★ Fall 2020 Midterm 1

a] i] 1. This approach is not reasonable because the prediction value depends on number of p & n in the training data & not the value of parameters

2. No. Because such a path might not be possible

3. Is more reasonable because if ~~the~~ we reach a node with missing value, we will ~~still~~ take majority of leaves which is possible for all cases

4. It is a reasonable approach. ex:- mean can be used to impute the values

b] $(x_1 \vee x_2 \vee x_3) \wedge (-x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5) \wedge (-x_4 \vee x_5)$

For +ve,

x_5 has to be 1

x_4 does not matter

x_2 or x_3 has to be 1

x_1 does not matter

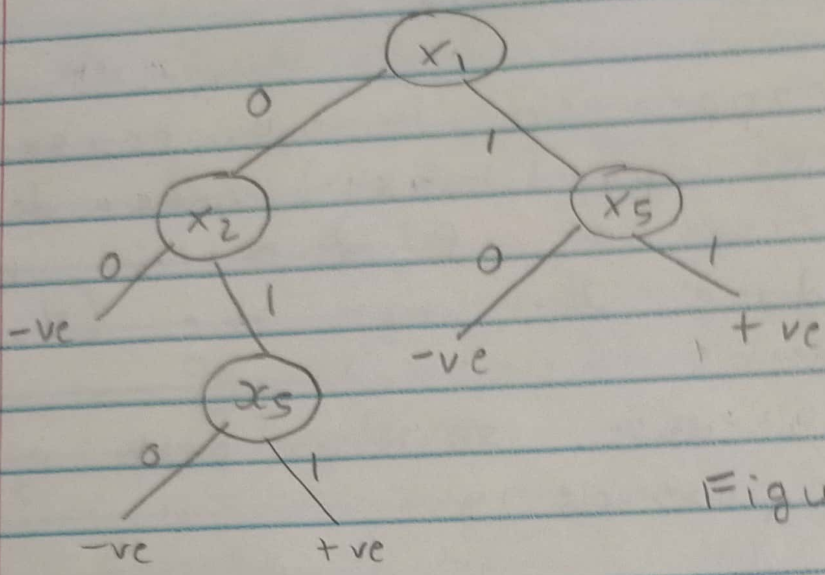


Figure 1

★ Using maths

$$(x_1 \vee x_2 \vee x_3) \wedge (-x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5) \wedge (-x_4 \vee x_5)$$

$$\Rightarrow \left(\underbrace{(x_1 \wedge (x_2 \vee x_3))} \vee \underbrace{\cancel{(x_1 \wedge -x_1)}} \vee \underbrace{(-x_1 \wedge (x_2 \vee x_3))} \right)$$

$$\wedge \left(\underbrace{(x_4 \wedge x_5)} \vee \underbrace{\cancel{(x_4 \wedge -x_4)}} \vee (-x_4 \wedge x_5) \right)$$

$$\Rightarrow \left((x_1 \wedge (x_2 \vee x_3)) \vee (-x_1 \wedge (x_2 \vee x_3)) \right) \wedge \left((x_4 \wedge x_5) \vee (-x_4 \wedge x_5) \right)$$

$$\Rightarrow \left(\cancel{(x_1 \wedge -x_1)} \vee (x_2 \vee x_3) \right] \wedge \left(\cancel{(x_4 \wedge -x_4)} \vee (x_5) \right]$$

$$\Rightarrow \underbrace{(x_2 \vee x_3) \wedge (x_5)}$$

Draw a tree for this Fig 1

c] First of all, converting a complex CNF to Decision tree is not a straight forward formula based approach as we have to simplify the CNF. Not doing so will give us a very large Decision Tree which is inefficient.

CNF form can have redundancy of patterns.

(Answer maybe imprecise because of my lack of knowledg of CNF)

Q2] Likelihood,

$$P(Y|X) = \prod_{k=1}^{m} P(Y=c) \prod_{i=1}^{2} P(X=x_i | Y=c)$$

$$\Rightarrow \prod_{k=1}^{m} \theta^{\{y^k=A\}} (1-\theta)^{\{y^k=B\}} \prod_{i=1}^{2} \left[\left(\frac{e^{-\lambda A_i} (\lambda A_i)^{x^k}}{x^k!} \right)^{\{y^k=A\}} \times \left(\frac{e^{-\lambda B_i} (\lambda B_i)^{x^k}}{x^k!} \right)^{\{y^k=B\}} \right]$$

Log likelihood, (ignore the $P(Y=c)$ term)

$$\Rightarrow \sum_{k=1}^{m} \sum_{i=1}^{2} \left[\{y^k=A\} \left(-\lambda A_i + x^k \ln(\lambda A_i) - \ln(x^k!) \right) + \{y^k=B\} \left(-\lambda B_i + x^k \ln(\lambda B_i) - \ln(x^k!) \right) \right]$$

MLE for λA_i,

$$\frac{\partial L(D|\theta)}{\partial \lambda A_i} = 0$$

$$\Rightarrow \sum_{K=1}^{m} \{y=A\} \left[-1 + \frac{x^K}{\lambda A_i} \right] = 0$$

let $\sum_{K=1}^{m} \{y=A\}$ be d_A

$$\Rightarrow \sum_{K=1}^{m} \{y=A\} = \sum_{K=1}^{m} \frac{x^K \{y=A\}}{\lambda A_i}$$

$$\Rightarrow \boxed{\lambda A_i = \sum_{K=1}^{m} \frac{x^K \{y=A\}}{d_A}}$$

Similarly,

$$\Rightarrow \boxed{\lambda B_i = \sum_{K=1}^{m} \frac{x^K \{y=B\}}{d-d_A}}$$

where $i \in 1,2$

b] $P_r(Y=A) = \frac{3}{7}$, $P_r(Y=B) = \frac{4}{7}$

$\lambda A_1 = \frac{0+4+2}{3}$, $\lambda A_2 = \frac{3+8+4}{3}$

$\lambda B_1 = \frac{6+3+2+5}{4}$, $\lambda B_2 = \frac{2+5+1+4}{4}$

Q3] Out of syllabus

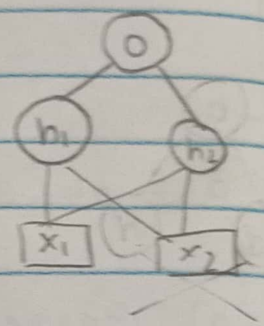
Q4] a] False,
NB has higher bias than LR because it assumes class independent variance which makes it a weaker model on sufficient data size.
Also, NB assumes feature conditionally independence which adds to the bias.

In fact, NB formula has a bias term 

b] No, we cannot represent k out of n features using a linear classifier. because
Assume in data 1 first k features are true, weight setting would be 1 1 1 for w_1 to w_k weights & the w_{k+1} to $w_n = 0$
Now, if data 2 has last k features true, we cannot come with a weigh setting that can satisfy both.

And also,
1 out of 2 True is Not (XOR)
or XNOR which cannot be
represented by a single perceptron

c]



Our function
$x_1 + x_2 = 0$

sign function
$w_1 x_1 + w_2 x_2 + w_0 \geq 0 \rightarrow 1$
elese $\rightarrow -1$

For $\underline{h_1}$,
Use $w_1 = 1$, $w_2 = 1$, $w_0 = 0$

$h_1 = 1$ if $x_1 + x_2 \geq 0$

For h_2,
Use $w_1 = -1$, $w_2 = -1$, $w_0 = 0$

$h_2 = 1$ if $x_1 + x_2 \leq 0$

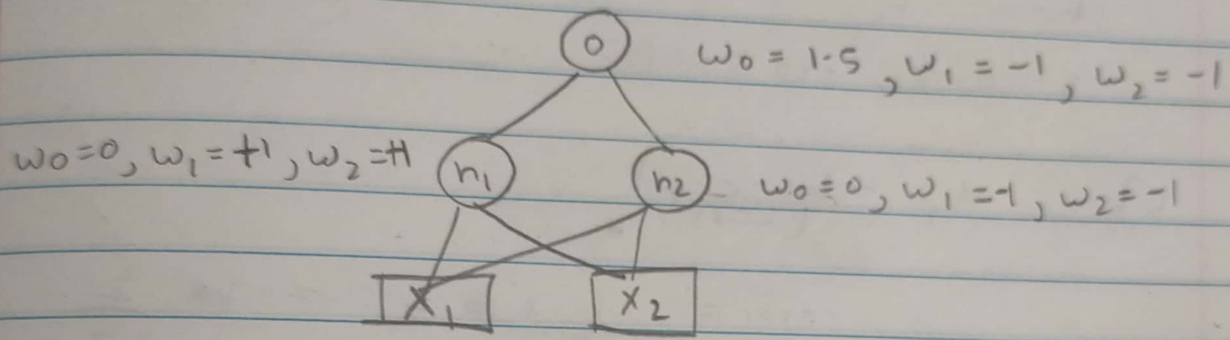
h_1 & h_2 give $x_1 + x_2 = 0$, $\Rightarrow 1$ elcs 0

cuz
if both are true
$x_1 + x_2 \geq 0$
$x_1 + x_2 \leq 0$
then
$x_1 + x_2 = 0$

← we have to create And in output
For O,
$w_0 = +1.5$, $w_1 = -1$, $w_2 = -1$

$o \Rightarrow -1$ if h_1 ~~&~~ h_2 $(x_1 = x_2)$
1 else

Neural net



$\Rightarrow$ Lets test it

$x_1 = 1$ $x_2 = 2$

output $h_1 \Rightarrow \text{sign}(1+2) = 1$

$h_2 \Rightarrow \text{sign}(-1-2) = -1$

$o \Rightarrow \text{sign}(1.5 + 1 - 1) = 1$

$x_1 = -2$, $x_2 = +2$

output $h_1 \Rightarrow \text{sign}(-2+2) = 1$

$\text{sign}(2-2) = 1$

$o \Rightarrow \text{sign}(1.5 - 1 - 1) = \text{si}(-0.5) = -1$

Try out with -1 -2

Q5] Ada Boost

a] we would choose h_1 which makes 2 mistakes. Because it would reduce the error more than h_2. So it is a smarter stupid classifier.

b] $\alpha_1 = \ln\left(\frac{1-err}{err}\right)$

$$error = \frac{\frac{1}{17} \times 2}{1} = \frac{2}{17}$$

$$\alpha_1 = \ln\left(\frac{1-\frac{2}{17}}{\frac{2}{17}}\right) = \ln\left(\frac{15}{2}\right)$$

c] When choosen lerner makes mistake

$$w_i \rightarrow w_i \times e^{\ln\left(\frac{15}{2}\right)}$$

$$w_i \rightarrow w_i \times \frac{15}{2}$$

Where chosen learner does'nt make mistake

$w_i \rightarrow w_i$
remains same
(As per slides)

d] I give up. Please provide u soln.

Qustion 6]

a] we know that

$$w_1 = \frac{\text{Covariance}(x, y)}{\text{Covariance}(x)}$$

$$w_1 = \frac{C_{xy}}{C_{xx}}$$

So, we need 2 statistics (C_{xy}, C_{xx}

Proof

Denominator $= C_{xx} = \frac{1}{m}\left[\sum_{i=1}^{m} x_i^2 + \sum_{i=1}^{m} \bar{x}^2 - 2\sum_{i=1}^{m} x_i \bar{x}\right]$

$$= \frac{1}{m}\left[\sum_{i=1}^{m} x_i^2 + \frac{\cancel{m}\left(\sum x_i\right)^2}{m^{\cancel{2}}} - 2\frac{\left(\sum_{i=1}^{m} x_i\right)^2}{m}\right]$$

$$= \frac{1}{m}\left[\sum_{i=1}^{m} x_i^2 - \frac{\left(\sum_{i=1}^{m} x_i\right)^2}{m}\right]$$

$$= \frac{1}{m^2}\underbrace{\left[m\sum_{i=1}^{m} x_i^2 - \left(\sum_{i=1}^{m} x\right)^2\right]}_{\text{Linear regression denominator}}$$

★ Similarly prove the numerator

b] $w_0 = \frac{\sum y_i - w_1 \sum x_i}{m}$

we need $\bar{x}$ & $\bar{y}$ to calculate w_0

For w_1 we needed C_{xx}, C_{xy}

Counting ~~that~~ at we need 4.

Assuming we have w_1 we need 2.

because,

$$\sum_{i=1}^{m} y_i = m\bar{y} , \sum_{i=1}^{m} x_i = m\bar{x}$$

☆ $w_0 = \bar{y} - w_1 \bar{x}$

c] $$w_0 = \underbrace{\left(\frac{m\bar{y} + y_{m+1}}{m+1}\right)}_{\text{new } \bar{y}} - w\underbrace{\left(\frac{\bar{x}m + x_{m+1}}{m+1}\right)}_{\text{new } \bar{x}}$$

$$w_1 = \frac{\sum_{i=1}^{m+1}\left[x_i - \left(\frac{\bar{x}m + x_{m+1}}{m+1}\right)\right]\left(y_i - \left(\frac{\bar{y}m + y_{m+1}}{m+1}\right)\right)}{\sum_{i=1}^{m+1}\left[x_i - \left(\frac{\bar{x}m + x_{m+1}}{m+1}\right)\right]^2}$$

☆ Just update the old mean's

Question 7] couldn't prove it

Answer source : ChatGPT
(how is this LLM so smart)

Gaussian naive bayes is of form

$$w_0 + \sum_{i=1}^{n} w_i x_i > 0$$

$$w_0 = \ln\left(\frac{\theta}{1-\theta}\right) - \sum_{i=1}^{n} \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma^2}$$

$$w_1 = \frac{\mu_{i1} - \mu_{i0}}{\sigma_i^2}$$

LR uses same linear form

$$w_0 + \sum_{i=0}^{n} w_i x_i > 0$$

(Taking logs on the LR formula)

Both of them use the same linear decision boundaries & this assumption introduces same bias in both models.

(★ Although I wonder why class independence variance is not a stronger bias assumption)

☆ RIP Fall 2020 students

b] Likelihood => $p^5 (1-p)^3$

MLE for p,

let p = 0.6

MLE

=> $(0.6)^5 (0.4)^3$

let p = 0.3

MLE

=> $(0.3)^5 (0.7)^3$

$L(0.6) > L(0.3)$

MLE over possible values is 0.6

c] MAP Estimate

$P(p=0.3) = 0.2$

=> $0.2 \times (0.3)^3 (0.7)^5 = 907578 \times 10^{-9}$

MAP for $P(P=0.6) = 0.8$

=> $0.8 \times (0.6)^5 (0.4)^3 = 3981312 \times 10^{-9}$

Map estimate [redacted] - P = 0.6

(please re-check calculations)

Scanned with CamScanner