Assignment 6 Report

Saurabh Daptardar, Srikanth Kuthuru

November 28, 2017

1 Problem 1:

Notation: X denote the random variable, x and $\neg x$ denote `True` and `False` respectively.
Utility function:

$$U(P,B) = \begin{cases} 0 & \neg p, \neg b \\ -100 & \neg p, b \\ 2000 & p, \neg b \\ 1900 & p, b \end{cases}$$

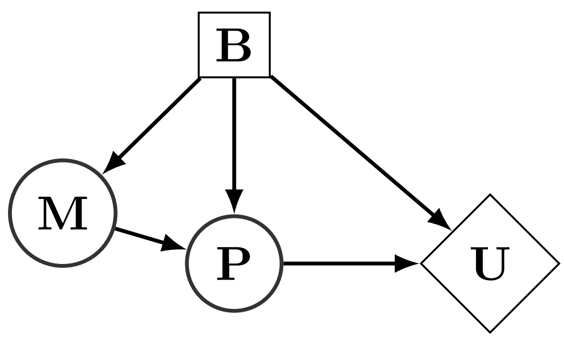


Figure 1: Decision Network

-
- Expected utility of buying the book:

$$\begin{aligned} U(b) &= \mathbf{E}_{P|b}[\ U(P,b)\ ] \\ &= \sum_P U(P,b)\,\Pr(P|b) \\ &= U(p,b)\,\Pr(p|b) + U(\neg p,b)\,\Pr(\neg p|b) \\ \Pr(P|b) &= \sum_M \Pr(P|b,M)\,\Pr(M|b) \\ &= \langle 0.9, 0.1\rangle 0.9 + \langle 0.5, 0.5\rangle 0.1 \\ &= \langle 0.86, 0.14\rangle \\ \implies U(b) &= 1900 \times 0.86 + (-100) \times 0.14 \\ &= \mathbf{1620} \end{aligned}$$

Similarly calculating expected utility of not buying the book:

$$\begin{aligned}
U(\neg b) &= \mathbf{E}_{P|\neg b}[\ U(P, \neg b)\ ] \\
&= \sum_{P} U(P, \neg b) \Pr(P|\neg b) \\
&= U(p, \neg b) \Pr(p|\neg b) + U(\neg p, \neg b) \Pr(\neg p|\neg b)
\end{aligned}$$

$$\begin{aligned}
\Pr(P|\neg b) &= \sum_{M} \Pr(P|\neg b, M) \Pr(M|\neg b) \\
&= \langle 0.8, 0.2 \rangle 0.7 + \langle 0.3, 0.7 \rangle 0.3 \\
&= \langle 0.65, 0.35 \rangle \\
\implies U(\neg b) &= 2000 \times 0.65 + 0 \times 0.35 \\
&= \mathbf{1300}
\end{aligned}$$

- From above values it is clear that the optimal decision for Sam would be to **buy** the book

2 Problem 2:

- Required code is provided in the folder ner.

- For Problem 2.3: Gibbs sampling for linear chain CRFs
  Using chain rule we get:

$$P(y_t/y_{-t}, x_s, \theta) = \frac{P(y_t, y_{-t}/x_s, \theta)}{\sum_{y_t} P(y_{-t}/x_s, \theta)} = \frac{G(y_{t-1}, y_t, x_s, \theta) G(y_t, y_{t+1}, x_s, \theta)}{\sum_{y_t} G(y_{t-1}, y_t, x_s, \theta) G(y_t, y_{t+1}, x_s, \theta)} \tag{1}$$

3 Problem 3:

3.1 Spam Classification

3.1.1 Rule based system

	$k = 10000$	$k = 20000$	$k = 30000$
$n = 1$	0.158255	0.105919	0.471028
$n = 2$	0.205607	0.096573	0.457321
$n = 3$	0.256075	0.110903	0.432399

Table 1: Dev error rate

3.1.2 Linear Classifiers

Code attached

3.1.3 Learning

no. examples	dev error rate
500	0.094081
1000	0.057321
1500	0.043614
2000	0.040498
2500	0.043614
3000	0.036760
3500	0.034268
4000	0.044860
4500	0.031776
5000	0.024922

Table 2: Varying number of examples

3.2 Sentiment Classification

	training error rate	dev error rate
Unigram	0.026764	0.247191
Bigram	0.00	0.224719

Table 3: Varying number of examples

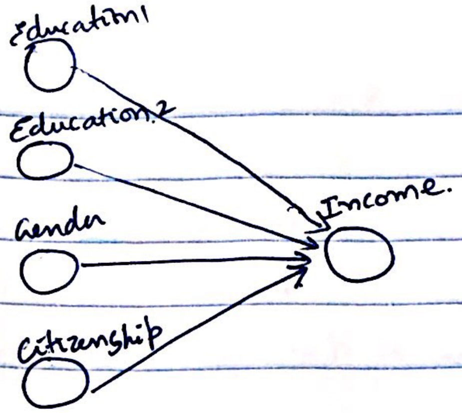
For rest of the parts find code attached.

4 Problem 4:

Code is provided the image classification folder.

5 Problem 5:

5 a)



Perceptron:

Each node above is binary. Since, education has ~~Education1 = {~~ 3 categories, we use 1-hot encoding ~~and~~ which results in 2-binary nodes.

Education

	~~Binary~~ value
BS	→ 0 0
MS	→ 0 1
PhD	→ 1 0

Gender

Male	→ 0
Female	→ 1

Citizenship.

US	→ 0
non-US	→ 1

Income (Output node):

≤50K	→ 0
>50K	→ 1

Scanned by CamScanner

Weights

• Samples

W_1	W_2	W_3	W_4	b
0	0	0	0	-0.5
0	1	0	1	0.5
0	1	-1	1	-0.5
-1	1	-1	0	-1.5
-1	2	0	0	-0.5
-1	2	0	0	-0.5
-1	2	0	0	-0.5
0	2	0	0	0.5
0	2	-1	-1	-0.5
1	2	0	-1	0.5

• Yes, It would converge to [1, 2, 0, -1, 0.5]
This set of weights result in zero-training error. So, unless the training data changes, these weights will not changes.

b) ~~Educ~~ Education

Income

	$\leq 50K$	$\geq 50K$	
BS	4	0	4
MS	0	2	2
PhD	2	2	4
	6	4	

$$\text{Entropy} = (0)\frac{4}{10} + 0\left(\frac{2}{10}\right) + \cancel{0.5}(1)\left(\frac{4}{10}\right)$$

$$= 0.4$$

$$\frac{1}{\log 10} = \frac{0.29}{\ln 2} = \frac{\ln 10}{\ln 2} = 0.97$$

Initial Entropy = $-0.6 \log 0.6 - 0.4 \log 0.4 = 0.97$

Info gain = 0.57

For Gender

Gender	Income ≤50K	>50K
Male	3	2
Female	3	2

Entropy $\left[-(0.6\log 0.6 + 0.4\log 0.4)\right](0.5+0.5)$

$= 0.97$

Info gain = 0.

Citizenship

C	Income ≤50K	>50K
US	3	3
non-US	3	1

Entropy $\left(\frac{6}{10}\right)\left(-0.5\log 0.5 - 0.5\log 0.5\right) + \frac{4}{10}\left[-0.75\log 0.75 - 0.25\log 0.25\right]$

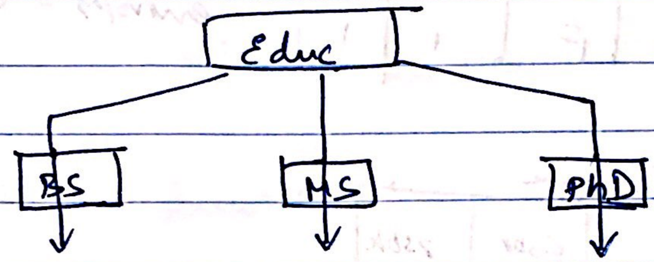
$= 0.6 + (0.4)(0.811)$

$= 0.924$

Info gain = 0.045

∴ Root Node = "Education"

– Because it has maximum info gain.



BS

Gender \ Income	≤ 50k	> 50k
M	~~1~~2	0
F	2	0

C \ Income	≤ 50k	> 50k
US	3	0
Non-US	1	0

"NO SPLIT" because info gain = 0 in both cases.

MS

Gender \ Income	≤ 50k	> 50k
M	0	1
F	0	1

info gain = 0 in both cases (Gender, Citizenship)

"NO SPLIT"

PhDn

Gender

	Income	
	≤50K	>50K
M	1	1
F	1	1

entropy = 0.5

C

	Income	
	≤50K	>50K
US	0	2
Non-US	2	0

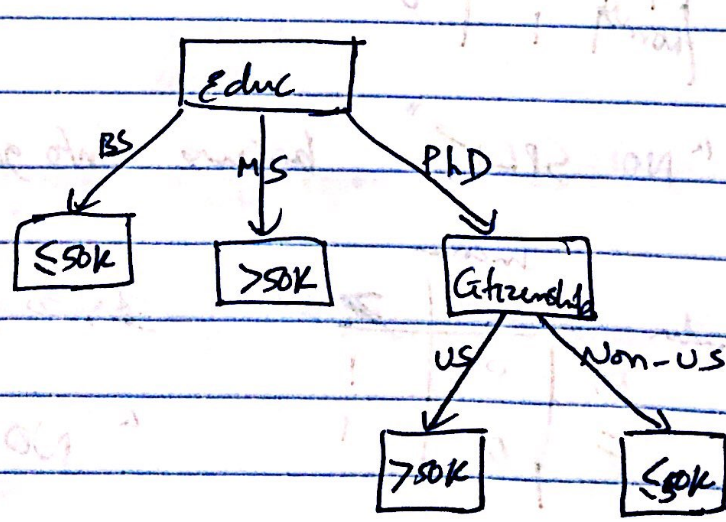
→ Entropy = 0.

info gain = 0.57 − (0)

= 0.57

split at citizenship

Final Tree:



Test set prediction.

1. PhD male US → >50K
2. PhD male non US → ≤50K
3. MS female non US → >50K

c) Neural Networks:

RelU activation f^n.

- Same encoding as the perceptron in ⓐ.

max_iterations = 2500

-

n_hidden layers	C.V. Training error (log-loss)	~~Test~~ Prediction
2	~~0.56~~ ~~0.33~~ 0.49	1 0 1
3	~~0.45~~ 0.127	1 0 1
4	~~0.55~~ 0.114	1 0 1
5	~~0.35~~ 0.078	

0.078 → gives same ~~correct~~ 1 0 1 prediction on test set as Decision Tree.

- Final predictions - All of them predict the same on the Test set.