005230642 Rui Xu

1.(a) A: if $\exists x_i \in Strain \neq 0$ $\quad i=1,2,\cdots,m$

$A(Strain) = h_z$, $z = x_i$

else

$A(Strain) = h^-$

(b). ① If $h^* = h^-$, the algorithm gives h^-.

$\Rightarrow L(D,h^*)(A(Strain)) = E_{x\sim D}(0\text{-}1\text{-}Loss(h^*(x), A(Strain)(x)) = P_{x\sim D}(h^*(x) \neq A(Strain)(x)) = 0$

$\Rightarrow P(L(D,h^*)(A(Strain)) > \varepsilon) = 0 \Rightarrow m > 0$ is enough

② If $h^* = h_z$:

If the algorithm gives h_z, $\Rightarrow h_z = h^*$,

$\Rightarrow L(D,h^*)(A(Strain)) = 0$

If the algorithm gives h^-, $\Rightarrow h^- \neq h^*$.

$\Rightarrow L(D,h^*)(A(Strain)) = E_{x\sim D}(0\text{-}1\text{-}Loss(h^*(x), A(Strain)(x))) = P_{x\sim D}(h^*(x) \neq A(Strain)(x))$

$= \frac{1}{N}(0+0+\cdots+0+1) = \frac{1}{N} > \varepsilon$

$\Rightarrow P(L(D,h^*)(A(Strain)) > \varepsilon) \Leftrightarrow$ The possibility that the algorithm gives h^- ($h^* = h_z$)

$\Leftrightarrow$ The possibility that $z \notin Strain$

$\Rightarrow P(L(D,h^*)(A(Strain)) > \varepsilon) = (1-\frac{1}{N})^m \leq \alpha$

$\Rightarrow (1-\frac{1}{N})^m \leq e^{-\frac{m}{N}} < e^{-m\varepsilon} \leq \alpha$

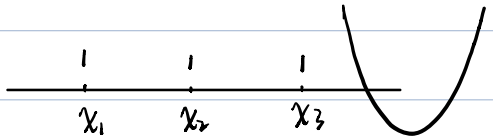
$\Rightarrow m > \frac{1}{\varepsilon}\log\frac{1}{\alpha}$

$\Rightarrow$ If training set Strain has a sample size larger than $\frac{\log(1/\alpha)}{\varepsilon}$, then the probability that the generalization error of A(Strain) larger than ε is at most α.

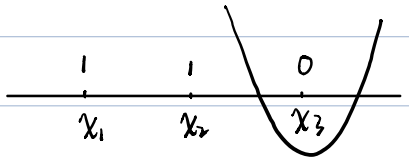
2. VC dimension of H is 3.

① Suppose we have three points x_1, x_2, x_3 and $x_1 < x_2 < x_3$. They can always be shattered by H, no matter how they are labeled.

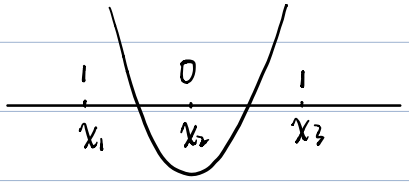
if x_1, x_2, x_3 all labeled 1,
Choose a, b, c to form the curve below:



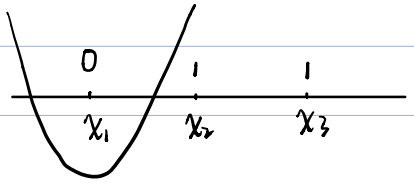
if x_1, x_2 labeled 1, x_3 labeled 0,
Choose a, b, c to form the curve below:



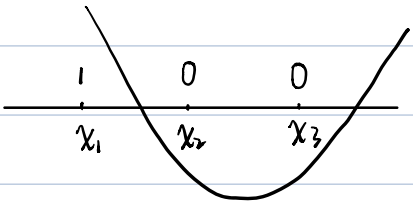
if x_1, x_3 labeled 1, x_2 labeled 0
Choose a, b, c to form the curve below:



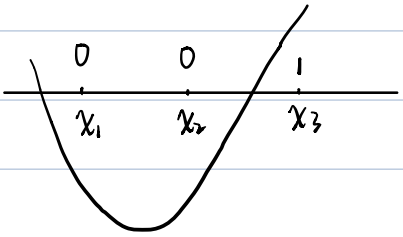
if x_1 labeled 0, x_2, x_3 labeled 1,
Choose a, b, c to form the curve below:



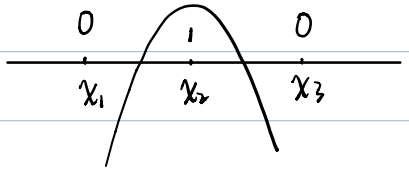
if x_1 labeled 1, x_2, x_3 labeled 0,
Choose a, b, c to form the curve below:



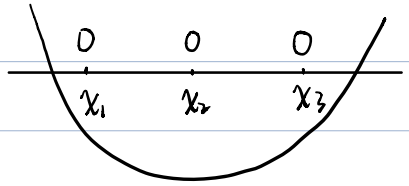
if x_1, x_2 labeled 0, x_3 labeled 1,
Choose a, b, c to form the curve below:



if x_1, x_3 labeled 0, x_2 labeled 1,
Choose a, b, c to form the curve below:



if x_1, x_2, x_3 all labeled 0,
Choose a, b, c to form the curve below:



⇒ Three points can be scattered.

⇒ VC ≥ 3

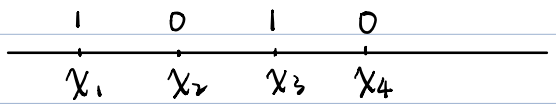
② If we have four points x_1, x_2, x_3, x_4 and $x_1 \le x_2 \le x_3 \le x_4$ and if they are labeled 1, 0, 1, 0

<1> if at least two of x_1, x_2, x_3, x_4 are equal

⇒ at least two equal points have different labels

⇒ It is obvious that these four points cannot be shattered since same points give same $\operatorname{sgn}(ax^2+bx+c)$

<2> if $x_1 < x_2 < x_3 < x_4$ as below:



$$\Rightarrow \begin{cases} ax_1^2+bx_1+c>0 \\ ax_2^2+bx_2+c\le 0 \\ ax_3^2+bx_3+c>0 \\ ax_4^2+bx_4+c\le 0 \end{cases} \Rightarrow \begin{cases} ax_1^2+bx_1+c>ax_2^2+bx_2+c \\ ax_3^2+bx_3+c>ax_2^2+bx_2+c \\ ax_3^2+bx_3+c>ax_4^2+bx_4+c \end{cases} \Rightarrow \begin{cases} (x_1-x_2)(a(x_1+x_2)+b)>0 \\ (x_3-x_2)(a(x_3+x_2)+b)>0 \\ (x_3-x_4)(a(x_3+x_4)+b)>0 \end{cases}$$

$$\& \begin{cases} x_1-x_2<0 \\ x_3-x_2>0 \\ x_3-x_4<0 \end{cases} \Rightarrow \begin{cases} a(x_1+x_2)+b<0 \\ a(x_3+x_2)+b>0 \\ a(x_3+x_4)+b<0 \end{cases} \Rightarrow \begin{cases} a(x_3+x_2)+b>a(x_1+x_2)+b \\ a(x_3+x_2)+b>a(x_3+x_4)+b \end{cases} \Rightarrow \begin{cases} a(x_3-x_1)>0 \\ a(x_2-x_4)>0 \end{cases}$$

$$\& \begin{cases} x_3-x_1>0 \\ x_2-x_4<0 \end{cases} \Rightarrow \begin{cases} a>0 \\ a<0 \end{cases}$$

⇒ No hypothesis in the set can seperate these points.

⇒ VC < 4

⇒ VC dimension of H is 3.

3.(a).

i	Label	Hypothesis 1 (1st iteration)				Hypothesis 2 (2nd iteration)			
		D_0	$f_1 \equiv [x > 2]$	$f_2 \equiv [y > 6]$	$h_1 \equiv [f_1]$	D_1	$f_1 \equiv [x > 10]$	$f_2 \equiv [y > 11]$	$h_2 \equiv [f_2]$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	−	$\frac{1}{10}$	−	+	−	$\frac{1}{16}$	−	−	−
2	−	$\frac{1}{10}$	−	−	−	$\frac{1}{16}$	−	−	−
3	+	$\frac{1}{10}$	+	+	+	$\frac{1}{16}$	−	−	−
4	−	$\frac{1}{10}$	−	−	−	$\frac{1}{16}$	−	−	−
5	−	$\frac{1}{10}$	−	+	−	$\frac{1}{16}$	−	+	+
6	−	$\frac{1}{10}$	+	+	+	$\frac{1}{4}$	−	−	−
7	+	$\frac{1}{10}$	+	+	+	$\frac{1}{16}$	+	−	−
8	−	$\frac{1}{10}$	−	−	−	$\frac{1}{16}$	−	−	−
9	+	$\frac{1}{10}$	−	+	−	$\frac{1}{4}$	−	+	+
10	+	$\frac{1}{10}$	+	+	+	$\frac{1}{16}$	−	−	−

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

(b). $\varepsilon(f_1) = \frac{1}{2} - \frac{1}{2}(8 \times \frac{1}{10} - 2 \times \frac{1}{10}) = 0.2$

$\varepsilon(f_2) = \frac{1}{2} - \frac{1}{2}(7 \times \frac{1}{10} - 3 \times \frac{1}{10}) = 0.3$

$\varepsilon(f_1) < \varepsilon(f_2) \Rightarrow h_1 = f_1$

(c). $\varepsilon_0 = \varepsilon(f_1) = 0.2$

$\alpha_0 = \frac{1}{2}\log_2(\frac{1-\varepsilon_0}{\varepsilon_0}) = 1$

$z_0 = D_0 \times 2^{-\alpha_0} \times 8 + D_0 \times 2^{\alpha_0} \times 2 = 0.8$

$$D_1(i) = \begin{cases} \frac{D_0 \times 2^{-\alpha_0}}{z_0} = \frac{0.05}{0.8} = \frac{1}{16} & \text{label}i = h_1(x_i) \\ \frac{D_0 \times 2^{\alpha_0}}{z_0} = \frac{0.2}{0.8} = \frac{1}{4} & \text{label}i \neq h_1(x_i) \end{cases} \qquad i = 1, 2, \cdots, 10$$

$\varepsilon(f_1) = \frac{1}{2} - \frac{1}{2}(\frac{1}{16} \times 9 + \frac{1}{4} \times 1 - \frac{1}{16} \times 2 - \frac{1}{4} \times 1) = 0.375$

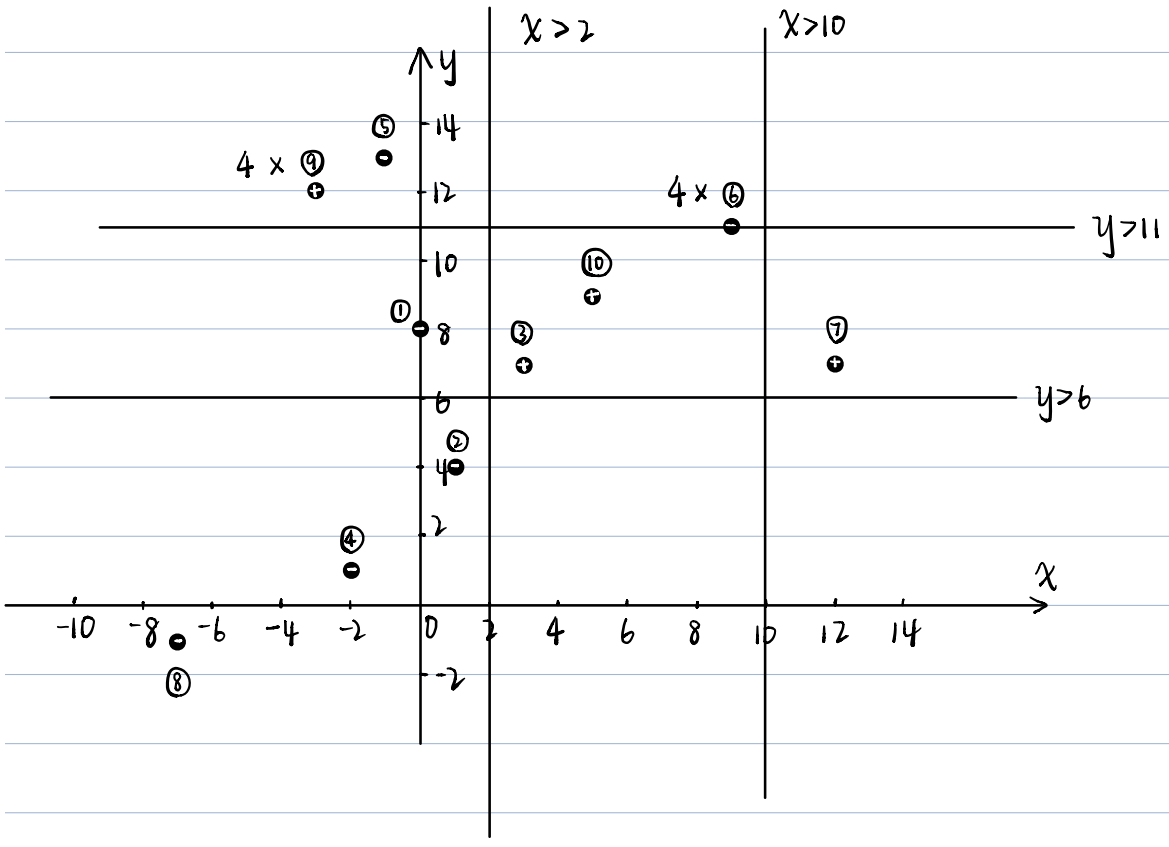
$\varepsilon(f_2) = \frac{1}{2} - \frac{1}{2}\left(\frac{1}{16}\times 4 + \frac{1}{4}\times 2 - \frac{1}{16}\times 4\right) = 0.25$

$\varepsilon(f_1) > \varepsilon(f_2) \Rightarrow h_2 = f_2$

(d). $\varepsilon_1 = \varepsilon(f_2) = 0.25$

$\alpha_1 = \frac{1}{2}\log_2\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right) = 0.8$

$\Rightarrow H_{final} = \text{sign}(\alpha_0 h_1 + \alpha_1 h_2) = \text{sign}([x>2] + 0.8\,[y>11])$



4.(a). i. One vs. All:

learns k classifiers

All vs. All:

learns $C_k^2 = \frac{k(k-1)}{2}$ classifiers

ii. One vs. All:

uses m examples to learn each classifier.
$\frac{m}{k}$ positive examples & $(m - \frac{m}{k})$ negative examples.

All vs. All:

uses $\frac{2m}{k}$ examples to learn each classifier.
$\frac{m}{k}$ positive examples & $\frac{m}{k}$ negative examples.

iii. One vs. All:

Suppose after the learning process, k classifiers give k vectors respectively: $\vec{w_1}, \vec{w_2}, \cdots, \vec{w_k}$, and each classifier classifies one example $\vec{x}$ as positive or negative based on $\text{sign}(\vec{w_i}^T \vec{x})$, $i = 1, 2, \cdots, k$ (like Perceptron algorithm)
The final class label is the label corresponding to the vector that gives the maximum $\vec{w_i} \vec{x}$, $i = 1, 2, \cdots, k$
$\Rightarrow$ label $= \text{argmax}_{y \in \{1, 2, \cdots, k\}} \vec{w_y}^T \vec{x}$

All vs. All:

After applying all the $\frac{k(k-1)}{2}$ classifiers to example $\vec{x}$ and geting different labels, the final class label is the result from majority vote of these labels.

iv. One vs. All:

There are k classifiers and each classifier uses m examples to learn: $k \cdot m$

$\Rightarrow O(mk)$

All vs. All:

There are $\frac{k(k-1)}{2}$ classifiers and each classifier uses $\frac{2m}{k}$ examples to learn: $\frac{k(k-1)}{2} \cdot \frac{2m}{k} = m(k-1)$

$\Rightarrow O(mk)$

(b). I do not have a preference for the two schemes because both of them are able to learn m examples with k labels, and they have the same computational complexity for Perceptron classifier.

(c). Using a KERNEL PERCEPTRON changes the analysis above.

The computational complexity of the KERNEL PERCEPTRON is $O(n^2)$, where n is the number of examples used during training.

$\Rightarrow$ One vs. All:

There are k classifiers and each classifier uses m examples to learn: $k \cdot O(m^2)$

$\Rightarrow O(m^2k)$

All vs. All:

There are $\frac{k(k-1)}{2}$ classifiers and each classifier uses $\frac{2m}{k}$ examples to learn: $\frac{k(k-1)}{2} \cdot O((\frac{2m}{k})^2) = O(m^2)$

$\Rightarrow O(m^2)$

I would prefer All vs. All scheme when using a KERNEL PERCEPTRON because

it has better computational complexity.

(d). One vs. All:

There are k classifiers and each classifier uses m examples with d dimensionality to learn: $k \cdot O(dm^2)$
$\Rightarrow O(m^2dk)$

All vs. All:

There are $\frac{k(k-1)}{2}$ classifiers and each classifier uses $\frac{2m}{k}$ examples with d dimensionality to learn: $\frac{k(k-1)}{2} \cdot O((\frac{2m}{k})^2 d)$
$\Rightarrow O(m^2d)$

All vs. All training paradigm is the most efficient.

(e). One vs. All:

There are k classifiers and each classifier uses m examples with d dimensionality to learn: $k \cdot O(dm^2)$
$\Rightarrow O(md^2k)$

All vs. All:

There are $\frac{k(k-1)}{2}$ classifiers and each classifier uses $\frac{2m}{k}$ examples with d dimensionality to learn: $\frac{k(k-1)}{2} \cdot O(d^2 \frac{2m}{k})$
$\Rightarrow O(md^2k)$

One vs. All and All vs. All are the same efficient.

(f). Counting:

Counting runs all the $\frac{k(k-1)}{2}$ classifiers on one example, and then does the majority vote. $O(k) \Rightarrow O(k^2+k)$

⇒ The overall evaluation time complexity per example is $O(k^2)$.

Knockout:

There are k classes. Since Knockout compares two classes at a time, and if one loses, it will never be considered again.

⇒ Knockout runs (k-1) classifiers to exclude (k-1) classes and remain one class.

⇒ The overall evaluation time complexity per example is $O(k)$.