☆ Fall 2013 Midterm

Section 1

1] True

Naive baye's classification

$$P(Y=c_i|X) \propto \underbrace{P(Y=c_i)}_{\text{This is prior in the NB rule}} \prod_{i=1}^{n} P(X=x_i|Y=c_i)$$

Naive Bayes uses prior from the training dataset in the form of number of occurances of their class & then uses it to make a classification

---

2] Likelihood of Thumback 1,

$$L(D|\theta) = \theta^3(1-\theta)^3$$

Likelihood of Thumback 2,

$$L(D|\theta) = (2\theta)^3(1-2\theta)^3$$

Log Likelihood,

(t)^3 * (1-2t)^33ln(t) +
3ln(1-2t)MLE,3/t -3/(1-2t) =
0t = 1/3

$\Rightarrow 3\ln(\theta$ ... back 2 ... $(2\theta) + 3\ln(1-2\theta)$

$\Rightarrow \frac{3}{\theta}$ ... $\times 2 \quad \frac{-3}{1-2\theta} \times 2 = 0$

$\Rightarrow \boxed{\theta = \frac{1}{2}}$ | $\boxed{\theta = \frac{1}{4}}$

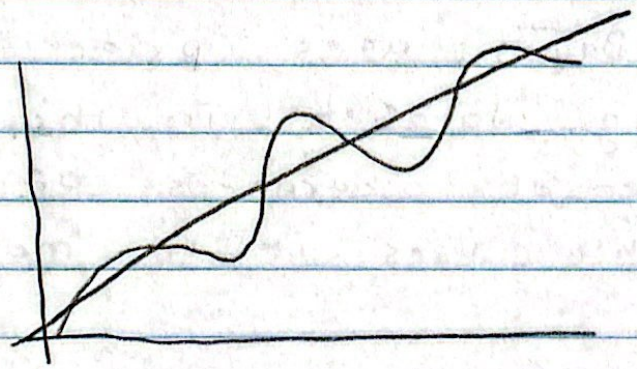
3] False.

Convergence meance loss or error is reduced to a threshold.

A data that is not linearly seperable does not give 0 error but it can still converge by doing some trade off. & it might be the best possible

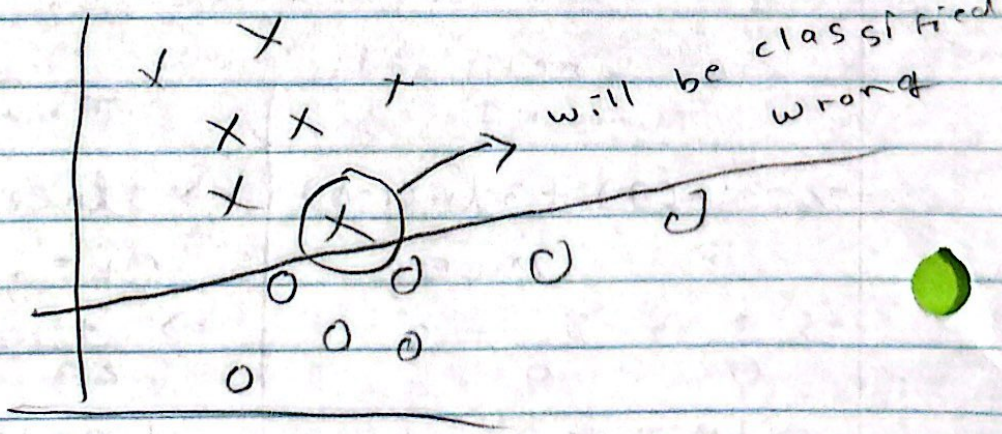
ex:-

what a converged line might look like



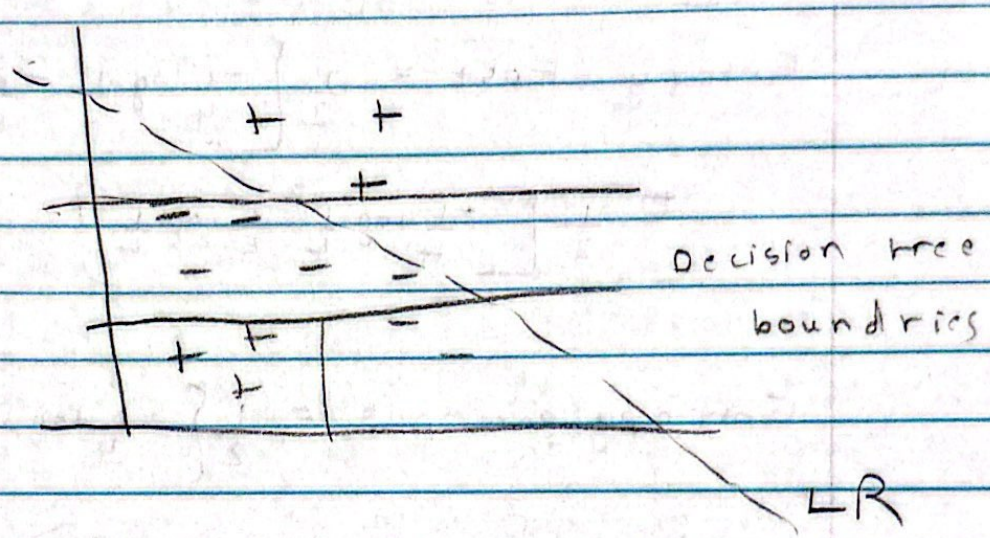
4] False. because on the decision boundary opposite class point can still be more closer

ex:-



5] No. Decision Trees can seperate non-linear data & LR cannot have 100% accuracy on such data

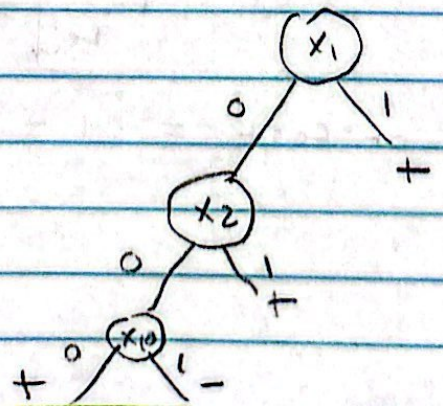
ex:-



Section 2

1] In that case smallest possible tree should have K height & K nodes.

ex:- $X_1 \vee X_2 \vee \neg X_{10}$



2] Entropy price $= \frac{1}{8}(1\log 1) + \frac{4}{8}\left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right]$

$+ \frac{2}{8}\left[-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right]$

Entropy Fast $= \frac{1}{2}\left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right]$

$+ \frac{1}{2}\left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{1}{4}\right]$

Entropy oncampus $= \frac{1}{2}\left[-\frac{1}{4}\log\frac{1}{4} \frac{3}{4}\log\frac{3}{4}\right]$

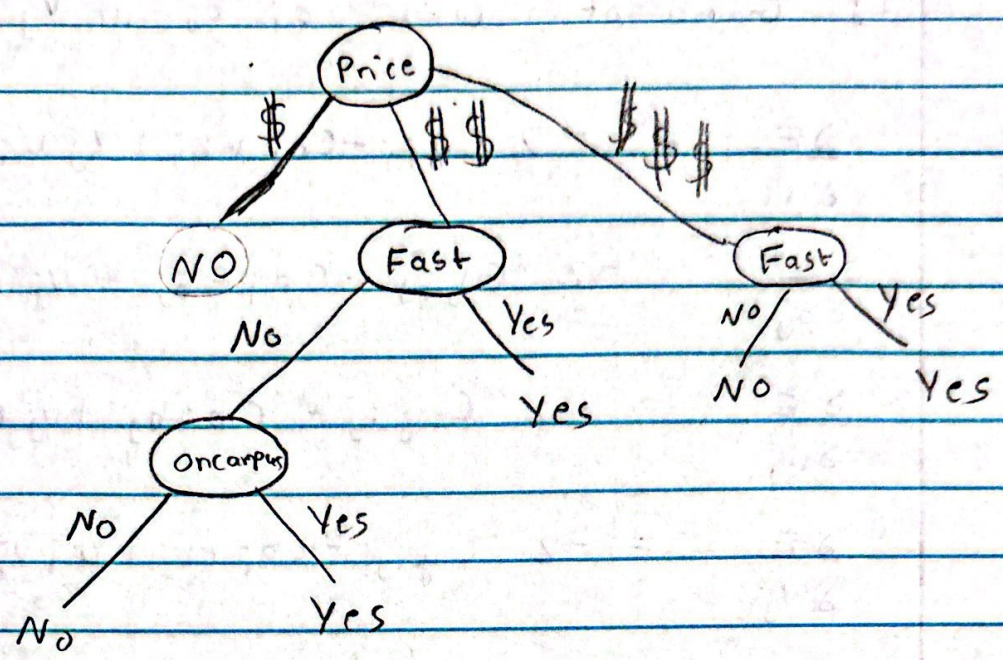
$+ \frac{1}{2}\left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right]$

3] Entropy Like $= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Info gain price $= 1 - \left[\frac{1}{2}\left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right] + \frac{1}{4}\right]$

4] Info gain of fast $= 1 - \left[-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right]$

5]



Section 3

1] Squared error

$$E = (y_{ij} - (a_i + b_j + u_i v_j))^2$$

Optimization task is to

$$\frac{\partial E}{\partial \text{param}} = 0 \quad \text{for } a_i, b_j, u_i, v_j$$

2] - Gradient wrt a_i b_j u_i v_j

$$\frac{\partial E}{\partial a_i} = -2\,(y_{ij} - (a_i + b_j + v_j u_i))$$

$$\frac{\partial E}{\partial b_j} = -2\,(y_{ij} - (a_i + b_j + v_j u_i))$$

$$\frac{\partial E}{\partial u_i} = -2\,(y_{ij} - (a_i + b_j + v_j u_i))\,v_j$$

$$\frac{\partial E}{\partial v_j} = -2\,(y_{ij} - (a_i + b_j + u_i v_j))\,u_i$$

3] Loop until convergence

$$a_i = a_i - \alpha \sum_i \sum_j 2\,(y_{ij} - (a_i + b_j + v_j u_i))$$

$$b_j = b_j - \alpha \sum_i \sum_j 2\,(y_{ij} - (a_i + b_j + v_j u_i))$$

$$u_i = u_i - 2\alpha \sum_i \sum_j (y_{ij} - (a_i + b_j + u_i v_j))\,v_j$$

$$v_j = v_j - 2\alpha \sum_i \sum_j (y_{ij} - (a_i + b_j + u_i v_j))\,u_i$$

End loop

(OK if this is correct)

4] • If missing data m is large then we might not have some values of a_i & b_j. Then we have to use stocastic gradient descent because we want to learn a_i & b_j with the values present in training data & use these values to learn the missing a_i & b_j. If we use batch in this case, we would just be using random values of a_i & b_j for the missing ones (not ideal).

• If missing data m is low then we can use both batch & stochastic gradient descents.

Section 4

1] Forward propogation, $x_1 = 0$, $x_2 = 1$, $y = 0$

$2 \Rightarrow 3 \times 1 + 2 \times 0 = 3$

$3 \Rightarrow 2 \times 1 = 2$

$1 \Rightarrow 3 \times 3 + 2 \times 2 = 13$

③ means output of node ③

2] $\text{Error} = (y - o_1)^2$

$$\frac{1}{2}\frac{\partial E}{\partial w_4} = \underbrace{-(y-o_1)\,(o_1)\,(1-o_1)}_{\delta_1} \times ③$$

$$\frac{1}{2}\frac{\partial E}{\partial w_2} = \underbrace{-(y-o_1)(o_1)(1-o_1)(③)(1-②)}_{\delta_2}\, x_1$$

$$\frac{1}{2}\frac{\partial E}{\partial w_3} = \underbrace{-(y-o_1)\,(o_1)(1-o_1)\,(③)(1-③)}_{\delta_3}\, x_2$$

3] $w_1 = w_1 - \delta_2 x_1 = w_1 - \delta_2$

$w_2 = w_2 - \delta_2 x_1$

$w_3 = w_3 - \delta_3 x_2$

$w_4 = w_4 - \delta_1 \times ②$

$w_5 = w_5 - \delta_1 \times ③$

4] $(x_1 \wedge x_2) \vee (-x_1 \wedge -x_2) \Rightarrow$ XOR Gate

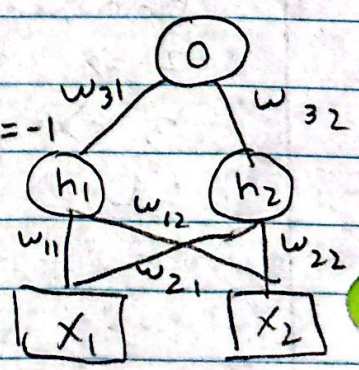
$x_1 \wedge x_2 \leftarrow h_1 \Rightarrow w_0 = 0.5 - 2 = -1.5$

$w_{11} = 1,\ w_{12} = 1$

$-x_1 \wedge -x_2 \leftarrow h_2 \Rightarrow w_0 = 0.5,\ w_{21} = -1,\ w_{22} = -1$

$h_1 \vee h_2 \leftarrow O \Rightarrow w_{31} = 1,\ w_{32} = 1$

$\underbrace{w_0}_{\text{Bias}} = +1.5$



☆ Assuming input is −1, 1 not 0, 1

5] No, because this neural network can represent an XOR, but a single perceptron cannot. (Refer to qn 4)

Section 5]

-> Not taught yet