Machine Learning Homework 4

Bodong Zhang

u0949206

1 PAC Learning

1(a) For each part, there are two possible conditions: used for product or not used for product. Also there are n parts, so the size of the hypothesis space is 2^n.

1(b) For each part, there are four possible conditions: 1: use neither of two pieces 2: use both of pieces

3: only use piece 1, 4: only use piece 2. There are n parts, so the size of hypothesis space is 4^n.

1(c) There are 6 available parts, so the size of hypothesis space is 4^6. Based on Occam's Razor,

m>$\frac{1}{\varepsilon}$*(ln(|H|)+ln($\frac{1}{\delta}$)), |H|=4^6, ε=0.01, $\boldsymbol{\delta}$=0.01, so m>1292.3, at least 1293 examples would the robot have to see.

2 Using Chernoff bounds, assume X_i=1 means an error, so Pr[X_i=1]=p is the true error, p=error_D(h), $\frac{S}{m}$=error_S(D) is training error. So our goal is to make P[error_D(h)> error_S(h)(1+ ε)]as small as possible. Then according to Chernoff bounds,

Pr[every h, error_D(h)<error_S(h)(1+ ε)]= Pr[∃ h, error_D(h)>error_S(h)(1+ ε)]≤|H|Pr[p> $\frac{S}{m}$(1+ ε)]= |H| Pr[$\frac{S}{m}$<(1- $\frac{\varepsilon}{1+\varepsilon}$)p]≤|H| $e^{-\text{mp}(\frac{\varepsilon}{1+\varepsilon})^2/2}$≤ $\boldsymbol{\delta}$, so m≥$\frac{2(1+\varepsilon)^2}{p\varepsilon^2}ln\frac{|H|}{\delta}$.

So with probability 1- $\boldsymbol{\delta}$ that true error is no worse than (1+ ε) error_S(h), the number of training examples should at least be $\frac{2(1+\varepsilon)^2}{p\varepsilon^2}ln\frac{|H|}{\delta}$.

2 VC Dimensions

1. Suppose VC dimension is larger than $\log_2$|C|, then there exist one subset of size $\log_2$|C|+1 that can be shattered, then the size of hypothesis space is $2^{log_2|C|+1}$>|C|. So there is contradiction, its VC dimension is at most $\log_2$|C|.

2. (a) Assume |X|>>k, then choose any subset of size k, no matter how we shatter it, we can always find a way to represent it since |{x:h(x)=1}|=k. But if we choose subset of size k+1, then if all of the element in the subset is positive, it will contradict with the constraint that |{x:h(x)=1}|=k. So the VC dimension is k.

2.(b) Assume |X|>>k, then choose any subset of size 2k+1, no matter how we shatter it, it is always true that |{x:h(x)=1}|≤k or |{x:h(x)=0}|≤k, so this hypothesis can give correct label of it. But if size of subset is 2k+2, if there are k+1 positive elements and k+1 negative elements, then the hypothesis can not label it. So the VC dimension is 2k+1.

3. In a data set of size 3, the dataset can be shattered. Also, dataset of size 4 can also be shattered. In fact, as long as there are no more than two separated intervals that represent positive, the dataset can always be shattered. So the simplest case that can not be shattered is that we have three separated intervals that represent positive, for example: + - + - +. In this case, it can not be represented. So the VC dimension is 4.

4. In dataset of size 1, it can definitely be shattered. In dataset of size 2, if coordinates of points are (0,0),(2,0), then no matter how we shatter it , the hypothesis can still label it. (If (0,0),(2,0) are both positive, set a=-10,b=10, if (0,0),(2,0) are both negative, set a=10, b=-10, if only (0,0) is positive, set a=-1, b=1,if only (2,0) is positive, set a=1, b=3). But if size of subset is 3, for convenience, we rotate it clockwise by 45 degrees. Then in three points, there is one point that horizontal coordinate is no smaller than the minimum of the rest two, and vertical coordinate is no smaller than the minimum of the rest two. If we set it to negative, and set the rest as positive, there is no way the hypothesis can label it. So the VC dimension is 2.

5. $H_1 \subseteq H_2$, so for any subset that can be shattered by H_1, it must can be shattered by H_2 because H_2 has higher label flexibility, assume $VC(H_1)=d$, which means that for a subset of size d, it can be shattered by H_1. So this subset can also be shattered by H_2, so $VC(H_2) \geq d = VC(H_1)$, then $VC(H_1) \leq VC(H_2)$.

3. AdaBoost

The process is below.

1 Initialize $D_1(i)=1/m$ for all i=1,2,…,m

2 For t=1,2,…T:

1. Find a classifier h_t whose weighted classification error is better than chance
2. Computer its vote $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$
3. Update the values of the weights for the training examples

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t}\exp(-\alpha_t y_i h_t(x_i))$$

3 Return the final hypothesis $H_{final}(x) = sgn(\sum_t \alpha_t h_t(x))$

Choose $h_a(x)=sgn(x_1)$, $\varepsilon_1=1/4$, $\alpha_1=\frac{ln3}{2}$, $Z_1=\frac{\sqrt{3}}{2}$,

$X=[x_1,x_2]$ y_i $h_a(x)$ D_1 $D_1(i)y_ih_t(x_i)$ D_2

Write a program for this, the output is below

```
The No. 0 iteration, the error for each function is:
0.25
0.25
0.75
0.25
choose 0th function. error= 0.25
alpha= 0.5493061443340549  Z= 0.8660254037844386
Table:
```

x=[x1,x2]	yi	ha(x)	D	D(i)*y*h(x)	new D
[1.0 1.0]	-1.0	1.0	0.25	-0.25	0.5000000000000001
[1.0 -1.0]	1.0	1.0	0.25	0.25	0.16666666666666666
[-1.0 -1.0]	-1.0	-1.0	0.25	0.25	0.16666666666666666
[-1.0 1.0]	-1.0	-1.0	0.25	0.25	0.16666666666666666

The estimated y is:
1.0 1.0 -1.0 -1.0

The No. 1 iteration, the error for each function is:
function already used, Not applicable
0.16666666666666663
0.4999999999999999
0.16666666666666663
choose 1th function. error= 0.16666666666666663
alpha= 0.8047189562170504 Z= 0.74535599249993
Table:

x=[x1,x2]	yi	ha(x)	D	D(i)*y*h(x)	new D
[1.0 1.0]	-1.0	-1.0	0.5000000000000001	0.5000000000000001	0.3
[1.0 -1.0]	1.0	-1.0	0.16666666666666666	-0.16666666666666666	0.5
[-1.0 -1.0]	-1.0	-1.0	0.16666666666666666	0.16666666666666666	0.09999999999999996
[-1.0 1.0]	-1.0	-1.0	0.16666666666666666	0.16666666666666666	0.09999999999999996

The estimated y is:
1.0 1.0 -1.0 -1.0

The No. 2 iteration, the error for each function is:
function already used, Not applicable
function already used, Not applicable
0.6999999999999998
0.09999999999999998
choose 3th function. error= 0.09999999999999998
alpha= 1.0986122886681098 Z= 0.6
Table:

x=[x1,x2]	yi	ha(x)	D	D(i)*y*h(x)	new D
[1.0 1.0]	-1.0	-1.0	0.3	0.3	0.16666666666666666
[1.0 -1.0]	1.0	1.0	0.5	0.5	0.2777777777777778
[-1.0 -1.0]	-1.0	1.0	0.09999999999999996	-0.09999999999999996	0.4999999999999999
[-1.0 1.0]	-1.0	-1.0	0.09999999999999996	0.09999999999999996	0.05555555555555553

The estimated y is:
-1.0 1.0 -1.0 -1.0

$H_{final}(x)=sgn(\sum_t \alpha_t h_t\,(x))$=sgn(0.549306$h_a$(x)+ 0.804718$h_b$(x)+ 1.09861$h_d$(x))

The No.3 iteration can be ignored because the error is larger than chance

The No. 3 iteration, the error for each function is:
function already used, Not applicable
function already used, Not applicable
0.8333333333333334
function already used, Not applicable
choose 2th function. error= 0.8333333333333334
alpha= -0.8047189562170503 Z= 0.0
Table:

x=[x1,x2]	yi	ha(x)	D	D(i)*y*h(x)	new D
[1.0 1.0]	-1.0	-1.0	0.16666666666666666	0.16666666666666666	0.16666666666666666
[1.0 -1.0]	1.0	-1.0	0.2777777777777778	-0.2777777777777778	0.2777777777777778
[-1.0 -1.0]	-1.0	1.0	0.4999999999999999	-0.4999999999999999	0.4999999999999999
[-1.0 1.0]	-1.0	1.0	0.05555555555555553	-0.05555555555555553	0.05555555555555553

The estimated y is:
-1.0 1.0 -1.0 -1.0