilon)$ $\therefore$ If $\sigma \geq P(loss > 0)$, then $\sigma \geq P(loss > \varepsilon)$

When $loss > \varepsilon$, $loss = \frac{1}{N} \Rightarrow \frac{1}{N} > \varepsilon$ $\therefore (1 - \frac{1}{N})^m < (1 - \varepsilon)^m$

$\therefore$ let $\sigma \geq (1 - \varepsilon)^m > (1 - \frac{1}{N})^m \geq P(loss > \varepsilon)$,

$\sigma \geq (1 - \varepsilon)^m$ if $e^{-\varepsilon m} \leq \sigma$

$-\varepsilon m \leq \log \sigma$

$m \geq -\frac{\log \sigma}{\varepsilon} = \frac{\log \frac{1}{\sigma}}{\varepsilon}$

$\therefore$ If $m \geq \frac{\log \frac{1}{\sigma}}{\varepsilon}$,

$\sigma \geq (1-\varepsilon)^m > (1-\frac{1}{N})^m$

$\because$ Loss is either 0 or $\frac{1}{N}$ $\therefore P(\text{loss} > \varepsilon) = P(\text{loss} > 0) = (1-\frac{1}{N})^m$

$\therefore \sigma \geq (1-\varepsilon)^m > P(\text{loss} > \varepsilon)$

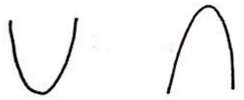
$\therefore$ when $m \geq \frac{\log \frac{1}{\sigma}}{\varepsilon}$, $\sigma \geq P(L_{(D,h^*)}(A(S_{train})) > \varepsilon)$

i	Label	D_0	$f_1 \equiv$ $[x > 2]$	$f_2 \equiv$ $[y > 6]$	$h_1 \equiv$ $[x>2]$	D_1	$f_1 \equiv$ $[x > 9]$	$f_2 \equiv$ $[y > 11]$	$h_2 \equiv$ $[y>11]$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	−	0.1	−	+	−	0.0625	−	−	−
2	−	0.1	−	−	−	0.0625	−	−	−
3	+	0.1	+	+	+	0.0625	−	−	−
4	−	0.1	−	−	−	0.0625	−	−	−
5	−	0.1	−	+	−	0.0625	−	+	+
6	−	0.1	+	+	+	0.25	−	−	−
7	+	0.1	+	+	+	0.0625	+	−	−
8	−	0.1	−	−	−	0.0625	−	−	−
9	+	0.1	−	+	−	0.25	−	+	+
10	+	0.1	+	+	+	0.0625	−	−	−

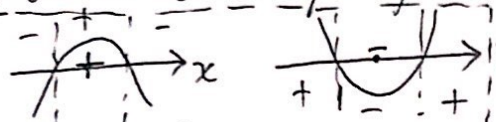
Table 1: Table for Boosting results

2 VC Dimension [15 pts]

VC(H) = 3. ax^2+bx+c has two directions of graphs on a Cartesian plane

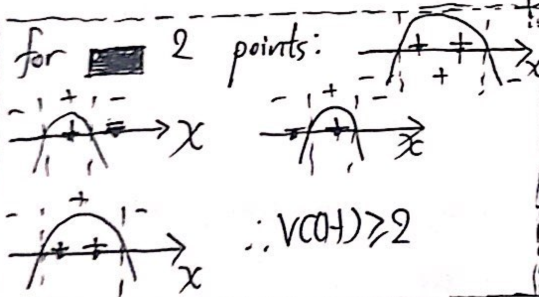


for any point:



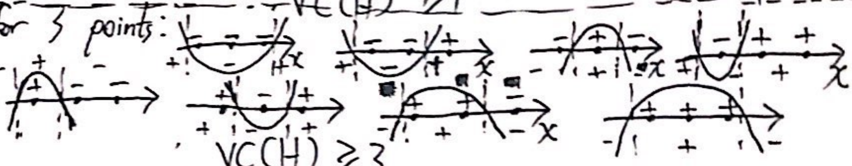
$\therefore VC(H) \geq 1$

for [redacted] 2 points:



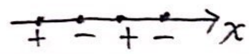
$\therefore VC(H) \geq 2$

for 3 points:



$\therefore VC(H) \geq 3$

for 4 points,



This assignment cannot be shattered by H.

$\therefore VC(H) = 3$

3 Boosting [40 pts]

i	x	y	Label
1	0	8	−
2	1	4	−
3	3	7	+
4	-2	1	−
5	-1	13	−
6	9	11	−
7	12	7	+
8	-7	-1	−
9	-3	12	+
10	5	9	+

(a). $D_0(i) = \frac{1}{10} = 0.1$

(b). $\varepsilon_{f_1} = \frac{2}{10} = 0.2$

$\varepsilon_{f_2} = \frac{3}{10} = 0.3 > \varepsilon_{f_1}$

$\therefore$ choose f_1

(c). $\varepsilon_1 = 0.2$

$\alpha_1 = \frac{1}{2}\log_2\left(\frac{1-0.2}{0.2}\right)$

$= \frac{1}{2}\log_2 4$

$= 1$

$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} 2^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ 2^{\alpha_t} & \text{else} \end{cases}$

$Z_1 = 2\sqrt{\varepsilon_1(1-\varepsilon_1)}$

$= 2\sqrt{0.2 \times 0.8} = 0.8$

∵ $D_0(i)=0.1$

~~∴ when $h_1(x_i)=y_i$, $D_2(i)=$~~ [redacted] 2^{-1}

∴ when $h_1(x_i)=y_i$, $D_{[redacted]1}(i)=\frac{0.1\cdot[redacted]}{0.8}=\frac{0.1[redacted]}{0.8[redacted]\times 2}\approx \boxed{0.45985}$ [redacted] 0.0625

when $h_1(x_i)\neq y_i$, $D_{[redacted]1}(i)=\frac{0.1\cdot[redacted]2}{0.8}\approx$ [redacted] 0.25

(d).

~~$H_{final}=sgn(\sum\alpha_t h_t(x_i))$~~ $\varepsilon_2=0.0625\times 4=0.25$

$$\alpha_2=\frac{1}{2}\log_2\left(\frac{1-0.25}{0.25}\right)=\frac{1}{2}\log_2 3\approx 0.792$$

$$H_{final}(x)=sgn\left(\sum_t \alpha_t h_t(x[redacted])\right)$$

$$=sgn(h_1(x)+0.792\,h_2(x))$$

$$=sgn([x>2]+[redacted]0.792\,[y>11])$$

4 Multi-class classification [60 pts]

Consider a multi-class classification problem with k class labels $\{1,2,\ldots k\}$. Assume that we are given m examples, labeled with one of the k class labels. Assume, for simplicity, that we have m/k examples of each type.

Assume that you have a learning algorithm L that can be used to learn Boolean functions. (E.g., think about L as the Perceptron algorithm). We would like to explore several ways to develop learning algorithms for the multi-class classification problem.

There are two schemes to use the algorithm L on the given data set, and produce a multi-class classification:

- **One vs. All:** For every label $i\in[1,k]$, a classifier is learned over the following data set: the examples labeled with the label i are considered "positive", and examples labeled with any other class $j\in[1,k], j\neq i$ are considered "negative".
- **All vs. All:** For every pair of labels $\langle i,j\rangle$, a classifier is learned over the following data set: the examples labeled with one class $i\in[1,k]$ are considered "positive", and those labeled with the other class $j\in[1,k], j\neq i$ are considered "negative".

(a). i. for One v.s. All, k classifiers are learned;
for All vs. All, $\frac{k(k-1)}{2}$ classifiers are learned.

ii. for One vs. All, m examples;
for All vs. All, $\frac{2m}{k}$ examples.

iii. for One vs. All, the label $y_i=\arg\max_{y\in\{1,2,\cdots,k\}}$ [redacted] $W_y^T x_i$
for All vs. All, each label gets $(k-1)$ votes,
there are 2 methods:

(1). Tournament: start with any pair of labels $\langle a, b \rangle$, compare ~~$W_j^T x$ and W~~ $W_a^T x_i$ and $W_b^T x_i$

the winner (with larger $W^T x_i$) compare with another label; continue with winners until ~~f~~ the final winner is found

(2). Majority: compare each pair of labels $\langle a, b \rangle$, the larger one of $W_a^T x_i$ and $W_b^T x_i$ indicates the winner. Then choose the label which wins more often than any other label.

~~(b). I would choose All vs. All because for One against All, each class needs to be separable from the others, which is a strong assumption. Each comparation of One vs. All involves a class with $\frac{m}{k}$ examples and the other class with $\frac{(k-1)m}{k}$ examples, so they may be very unbalanced.~~

iv. The computational complexity for One vs. All is $km \Rightarrow O(km)$; that of All vs. All is $\frac{k(k-1)}{2} \cdot \frac{2m}{k} \Rightarrow O(km)$. ~~d is the dimension of each example.~~

(b). I would choose One vs. All because it ~~is more efficient and~~ only need to store k classifiers instead of $\frac{k(k-1)}{2}$. It is also easier to implement, and performs as good as All vs. All in this case.

(c). ~~Let d be the dimension of each example,~~ Kernal Preceptron calculates each classifier in $O(n^2)$. ∴ One vs. All: $O(km^2)$

All vs. All: $\frac{k(k+1)}{2} \cdot \left(\frac{2m}{k}\right)^2 \Rightarrow O(m^2)$

∴ Now All vs. All is more eficient, so I would prefer All vs. All.

(since it needs $K(x_i, x_j)$ between each pair)

(d). One vs. All: $k \cdot O(dm^2)$ ~~kdm~~ $\Rightarrow O(kdm^2)$

All vs. All: $\frac{k(k-1)}{2} \cdot O(d\frac{4m^2}{k^2}) \Rightarrow O(dm^2)$

∴ All vs. All is more efficient.

(e). One vs. All: $k \cdot O(d^2 m) \Rightarrow O(kd^2 m)$

All vs. All: $\frac{k(k-1)}{2} \cdot O(d^2 \frac{2m}{k}) \Rightarrow O(kd^2 m)$

∴ They are the same efficient.

(f). Counting: for each example, test all $\frac{k(k-1)}{2}$ classifiers

∴ $\frac{k(k-1)}{2} \cdot d \Rightarrow O(k^2 d)$

Knockout: for each example, eliminate 1 classifier at each step until the winner is found. ∴ $(k-1)$ steps.

$(k-1) \cdot d \Rightarrow O(kd)$