CS 412 — Introduction to Machine Learning
Fall 2017 Assignment 2
Department of Computer Science, University of Illinois at Chicago

Sarit Adhikari
sadhik6@uic.edu

Problem 1. Independence

a)

$P(A=0) = 0.5$
$P(A=1) = 0.5$
$P(B=0) = 0.5$
$P(B=1) = 0.5$

$P(A=0,B=0) = 0.5$
$P(A=0,B=1) = 0$
$P(A=1,B=0) = 0$
$P(A=1,B=1) = 0.5$

$P(A=0 \mid B=0) = P(A=0,B=0) / P(B=0) = 0.5 / 0.5 = 1$
$P(A=0 \mid B=1) = P(A=0,B=1) / P(B=1) = 0/ 0.5 = 0$
$P(A=1 \mid B=0) = P(A=1,B=0) / P(B=0) = 0 / 0.5 = 0$
$P(A=1 \mid B=1) = P(A=1,B=1) / P(B=1) = 0/ 0.5 = 1$

$P(A \mid B) \neq P(A)$

A and B are not independent

b)

A and B are conditionally independent given C if

$P(A \mid BC) = P(A|C)$

$P(A \mid BC) = P(ABC) / P(BC)$

$P(A=0 \mid B=0, C=0) = P(A=0, B=0, C=0) / P(B=0,C=0)$
$= 0.056 / 0.056+0.024 = 0.7$

$P(A \mid C) = P(AC) / P(C)$
$P(A=0 \mid C=0) = P(A=0\ C=0) / P(C=0)$

= 0.056 / 0.484
= 0.115

A and B are not conditionally independent given C

c)

If A ⊥ B (i.e., A and B are independent),

$P(A = 0, B = 0) = P(A=0)\ P(B=0)$
$P(A=0)\ P(B=0) = 0.18$ ………………………….(i)

$P(A = 1, B = 0) = P(A=1)\ P(B=0)$
$P(A=1)\ P(B=0) = 0.28$ ………………………….(ii)

From (i) and (ii),

$P(A=0) / 1 - P(A=0) = 0.18 / 0.28$

$0.28 * P(A=0) = 0.18 * (1 - P(A=0))$
$0.46 * P(A=0) = 0.18$

$P(A=0) = 0.18 / 0.46$

$P(A=0) = 0.3913$

$P(A=1) = 1 - 0.3913 = 0.6087$

$P(B=0) = 0.18/0.3913 = 0.46$

$P(B=1) = 1 - 0.46 = 0.54$

$P(A = 0, B = 1) = P(A=0)\ P(B=1) = 0.3913 * 0.54 = 0.2113$

problem-2

a) The prior distribution is, $\lambda \sim \text{Gamma}(\alpha, \beta)$

$$P(\lambda) = \frac{\beta^{\alpha} \lambda^{\alpha-1} e^{-\lambda\beta}}{\Gamma(\alpha)}$$

Likelihood is given as,

$$p(x_i \mid \lambda) = \lambda e^{-\lambda x_i}$$ (from density function of an exponential distribution)

After getting inputs x_1, x_2, x_3

$$p(x_1, x_2, x_3 \mid \lambda) = \prod_{i=1}^{3} \lambda e^{-\lambda x_i}$$ [Assuming x_1, x_2, x_3 are independent]

$$= \lambda^3 e^{-\lambda \sum_{i=1}^{3} x_i}$$

$$= \lambda^3 e^{-\lambda (x_1 + x_2 + x_3)}$$

Now, the posetrior $\propto$ likelihood $\times$ prior

$$p(\lambda \mid x_1, x_2, x_3) \propto \cancel{\propto} \, P(\lambda) \cdot p(x_1, x_2, x_3 \mid \lambda)$$

$$\propto \frac{\beta^{\alpha} \lambda^{\alpha-1} e^{-\lambda\beta}}{\Gamma(\alpha)} \lambda^3 e^{-\lambda(x_1+x_2+x_3)}$$

$$\propto \lambda^{3+\alpha-1} e^{-\lambda(\beta + x_1 + x_2 + x_3)}$$

$$\boxed{\begin{aligned} \therefore \alpha' &= \alpha + 3 \\ \beta' &= (\beta + x_1 + x_2 + x_3) \end{aligned}}$$ //

b) From question (a),

$$P(\lambda | x_1, x_2, x_3) \sim \text{Gamma}(\alpha+3, \beta+x_1+x_2+x_3)$$

$$\sim \text{Gamma}(5, 1+3.7+4.5+4.8)$$

$$\sim \text{Gamma}(5, 14)$$

c)

MAP estimate is an estimate of an unknown quantity, that equals the mode of posterior distribution

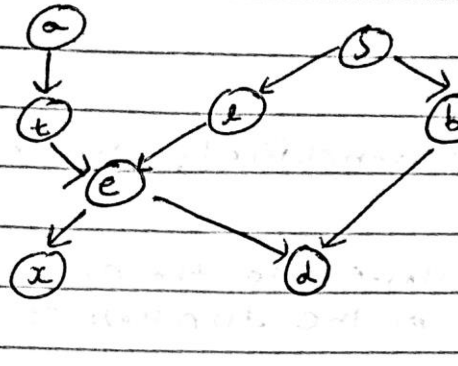
Here, mode of Gamma distribution $= \frac{\alpha-1}{\beta}$
$= \frac{5-1}{14} = \frac{4}{14}$

$$P(x_4 = 3.8 \mid \lambda = (3/14)) = \lambda e^{-\lambda x_4}$$
$$= \frac{4}{14} e^{-3.8 \times \frac{4}{14}}$$

$$= 0.096$$

Problem 3

Problem 3: Bayesian Networks



x = Positive X-ray

d = Dyspnea

e = either tuberculosis or lung cancer

t = Tuberculosis

l = lung cancer

b = bronchitis

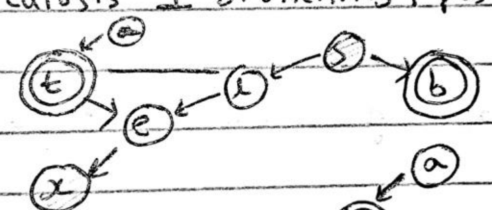
a = visited Asia

s = Smoker

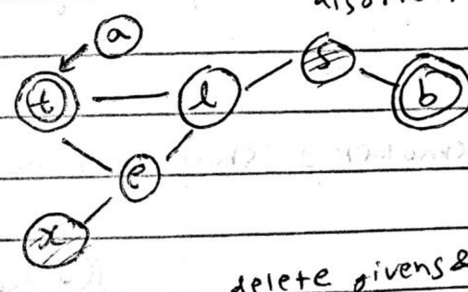
a)

i) tuberculosis $\perp$ bronchitis | positive x-ray, smoker

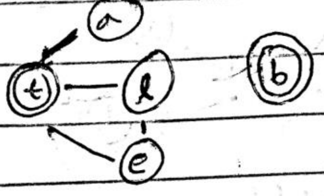
ancestral graph & moralize



disorient



delete givens & edges



$\therefore$ TRUE that tuberculosis $\perp$ bronchitis | positive x-ray, smoker

Ball passing, blocked at (S).

~~⇒Local semantics~~

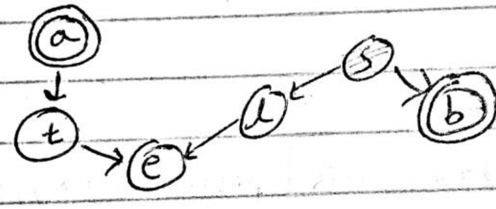
problem-3

a) i) tuberculosis ⊥ bronchitis | positive x-ray, smoker

~~⇒TRUE~~

(balls are allowed to travel in any direction, independent of the direction of edges in graph)

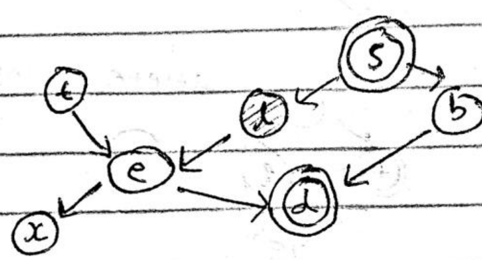
ii) Visit to Asia ⊥ bronchitis | smoker



TRUE

ball passing (ball blocked at (s))

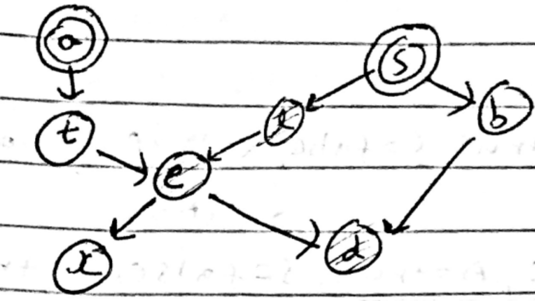
iii) smoker ⊥ shortness of breath | lung cancer



FALSE

ball passing (ball can pass from d to s)

iv) $a \perp s \mid e, l, d$



~~FALSE~~

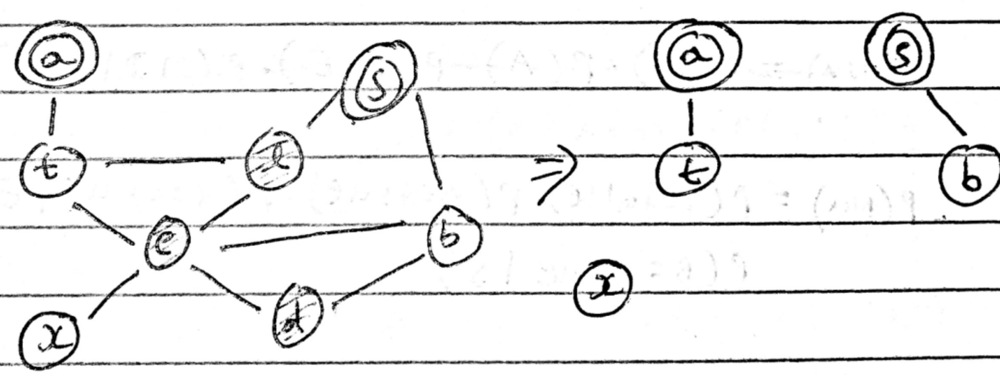
~~ball passing (ball can pass from a to s~~

TRUE

ball cannot pass from 'a' to 's'.

It is blocked at 'e'

<u>confirmation by d-separation</u>



Q.N.3)

b)

i)

$P(B=true \mid A=true, S=false, X=true, D=false)$

$$= \frac{P(B=true, A=true, S=false, X=true, D=false)}{P(A=true, S=false, X=true, D=false)}$$

~~For~~ numerator,

$= P(D=false \mid B=true, E) \cdot P(X=true \mid E) \cdot P(A=true)$
$P(L \mid S=false) \cdot P(B=true \mid S=false) \cdot P(S=false)$
$P(E \mid L, T) \cdot P(T \mid A=true)$

Eliminating t,

$$f_t(E, L, A) = \sum_t P(E \mid L, T) \cdot P(T \mid A=true)$$

Remaining,

$P(D=false \mid B=true, E) \cdot P(X=true \mid E) \cdot P(A=true)$
$P(L \mid S=false) \cdot P(B=true \mid S=false) \cdot P(S=false)$
$f_t(E, L, A=true)$

Eliminating L,

$$f_l(E, A, S) = \sum_L f_t(E, L, A=\text{true}) \cdot P(L \mid S=\text{false})$$

$$P(L=\text{false} \mid S=\text{false})$$

$$= f_t(E, L=\text{false}, A=\text{true}) \, \cancel{P(L \mid S=\text{false})}$$

$$+ f_t(E, L=\text{true}, A=\text{true}) \cdot P(L=\text{false} \mid S=\text{false})$$

remaining factors,

$$P(D=\text{false} \mid B=\text{true}, E) \cdot P(X=\text{true} \mid E) \cdot P(A=\text{true})$$
$$P(B=\text{true} \mid S=\text{false}) \cdot P(S=\text{false}) \, f_l(E, A, S)$$

eliminating E,

$$f_e(B, D, X, S, A)$$

$$= \sum_e P(D=\text{false} \mid B=\text{true}, E) \cdot P(X=\text{true} \mid E) \cdot f_l(E, A=\text{true}, S=\text{false})$$

$$= P(D=\text{false} \mid B=\text{true}, E=\text{true}) \cdot P(X=\text{true} \mid E=\text{true})$$
$$f_l(E=\text{true}, A=\text{true}, S=\text{false})$$
$$+ P(D=\text{false} \mid B=\text{true}, E=\text{false}) \cdot P(X=\text{true} \mid E=\text{false})$$
$$f_l(E=\text{false}, A=\text{true}, S=\text{false})$$

Remaining,

$P(A=true)$, $P(B=true|S=false) \cdot P(S=false)$

$f_e(B, D, X, S, A)$

Now,

- for each possible values of E & L,

E	L
T	T
T	F
F	T

$\cancel{P(E=t|L=t, T=true) \cdot P(T=true|A=true)}$
$\cancel{+ P(E=true|L=true, T=false) \cdot P(T=false|A=true)} = \begin{matrix} 0.1 \\ + \\ 0.9 \\ =1 \end{matrix}$

When, E = True & L = True

$f_t(E, L, A) = P(E=true|L=true, T=true) \cdot P(T=true|A=true)$
$+ P(E=true|L=true, T=false) \cdot P(T=false|A=true)$
$= 0.1 + 0.9 = 1$

When, E = True & L = False,

$f_t(E, L, A) = P(E=true|L=\overset{False}{\cancel{true}}, T=\overset{true}{\cancel{false}}) \cdot P(T=true|A=true)$
$+ P(E=true|L=False, T=true) \cdot P(T=true|A=true)$
$= 0.1$

When, E = False, L = true, $f_t(E, L, A) = 0$

When, E = False, L = False,

$f_t(E, L, A) = P(E=false|L=false, T=true) \cdot P(T=true|A=true)$
$+ P(E=false|L=false, T=False) \cdot P(T=false|A=true)$
$= 0.9$

Also,

When, E = True,

$f_{\ell}(E,A,S) = f_t(E=true, L=false, A=true) \cdot P(L=false|S=false)$
$+ f_t(E=true, L=true, A=true) \cdot P(L=true|S=false)$
$= 0.1 * 0.99 + 1 * 0.01 = 0.099 + 0.01 = 0.109$

When, E = False,

$f_{\ell}(E,A,S) = f_t(E=false, L=false, A=true) \cdot P(L=false|S=false)$
$+ f_t(E=false, L=true, A=true) \cdot P(L=true|S=false)$
$= 0.9 * 0.99 = 0.891$

Finally,

$fe(B, D, X, S, A)$
$= 0.05 * 0.9 * 0.109 + 0.22 * 0.05 * 0.891$
$= 0.004905 + 0.009801 = 0.0147$

Numerator =

$P(A=true) \cdot P(B=true|S=false) \cdot P(S=false)$
$\cdot fe(B, D, X, S, A)$
$= 0.03 * 0.4 * 0.5 * 0.0147 = 0.0000882$

Now, denominator is,

$P(A=\text{true}, S=\text{false}, X=\text{true}, D=\text{false})$

$$= \sum_{e,l,t,b} P(D=\text{false} \mid B,E) \cdot P(X=\text{true} \mid E) \cdot P(E \mid L,T) \cdot P(T \mid A=\text{true}) \cdot P(A=\text{true}) \cdot P(L \mid S=\text{false}) \cdot P(B \mid S=\text{false}) \cdot P(S=\text{false})$$

eliminating t & l, same as before,

Eliminating e,

when, B = true

$f_e(B, D, X, S, A) = 0.0147$ (from numerator computation)

~~when, B = true~~

when, B = false

$f_e(B, D, X, S, A)$

$= P(D=\text{false} \mid B=\text{false}, E=\text{True}) \cdot P(X=\text{true} \mid E=\text{true}) \cdot f_l(E=\text{true}, A=\text{true}, S=\text{false})$

$+ P(D=\text{false} \mid B=\text{false}, E=\text{false}) \cdot P(X=\text{true} \mid E=\text{false}) \cdot f_l(E=\text{false}, A=\text{true}, S=\text{false})$

$= 0.4 * 0.9 * 0.109 + 0.95 * 0.05 * 0.891$

$= 0.03924 + 0.0423$

$= 0.08154$

Eliminating, B,

$f_B(B, D, X, S, A)$

$= \sum_B f_e(B, D, X, S, A) \cdot P(B \mid S = \text{false})$

$= f_e(B = \text{true}, D, X, S, A) \cdot P(B = \text{true} \mid S = \text{false})$
$+ f_e(B = \text{false}, D, X, S, A) \cdot P(B = \text{false} \mid S = \text{false})$

$= 0.0147 * 0.4 + 0.08154 * 0.6$
$= 0.00588 + 0.048924 = 0.055$

Denominator

$= f_B(B, D, X, S, A) * P(A = \text{true}) \cdot P(S = \text{false})$
$= 0.055 * 0.03 * 0.5$
$= 0.000825$

$\text{prob} = \frac{0.0000882}{0.000825} = 0.107$ //

ii) $P(B=true \mid A=true, S=true, X=false, D=true)$

$$= \frac{P(B=true, A=true, S=true, X=false, D=true)}{P(A=true, S=true, X=false, D=true)}$$

Numerator,

$= P(D=true \mid B=true, E) \cdot P(X=\cancel{true}\ false \mid E) \cdot P(A=true) \cdot P(L \mid S=\cancel{false}\ true) \cdot P(B=true \mid S=true)$
$P(S=true) \cdot P(E \mid L, T) \cdot P(T \mid A=true)$

eliminating t,

$$f_t(ELA) = \sum_t P(E \mid L, T) \cdot P(T \mid A=true)$$

$$= P(E \mid L, T=true) \cdot P(T=true \mid A=true) + P(E \mid L, T=false) \cdot P(T=false \mid A=true)$$

E	L	$f_t(E, L, A)$
True	True	1
True	False	0.1
False	True	0
False	False	0.9

From (i)

Eliminating L,

$$f_l(E,A,S) = \sum_l f_t(e,a,l) \cdot P(l|s)$$

When, E = True,

$f_t(E=true, A=true, S=true)$
$= f_t(E=true, A=true, L=false) \cdot P(L=false|S=true)$
$+ f_t(E=true, A=true, L=true) \cdot P(L=true|S=true)$
$= 0.1 * 0.95 + 1 * 0.05 = 0.145$

When, E = False,

$f_t(E=false, A=true, \cancel{L=false}\ S=true)$
$= f_t(E=false, A=true, L=false) \cdot P(L=false|S=true)$
$+ f_t(E=false, A=true, L=true) \cdot P(L=true|S=true)$
$= 0.9 * 0.95 = 0.855$

Eliminating E, $f_e(B, D, X, S, A)$
$P(D=true|B=true, E=true) \cdot P(X=false|E=true)$
$f_l(E=true, A=true, S=true)$
$+ P(D=true|B=true, E=false) \cdot P(X=false|E=false)$
$f_l(E=false, A=true, S=true)$

$= 0.95 * 0.1 * 0.145 + 0.75 * 0.95 * 0.855$

$= 0.013775 + 0.178695$

$= 0.19247$

Numerator $= 0.19247 * 0.03 * 0.5 * 0.75$

$= 0.002165$

For,

Denominator,

eliminating e,

$f_e(B, D, X, S, A)$

when, $B = true \Rightarrow 0.19247$

$B = false \Rightarrow P(D = true \mid B = false, E = true)$
$P(X = false \mid E = true) \cdot f_1(E = true, A = true, S = true)$
$+ P(D = true \mid B = false, E = FALSE) \cdot P(X = false \mid E = false)$
$f_e(E = false, A = true, S = true)$

$= 0.4 * 0.1 * 0.145 + 0.05 * 0.95 * 0.855$

$= 0.0058 + 0.0406125$

$= 0.0464125$

Eliminating B,

$f_B(D, X, S, A) = f_e(B = true, D, X, S, A) \cdot P(B = true \mid S = true)$
$+ f_e(B = false, D, X, S, A)\ P(B = false \mid S = true)$

$= 0.19247 * 0.75 + 0.0464125 * 0.25$

$= 0.14435 + 0.0116 = 0.15595$

Denominator $= 0.15595 * 0.5 * 0.03$

$= 0.00234$

$$p = \frac{0.002165}{0.00234}$$

$= 0.925$ //

Denominator $= 0.155 * 0.5 * 0.03$

$= 0.0023$

~~P~~ Probability $= \frac{0.0022}{0.0023}$

$= 0.96$ //

Problem 4

a)

<u>Output</u>

```
>> nbayes
When , alpha=0.1, Test Set Performance Measure

confusion_matrix =

   138    19
    10   949


accuracy =

    0.9740


precision =

    0.8790


recall =

    0.9324


f_score =

    0.9049
```

<u>Code - Snippet</u>

The code folder consists of main script file
'nbayes.m'

and three function files

'calculate_likelihood.m' ,
'calculate_performance_measure.m' ,
'Count_spam_ham.m'
'predict.m'.

The likelihood is calculated from the words map with smoothing as follows

```
function prob = calculate_likelihood(words_map, word, words_count,alpha)
  prob = (words_map.get(word)+alpha)/(words_count+alpha*20000);

  %if word is not in map, [] is returned which should be taken care of

  if isempty(prob)
     prob = alpha/(words_count+alpha*20000);
  end

end
```

'count_spam_ham.m' function returns the hashmaps spamcounts and hamcounts , along with count of total spam and ham documents which are used to calulate prior

```
alpha = 0.1;

% count occurences for spam and ham
[spamcounts,hamcounts,numspamwords,numhamwords,spam_msg_count,ham_msg_count] =  count_spam_ham(alpha,train_examples);


%calculate prior probability

spam_prior = spam_msg_count / (spam_msg_count+ham_msg_count);
ham_prior = ham_msg_count / (spam_msg_count + ham_msg_count);
```

Prediction is made on the test data set and performance measures are returned by functions 'predict.m' and 'performance_measure.m' respectively.

```
% Prediction
[actual_label,predicted_label] = predict(spam_prior,ham_prior,spamcounts,hamcounts,numspamwords,numhamwords,test_examples,alpha);

[confusion_matrix, precision,recall,f_score,accuracy] = calculate_performance_measure(actual_label,predicted_label);
```

In predict.m,

For each training example, loop is run for each word and likelihood for that word is calculated. The product of likelihood of each word is multiplied with prior probability to get a number proportional to posterior probability (*please note that this value is not actual posterior probability. In fact, we don't need to calculate exact posterior probability because the normalizing sum would be same for both spam_probability and ham probability.*) The predicted label along with actual label is returned from the function.

```
actual_label = [];
predicted_label = [];
for i =1:length(examples)
  actual_label(i) = examples(i).spam;

  product_likelihood_spam = 1;
  product_likelihood_ham = 1;
  for j=1:length(examples(i).words)
    current_word = char(examples(i).words(j));
    product_likelihood_spam = product_likelihood_spam * calculate_likelihood(spamcounts,current_word,numspamwords,alpha);
    product_likelihood_ham = product_likelihood_ham * calculate_likelihood(hamcounts,current_word,numhamwords,alpha);
  end

  %calculating proportional posterior probability ,
  %not actual posterior probability , not divided by probability of word
  posterior_prob_spam =  spam_prior * product_likelihood_spam;
  posterior_prob_ham = ham_prior * product_likelihood_ham;

  if posterior_prob_spam > posterior_prob_ham
    predicted_label(i) = 1; %spam
  else
    predicted_label(i) = 0; %not spam
  end

end
end
```

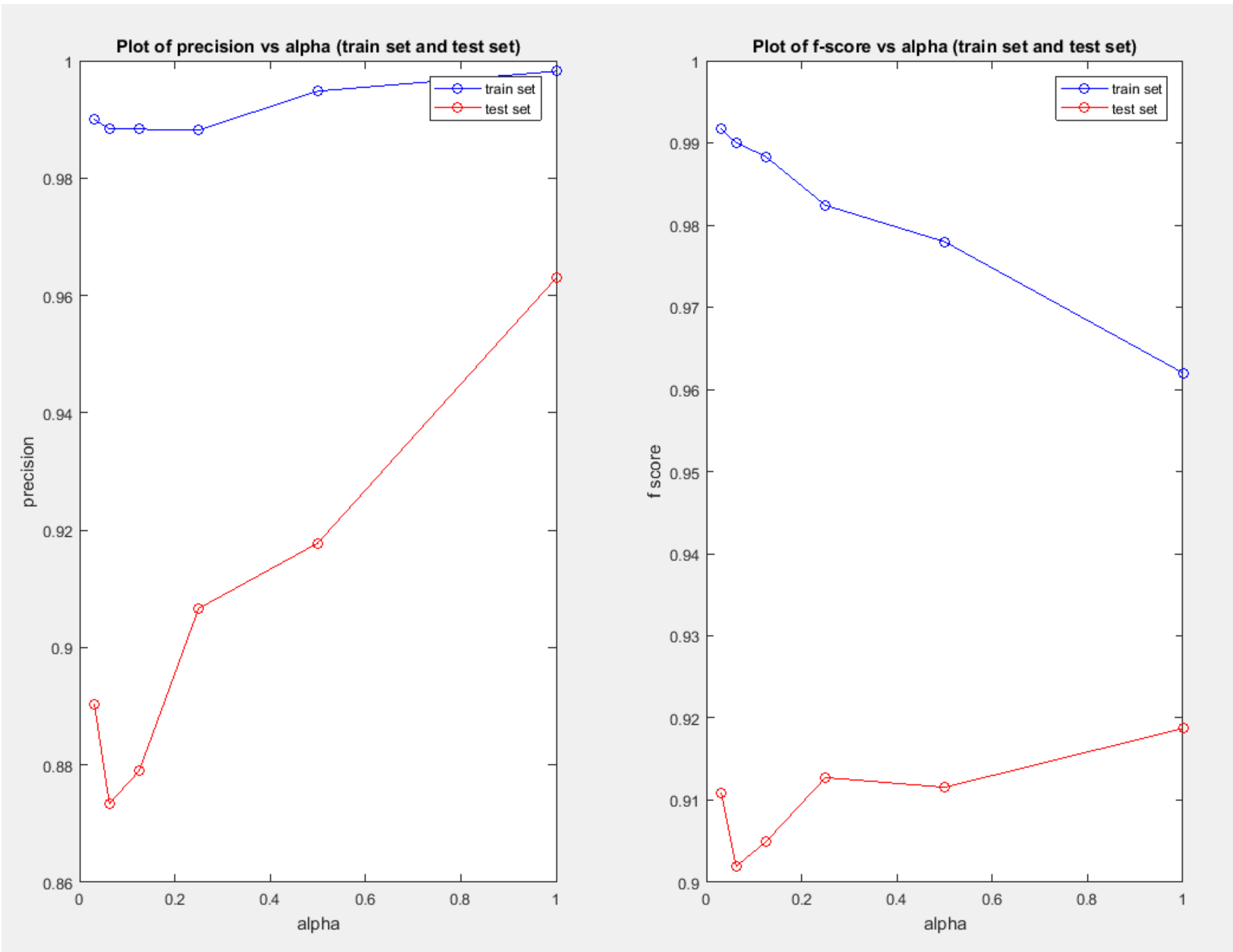
"calculate_perormance_measure.m" calculates true positives, true negatives, false positives and false negatives by comparing true labels and actual labels . Then it calculates accuracy, precsion, recall and f-score from it.

```
for i=1:length(actual_label)
  if actual_label(i)==1 && predicted_label(i)==1
    tp = tp+1;
  elseif actual_label(i)==1 && predicted_label(i)==0
    fn = fn+1;
  elseif actual_label(i)==0 && predicted_label(i)==1
    fp = fp+1;
  else
    tn = tn+1;
  end

 end

confusion_matrix = [tp fp;fn tn];

precision = tp / (tp+fp);
recall = tp / (tp+fn);

f_score = (2 * precision * recall) / (precision+recall);
accuracy = (tp+tn) / (tp+fp+tn+fn);
```

b)

<u>Output</u>

Following graph shows the effect of change of α on the classifier performance



<u>Code Snippet</u>

```
precision_train_list = [];
f_score_train_list = [];
precision_test_list = [];
f_score_test_list = [];

alpha_list =[2^-5, 2^-4 , 2^-3, 2^-2,2^-1,2^0];
for alpha= alpha_list
```

```
…………………………
………………………………..
  Peformance measure calulated as Q 4(a)



…………………………
……………………………..


  %update precision and f score list
  precision_train_list = [precision_train_list precision];
  f_score_train_list = [f_score_train_list f_score];

……………………………………..
…………………………………..

  %update precision and f score list
  precision_test_list = [precision_test_list precision];
  f_score_test_list = [f_score_test_list f_score];


end

%plot

  subplot(1,2,1)
  plot(alpha_list,precision_train_list,'o-b',alpha_list,precision_test_list,'o-r');
  xlabel('alpha');
  ylabel('precision');
  title('Plot of precision vs alpha (train set and test set)');
  legend('train set','test set');

  subplot(1,2,2)
  plot(alpha_list,f_score_train_list,'o-b',alpha_list,f_score_test_list,'o-r');
  xlabel('alpha');
  ylabel('f score');
  title('Plot of f-score vs alpha (train set and test set)');
  legend('train set','test set');
```

Please refer to lines 70 to 130 in nbayes.m file for relevant code for this question