NAME:- SAM DAVID CHRIST DOSS PUSHPAM

10601 B - HOMEWORK-2

PROBLEM 1:- INDEPENDENT EVENTS AND BAYES THEOREM

a) Prove:- $P(A/B) = \frac{P(B|A)P(A)}{P(A)}$

$P(A,B) = P(A/B)P(B) = P(B|A)P(A)$ => From chain rule,

$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ This is bayes theorem

b) ~~Prove~~:- $P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$

Given:- $\bigcup_{i=1}^{n} A_i = \Omega$

$P(B) = P(B|\Omega) = P(B|\bigcup_{i=1}^{n} A_i)$

$= P(B|A_1, A_2, \ldots A_n)$

Since A_i's are disjoint, $\{A_1\}$ & $\{A_2, A_3, \ldots A_n\}$ are disjoint so $= P(B|A_1) + P(B|A_2, A_3, \ldots A_n)$

$= P(B|A_1) + P(B|A_2) + P(B|A_3, \ldots A_n)$

Similarly, $= \sum P(B|A_i)$

b) Prove :- $P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$

$$\Omega = \cup_i A_i$$

$$B = B \cap \Omega$$
$$= B \cap (\cup_i A_i)$$
$$B = \cup_i (B \cap A_i) \Rightarrow \text{by distributive law}$$

$$\Rightarrow P(B) = P(\cup_i (B \cap A_i))$$

$$= \sum_i P(B \cap A_i)$$
$$P(B) = \sum_i P(B|A_i)\, P(A_i) = \sum_i P(B|A_i)P(A_i)$$

c)

From problem a),
$$P(A_i|B) = \frac{P(B|A_i)\ P(A_i)}{P(B)}$$

From problem b),
$$P(B) = \sum_i P(B|A_i)\, P(A_i)$$

Combining both gives,
$$P(A_i|B) = \frac{P(B|A_i)\ P(A_i)}{\sum_i P(B|A_i)P(A_i)}$$

d) 1) Using the chain rule, its True.

$$P(A,B,C) = P(A|B,C)\,P(B,C)$$
$$= P(A|B,C)\,P(B|C)\,P(C)$$

True because of chain rule.

2) $P(A,B) = P(A|B)\,P(B|A)$ ⇒ It's False.

Because,

$$P(A,B) = P(A|B)\,P(B)$$

3) $P(A,B,C) = P(B|A,C)\,P(C,A)$ ⇒ True

By chain rule,

$$P(A,B,C) = P(B/A,C)\,P(A,C) = P(B/A,C)\,P(C,A)$$

4) $P(A,B,C) = P(B|A,C)\,P(C,A)\,P(C)$ ⇒ False.

By chain rule,

$$P(A,B,C) = P(B/A,C)\,P(A,C)$$
$$= P(B/A,C)\,P(A|C)\,P(C)$$

5) $P(A,B) = P(A)\,P(B)$ ⇒ False

Because, $P(A,B) = P(A|B)\,P(B)$

e) $$E(x) = 0 \cdot P(x=0) + (-1)\,P(x=1)$$
$$= -P(A)$$

$E(x) + P(A) = 0$ ⇒ This result is not true if the value is 1. The result will be

$$E(x) = P(A)$$

PROBLEM 2 : Maximum likelihood estimation.

a). likihood function is,

$$\Rightarrow \prod_{i=1}^{n} (1-\theta)^{k} \theta$$

$$\Rightarrow (1-\theta)^{\sum_i x_i} (\theta)^{n}$$

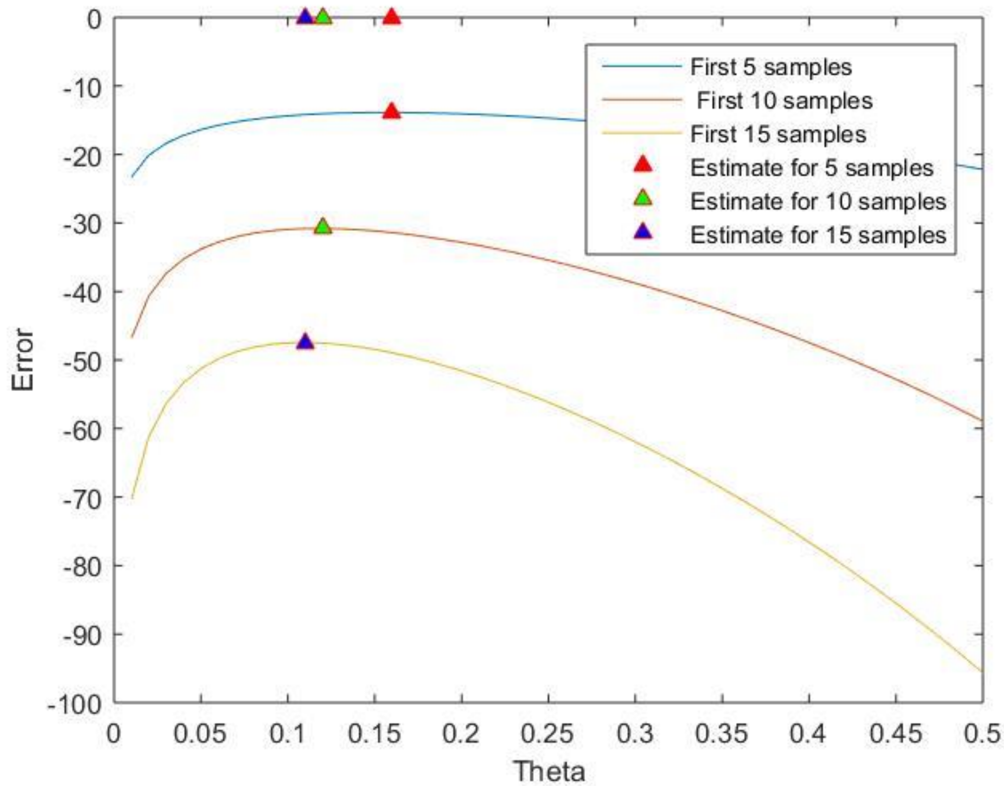
The log is likelihood is given by,

$$l(\hat{\theta}) = \log\left((1-\theta)^{\sum_i x_i} \theta^{n} \right)$$

$$l(\hat{\theta}) = \left(\sum_i x_i\right) \log(1-\theta) + n \log(\theta)$$

The log of likelihood depends on the sum of variables. It does not depend on the order of the variables.

B)



Plot of Theta vs Error for different number of samples

The plot shows the Error estimate for different n samples. Also the location of the estimator has been marked with the triangle symbols in the curve as well as on the axes. The values of theta obtained from the plots are,

Θ_1=0.1600

Θ_2=0.1200

Θ_3=0.1100

MATLAB CODE : 2b

```
clc
clear
x=[1 0 3 5 18 14 5 7 13 9 0 17 4 24 3]' ;
t=0 ;
for i=5:5:15
    a = x(1:i,:) ;
```

```
    theta=[0.01:0.01:0.5] ;
    t=t+1 ;
    l(t,:) = sum(a)*log(1-theta) + i*log(theta) ;
    figure
end

[a1,b1]=max(l(1,:),[],2) ;
[a2,b2]=max(l(2,:),[],2) ;
[a3,b3]=max(l(3,:),[],2) ;

plot(theta,l(1,:),theta,l(2,:),theta,l(3,:)) ;
xlabel('Theta'), ylabel('Error') ;
hold on ;

plot(theta(b1),0,'r^','markerfacecolor',[1 0 0]) ;
plot(theta(b2),0,'r^','markerfacecolor',[0 1 0]) ;
plot(theta(b3),0,'r^','markerfacecolor',[0 0 1]) ;

plot(theta(b1),a1,'r^','markerfacecolor',[1 0 0]) ;
plot(theta(b2),a2,'r^','markerfacecolor',[0 1 0]) ;
plot(theta(b3),a3,'r^','markerfacecolor',[0 0 1]) ;
legend('First 5 samples',' First 10 samples','First 15 samples','Estimate for
5 samples','Estimate for 10 samples','Estimate for 15 samples') ;
```

c) Closed form expression,

$\hat{\ell}(\theta) = \log(\text{likelihood}) = \left(\sum_i x_i\right)\left(\log(1-\theta)\right) + n \log(\theta)$

Taking the derivative, w.r.t. θ,

$$\frac{\partial}{\partial \theta}\hat{\ell}(\theta) = \left(\sum_i x_i\right)\frac{1}{(1-\theta)}(-1) + n \cdot \frac{1}{\theta} = 0$$

$$\frac{n}{\theta} = \frac{\sum_i x_i}{(1-\theta)} \Rightarrow \theta = \frac{n}{n+\sum_i x_i}$$

For the first 5 data,

$\sum_i x_i = 27$, $n = 5$

$\Rightarrow \boxed{\theta_1 = 0.15625}$

For the first 10 data,

$\sum_i x_i = 75$, $n = 10$

$\boxed{\theta_2 = 0.11764}$

For the first 15 data,

$\sum x_i = 123$, $n = 15$

$\boxed{\theta_3 = 0.1086}$

The value of θ from the plots are,

$\theta_1 = 0.16$, $\theta_2 = 0.12$, $\theta_3 = 0.11$

The value agrees well with plots

d) loglikelihood becomes more negative n increases.

$\hat{l}(\hat{\theta}) = \sum_i x_i \log(1-\theta) + n\log(\theta)$

$\hat{l}(\hat{\theta}) \propto \sum_i x_i$ and

$\hat{l}(\hat{\theta}) \propto n$, Since $\log(\theta)$ and $\log(1-\theta)$ are both negative, and n and $\sum_i x_i$ both increases with increase in n,

$l(\hat{\theta})$ becomes more negative with increase in n.

Problem 3: Implementing Naive Bayes:

a) $$\hat{y} = \underset{y}{\operatorname{argmax}} \prod_{w=1}^{V} \left(P(x_w \mid Y=y)\right) P(Y=y)$$

Given, $x = \langle x_1, \ldots x_V \rangle$, x_i is conditionall independent of x_j given Y — ①

$$\hat{Y} = \underset{y}{\operatorname{argmax}}\ P(Y=y \mid x)$$

By Bayes theorem,

$$\hat{Y} = \underset{y}{\operatorname{argmax}} \frac{P(x \mid Y=y)\ P(Y=y)}{P(x)}$$

Since, $P(x)$ does not depend on Y, it can removed from denominator

$$\hat{y} = \underset{y}{\text{argmax}}\ P(x \mid \cancel{Y=y})\, P(Y=y)$$

From the rule 1 of conditional independence,

$$\hat{y} = \underset{y}{\text{argmax}} \prod_{i=1}^{V} P(x_i \mid Y=y)\, P(Y=y)$$

b) No. of parameters needed :-
without naive bayes assumption, the number of parameters needed is 2^V
With the hep of naive bayes assumption the number of parameters is $4V$

For, $V \geq 3$, the naive bayes needs less parameters for the log likelihood estimate. So, naive bayes has big gain in making the assumption.

Problem 3,

c) Matlab code :

```
function [D] = NB_XGivenY(XTrain, yTrain)
  alpha = 1.001 ;
  beta = 1.9 ;
  y_1=find(yTrain==1);
  y_2=find(yTrain==2);

  t1=XTrain(y_1,:);
  t2=XTrain(y_2,:);

  D(1,:)=(sum(t1)+alpha-1)/(size(t1,1)+alpha+beta-2);
  D(2,:)=(sum(t2)+alpha-1)/(size(t2,1)+alpha+beta-2);
end
```

d) Matlab code:

```
function [p] = NB_YPrior(yTrain)
  p = 1-mean(yTrain-1) ;
end
```

e) Matlab code:

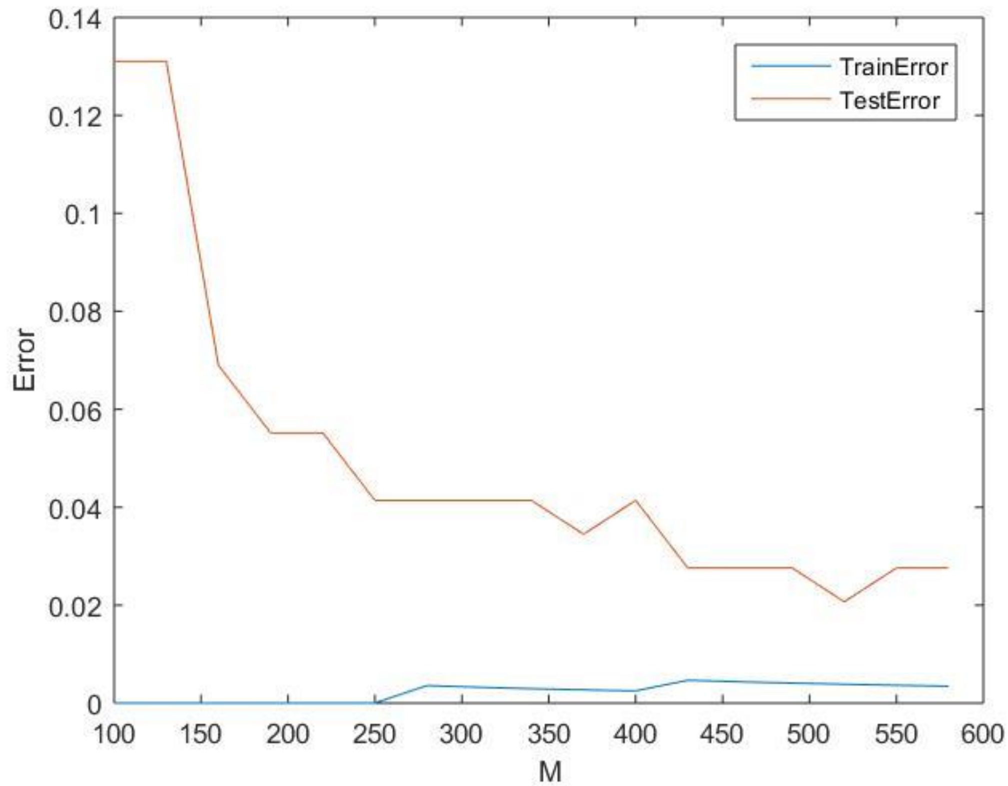
```
function [yHat] = NB_Classify(D, p, XTest)
 [m,n]=size(XTest) ;
  yHat=zeros(m,1) ;

  for i=1:m
    prob=zeros(2,1);
        prob(1)=sum(log(D(1,find(XTest(i,:)==1)))) + sum(log(1-
D(1,find(XTest(i,:)==0))));
        prob(2)=sum(log(D(2,find(XTest(i,:)==1)))) + sum(log(1-
D(2,find(XTest(i,:)==0))));
      yHat(i)=2-ge((prob(1)+log(p)),(prob(2)+log(1-p)));
  end
end
```

f) The train error is 0.0034. The test error is 0.0276.

The error on the training set is lower than the error on the test set. Naive bayes attempts to minimize the error on the training set. We would expect to have lower error on the test set, since we need a better prediction on the test data.

g) The plot is given below,



The training error is very low compared to test set error. The naive Bayes, works better for large number of samples in the training set. As the number of training samples increases, the training set error is increasing and the test set error is decreasing. The Naïve Bayes finds a finds better set of parameters for larger number of samples.

Higher values of m, makes the error in the Test set to go down.

MATLAB CODE : 3g

```
load('HW2Data.mat') ;
t=0;
for m=100:30:580
    D = NB_XGivenY(XTrain(1:m,:), yTrain(1:m,:));
    p = NB_YPrior(yTrain(1:m,:));
    yHatTrain = NB_Classify(D, p, XTrain(1:m,:));
    yHatTest = NB_Classify(D, p, XTest);
    trainError = ClassificationError(yHatTrain, yTrain(1:m,:));
    testError = ClassificationError(yHatTest, yTest);
    t=t+1 ;
    err_train(t)=trainError;
    err_test(t) =testError ;
end
    plot([100:30:580],err_train,[100:30:580],err_test) ;
```

```
xlabel('M') ;
ylabel('Error') ;
legend('TrainError','TestError') ;
```

h)

For $P(X_w=1 \mid Y=y)$:

The words are :

For Y=1 : “the” , “to” , “of” , “in” , “a”

For Y=2 : “a” , “and” , “the” , “to” , “of”

For $P(X_w=1 \mid Y=y)/ P(X_w=1 \mid Y\neq y)$:

The words are :

For Y=1 : “organis” , “reckon” , “favour” , “centr” , “labour”

For Y=2 : “4enlarg” , “5enlarg” , “percent” , “realiz” , “coach”

For $P(X_w=1 \mid Y=y)/\max_v P(X_v=1 \mid Y=y)$:

The words are :

For Y=1 : “the” , “to” , “of” , “in” , “a”

For Y=2 : “a” , “and” , “the” , “to” , “of”

The list of words according to the second metric, which is $P(X_w=1 \mid Y=y)/ P(X_w=1 \mid Y\neq y)$, is more informative. This set of words gives the unique information about each set of Y=1 and Y=2 because this is proportional to the probability of present in Y=y and also inversely proportional to the probability of not present in $Y\neq y$. This inversely proportional term increases the information available about each set, when compared to first and the third metric.

Matlab code : 3h

```
load('HW2Data.mat') ;
D = NB_XGivenY(XTrain(1:m,:), yTrain(1:m,:));
[B,I] = sort(D,2,'descend') ;
words_1=Vocabulary(I(:,1:5)) ;

%part_2
max(D,[],2) ;
[B2,I2]=sort(D(1,:)./D(2,:),2,'descend') ;
[B3,I3]=sort(D(2,:)./D(1,:),2,'descend') ;
words_2=Vocabulary(I2(:,1:5)) ;
words_3=Vocabulary(I3(:,1:5)) ;

%part_3
D1(1,:)=ones(1,26048)*0.9608 ;
D1(2,:)=ones(1,26048)*0.9940 ;
[B4,I4] = sort(D./D1,2,'descend') ;
words_4=Vocabulary(I4(:,1:5)) ;
```