Answer of Homework1

3120000060　秦昇

1. Machine Learning Problems

1) BF
2) A
3) AD
4) BG
5) AD
6) AD
7) BF
8) AE
9) BG

(b)

False.
There must be test data here, maybe the best way is to use Cross Validation because our purpose is to minimize the test data error so that we can get the best model.

2. Bayes Decision Rule

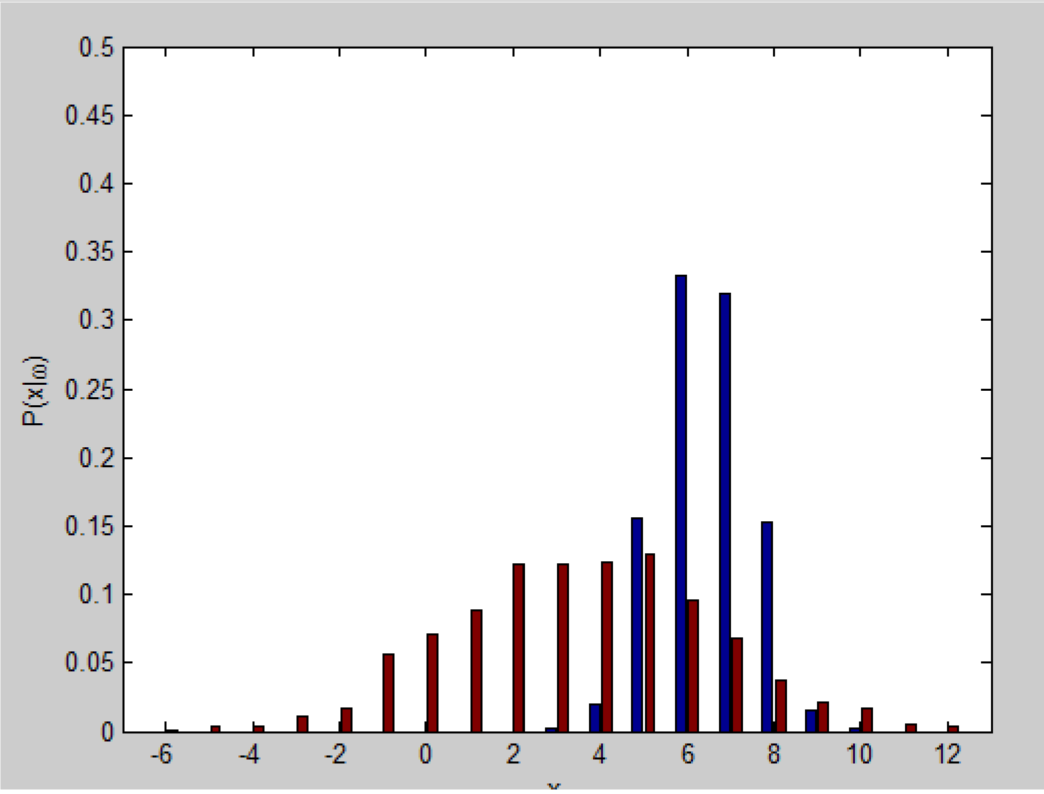
(a)

(i) P(B1=1) = 1/3
(ii) p(O|B1=1) = 1
(iii) $P(B_1=1|O) = P(O|B_1=1)*P(B_1=1)/P(O)$
I think whichever you chose, there must be at least 1 box doesn't contain the bonus, and TA knows which box contains the bonus. So P(O)=1.
So $P(B_1=1|O) = (1/3)*1/1 = 1/3$
(iv) According to Bayesian decision rule, I will have 2/3 probabilities to win the bonus, so I should change.

(b)

(i) likelihood.m

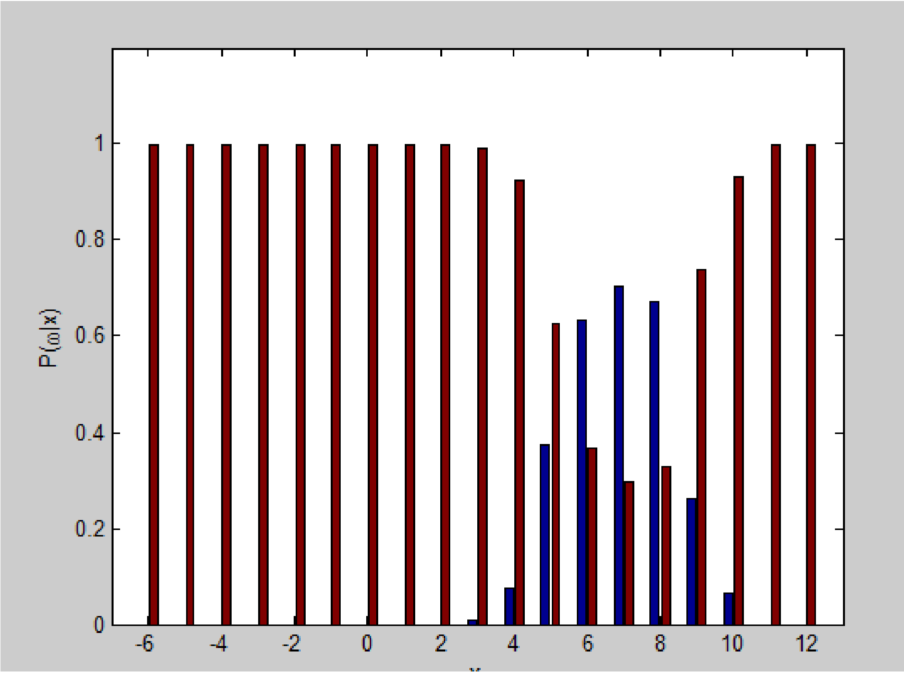
```
function l = likelihood(x)
%LIKELIHOOD Different Class Feature Liklihood
%
%   INPUT:  x, features of different class, C-By-N vector
%           C is the number of classes, N is the number of different
feature
%
%   OUTPUT: l,  likelihood of each feature(from smallest feature to
biggest feature) given by each class, C-By-N matrix
%

[C, N] = size(x);
l = zeros(C, N);
% Your code HERE
n = sum(x,2);
for k=1:C
    l(k,:)=x(k,:)/n(k);
end
end
```



```
mis =

    0.2133
```

(ii)

```
function p = posterior(x)
%POSTERIOR Two Class Posterior Using Bayes Formula
%
%   INPUT:  x, features of different class, C-By-N vector
%          C is the number of classes, N is the number of different
feature
%
%   OUTPUT: p,  posterior of each class given by each feature, C-By-N
matrix
%

[C, N] = size(x);
l = likelihood(x);
total = sum(sum(x));
% Your code HERE
p = zeros(C,N);
px = zeros(1,N);
prior = zeros(2,1);
px = x(1,:)+x(2,:);
px = px./total;
px
prior = [sum(x(1,:)),sum(x(2,:))]./total;
p(1,:) = (l(1,:)./px(1,:))*prior(1);
p(2,:) = (l(2,:)./px(1,:))*prior(2);
end
```

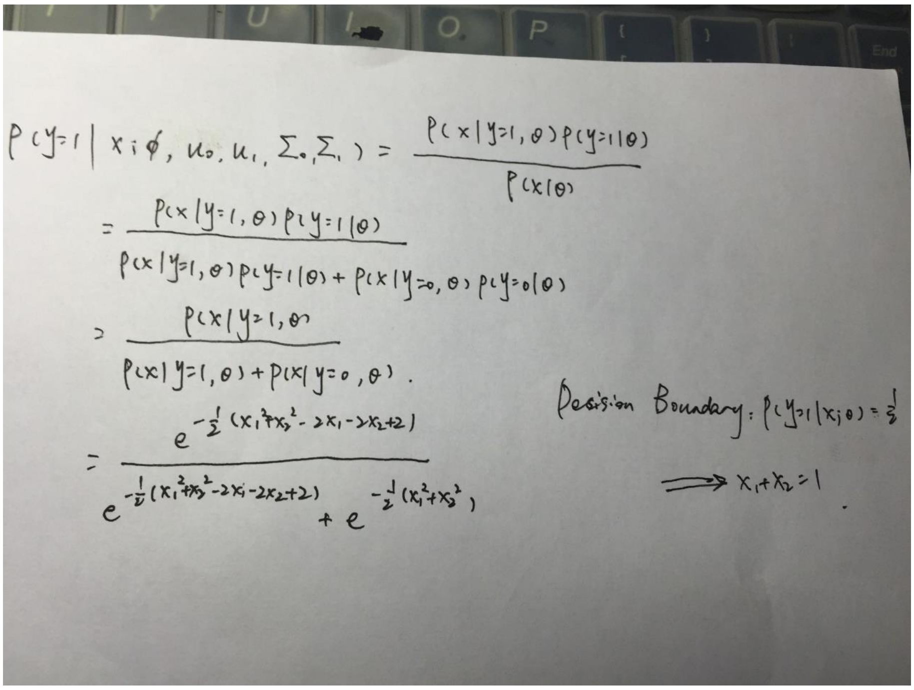



```
mis =

    0.1567

    riskn =

        62
```

(iii)

3. Gaussian Discriminant Analysis and MLE

(a)



$$p(y=1 \mid x; \phi, u_0, u_1, \Sigma_0, \Sigma_1) = \frac{p(x \mid y=1, \theta)\, p(y=1 \mid \theta)}{p(x \mid \theta)}$$

$$= \frac{p(x \mid y=1, \theta)\, p(y=1 \mid \theta)}{p(x \mid y=1, \theta)\, p(y=1 \mid \theta) + p(x \mid y=0, \theta)\, p(y=0 \mid \theta)}$$

$$= \frac{p(x \mid y=1, \theta)}{p(x \mid y=1, \theta) + p(x \mid y=0, \theta)}.$$

$$= \frac{e^{-\frac{1}{2}(x_1^2+x_2^2-2x_1-2x_2+2)}}{e^{-\frac{1}{2}(x_1^2+x_2^2-2x_1-2x_2+2)} + e^{-\frac{1}{2}(x_1^2+x_2^2)}}$$

Decision Boundary: $p(y=1 \mid x; \theta) = \frac{1}{2}$

$\Longrightarrow x_1 + x_2 = 1$

(b)

Gaussian.m

```
function p = gaussian(x, u, sigma)
p = (1/sqrt(det(2*pi*sigma)))*exp(-0.5*(x-u)'*inv(sigma)*(x-u));
end
```

Gaussian_pos_prob.m

```
function p = gaussian_pos_prob(X, Mu, Sigma, Phi)
%GAUSSIAN_POS_PROB Posterior probability of GDA.
%    p = GAUSSIAN_POS_PROB(X, Mu, Sigma) compute the posterior probability
%   of given N data points X using Gaussian Discriminant Analysis where the
```

```
%   K gaussian distributions are specified by Mu, Sigma and Phi.
%
%   Inputs:
%       'X'     - M-by-N matrix, N data points of dimension M.
%       'Mu'    - M-by-K matrix, mean of K Gaussian distributions.
%       'Sigma' - M-by-M-by-K matrix (yes, a 3D matrix), variance matrix
of
%                 K Gaussian distributions.
%       'Phi'   - 1-by-K matrix, prior of K Gaussian distributions.
%
%   Outputs:
%       'p'     - N-by-K matrix, posterior probability of N data points
%                 with in K Gaussian distributions.

N = size(X, 2);
K = length(Phi);
p = zeros(N, K);

% Your code HERE
for xI = 1:N
    px = 0;
    for wI = 1:K
        px = px + gaussian(X(:,xI), Mu(:,wI), Sigma(:,:,wI))*Phi(wI);
    end
    for wI = 1:K
    p(xI,wI)=gaussian(X(:,xI),Mu(:,wI),Sigma(:,:,wI))*Phi(wI)/px;
    end
end
```

(c)run.m

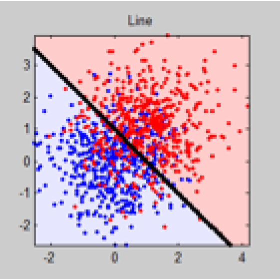
```
% mu: 2x1 matrix
% Sigma: 2x2 matrix
% phi: a number

mu0 = [0; 0];
Sigma0 = [1 0; 0 1];
mu1 = [1; 1];
Sigma1 = [1 0; 0 1];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Line', 1);

mu0 = [0; 1];
```

```
Sigma0 = [1 0; 0 1];
mu1 = [0; 0];
Sigma1 = inv([1 10; -10 1]);
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Line (one side)', 2);

mu0 = [0; 0];
Sigma0 = [1 0; 0 1];
mu1 = [0; 1];
Sigma1 = [2 0; 0 1];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Parabolic', 3);

mu0 = [0; 0];
Sigma0 = [1 0; 0 2];
mu1 = [0; 0];
Sigma1 = [3 0; 0 1];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Hyperbola', 4);

mu0 = [0; 0];
Sigma0 = [1 0; 0 2];
mu1 = [0; 0];
Sigma1 = [2 0; 0 1];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Non-continuous', 5);

mu0 = [0; 0];
Sigma0 = [1 0; 0 1];
mu1 = [0; 0];
Sigma1 = [2 0; 0 2];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Circle', 6);

mu0 = [0; 0];
Sigma0 = [1 0; 0 1];
mu1 = [0; 0];
Sigma1 = [3 0; 0 2];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'Ellipsoid', 7);

mu0 = [0; 0];
Sigma0 = [1 0; 0 1];
mu1 = [0; 0];
```

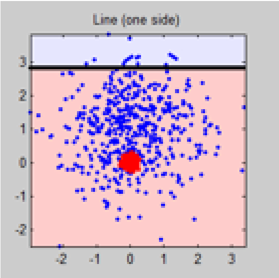
```
Sigma1 = [1 0; 0 1];
phi = 0.5;
plot_ex1(mu0, Sigma0, mu1, Sigma1, phi, 'No boundary', 8);
```

运行结果

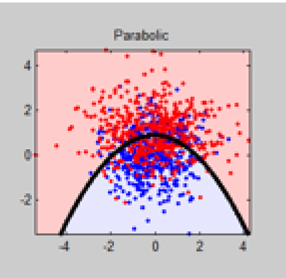
由于运行结果较慢，分开执行并获得结果



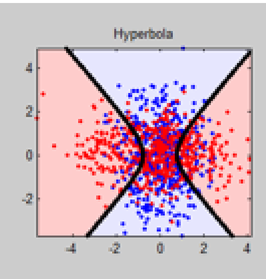
(i)



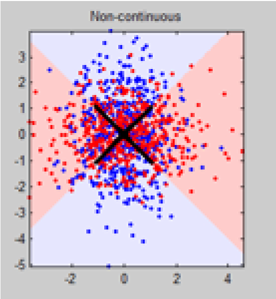
(ii)

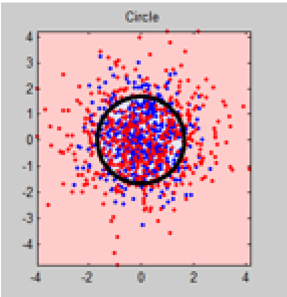


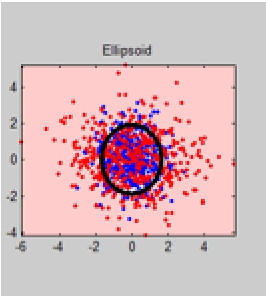
(iii)

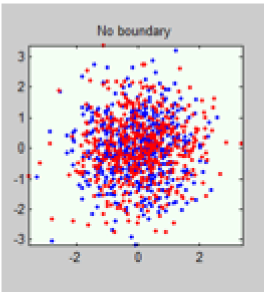


(iv)

(v)


(vi)


(vii)


(viii)


(d)

$$\phi = \frac{\sum_{i=1}^{m} y^{(i)}}{m}, \quad u_0 = \frac{\sum_{i=1}^{m} x^{(i)}(1-y^{(i)})}{\sum_{i=1}^{m} (1-y^{(i)})}, \quad u_1 = \frac{\sum_{i=1}^{m} x^{(i)} y^{(i)}}{\sum_{i=1}^{m} y^{(i)}}.$$

4. Text Classification with Naive Bayes

(i)

```
Top10 =

  Columns 1 through 6

      19957        56930        13613        37568        65398        9494

  Columns 7 through 10

      45153        38176        75526        30033
```

根据all_word_map.txt中的对应关系，前十的词分别为：width，sex，computron，meds，php，voip， cialis， pills， viagra， nbsp

(ii)

```
accuracy =

    0.8496
```

(iii) I think it’s false. Just like the situation in the hint, the filter is expected to have one mistake. If the mistake is to take the ham as spam, the precision will be 50%,and is not a high ratio. But if it mistake the spam as ham,that will be terrible.

(iv)

	Spam(label)	Ham(label)
Spam(predict)	TP	FP
Ham(predict)	FN	TN

```
ans =

   530     28
   594   2983
```

```
precision =

    0.9498


recall =

    0.8521
```

(v) In spam case, I think precision is more important, because we often do not care about spam in daily life. Instead we want useful ham. But in the airport, recall is more important because if the recall is higher, we will find more bombs. And we cannot miss any bomb or drug for safety even if sometimes we might find a wrong package.