Homework1 by Jiabo Cheng

Question 1:

The mean and covariance matrix values given in problem is:

$$\mathbf{m}_0 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad \mathbf{C}_0 = \begin{bmatrix} 2 & -0.5 & 0.3 & 0 \\ -0.5 & 1 & -0.5 & 0 \\ 0.3 & -0.5 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad \mathbf{m}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1 & 0.3 & -0.2 & 0 \\ 0.3 & 2 & 0.3 & 0 \\ -0.2 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

Utilized these distribution PDF and class priors to generate 10000 samples. Since the 4-dimensional data is hard to visualize, the plot of 2-dimension space distribution is in Figure 1 below.

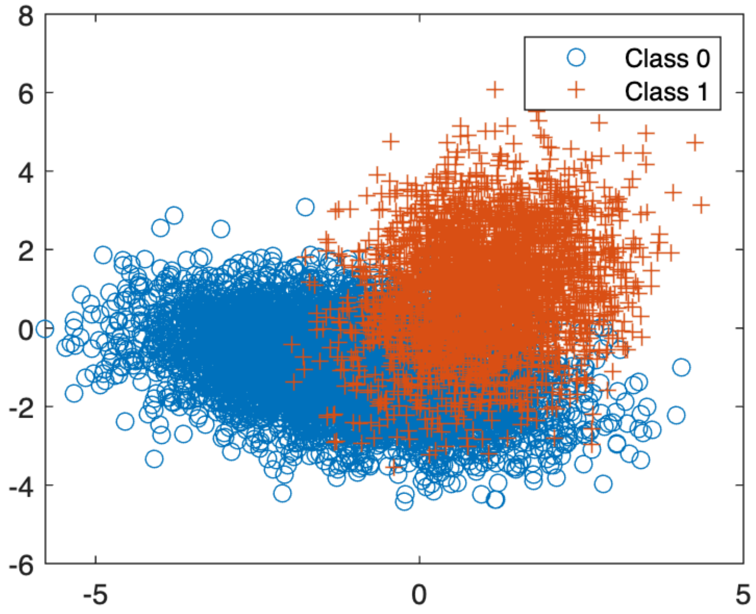


FIGURE 1

Part A:

1. Specify the minimum expected risk classification rule:

The expected risk minimization decision rule is:

$$D_{ERM}(x) = \underset{d \in \{0,1\}}{argmin}\ R(D = d\,|\,x)$$
$$= \underset{d \in \{0,1\}}{argmin} \lambda_{d0} P(L = 0\,|\,x) + \lambda_{d1} P(L = 1\,|\,x)$$

For special case which only has 2 labels, set λ is the cost matrix, then:

$$\lambda_{00}P(L=0|x)+\lambda_{01}P(L=1|x) \underset{<}{>} \lambda_{10}P(L=0|x)+\lambda_{11}P(L=1|x)$$

$$(D=1) \quad \frac{P(x|L=1)}{P(x|L=0)} \underset{<}{>} \frac{(\lambda_{10}-\lambda_{00})P(L=0)}{(\lambda_{01}-\lambda_{11})P(L=1)} \quad (D=0)$$

$$(D=1) \quad \frac{g(x|m_0,C_0)}{g(x|m_1,C_1)} \underset{<}{>} \frac{0.7}{0.3}\frac{(\lambda_{10}-\lambda_{00})}{(\lambda_{01}-\lambda_{11})} \quad (D=0)$$

2. The ROC curve of the minimum expected risk classifier is Figure 2 below.

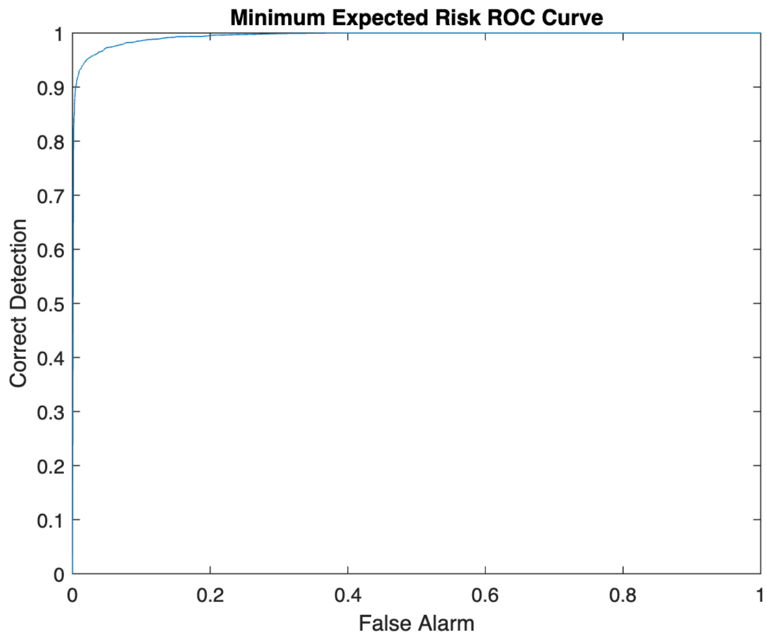


FIGURE 2

3. Calculate the theoretically optimal threshold:

Since the loss values are fixed for each of four cases, $\frac{\lambda_{10}-\lambda_{00}}{\lambda_{01}-\lambda_{11}}=1$, the threshold of $\gamma=\frac{0.7}{0.3}=2.33$, the following table compares the theoretical and the calculated minimum probability of error:

	Threshold	Minimum probability error
Theroetical	2.33	2.7338%
Calculated from data	3.1215	2.6917%

The minimum probability error of calculated from data is closely but a little lower than the theoretical minimum probability error.

The green plus sign in figure 3 represent the true positive and false positive values using theoretical threshold, the red circle sign using the threshold calculated from data.

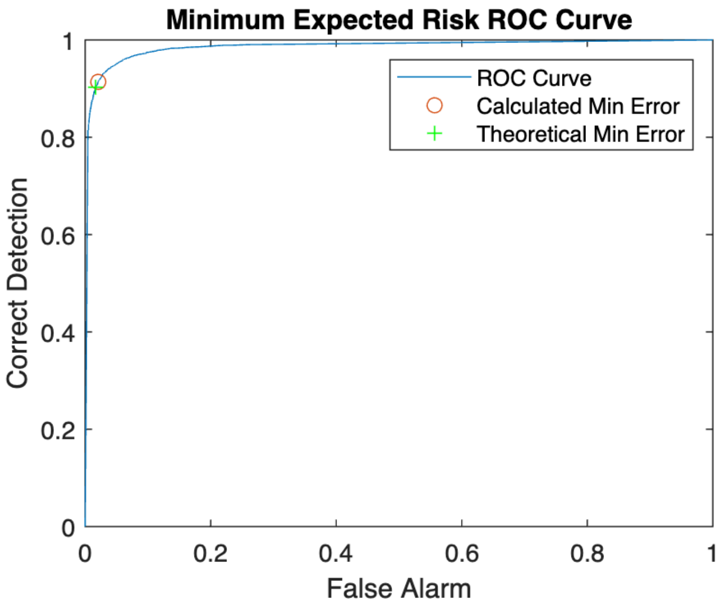


FIGURE 3

Part B:

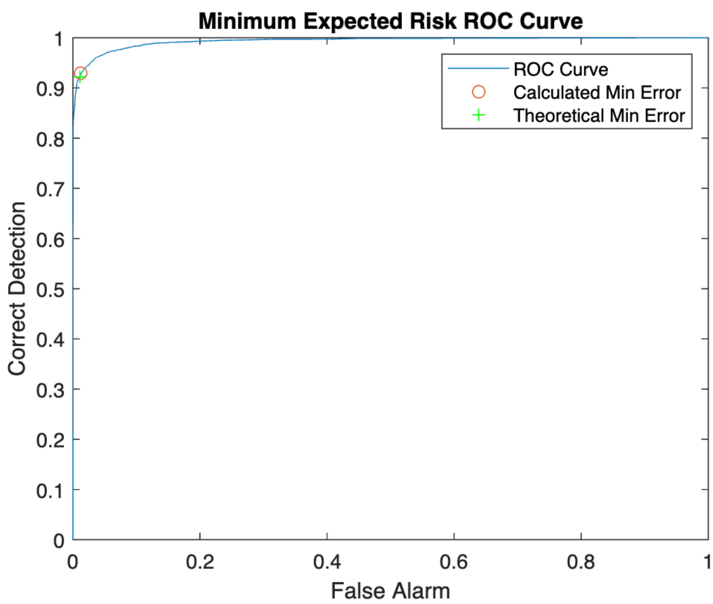
1. Since the covariance matrix used incorrectly is the identical matrix, the expected risk classification rule has changed to:

$$\lambda_{00}P(L=0|x)+\lambda_{01}P(L=1|x) \underset{<}{>} \lambda_{10}P(L=0|x)+\lambda_{11}P(L=1|x)$$

$$(D=1) \quad \frac{P(x|L=1)}{P(x|L=0)} \underset{<}{>} \frac{(\lambda_{10}-\lambda_{00})P(L=0)}{(\lambda_{01}-\lambda_{11})P(L=1)} \quad (D=0)$$

$$(D=1) \quad \frac{g(x|m_0,I)}{g(x|m_1,I)} \underset{<}{>} \frac{0.7}{0.3}\frac{(\lambda_{10}-\lambda_{00})}{(\lambda_{01}-\lambda_{11})} \quad (D=0)$$

2. The ROC curve generated after using Naive-Bayesian classifier is shown as Figure 4 below:



3. The theoretical threshold of $\gamma = \dfrac{0.7}{0.3} = 2.33$, which is the same as the part A.The following table compares the theoretical and the calculated minimum probability of error:

	Threshold	Minimum probability error
Theroetical	2.33	4.3924%
Calculated from data	2.9148	4.2969%

The minimum probability error of calculated from data is closely but a little lower than the theoretical minimum probability error.

Comparing with the result from part A, the minimum probability error is increased, the ROC curve is more 'flat', which means the prediction accuracy is lower than the part A. It make sense because we chose the covariance matrix incorrectly, which has negatively impact on prediction. However, the result is not too bad to classify all the samples wrongly. The reason is that the correct covariance matrix shows there'e no strong linear relationship between variables in both multivariate normal distribution, Naive-Bayes classifier assumes the data is independent, which caused a limited negative impact.

Part C:

Figure 5 below shows the resulting projection of data:

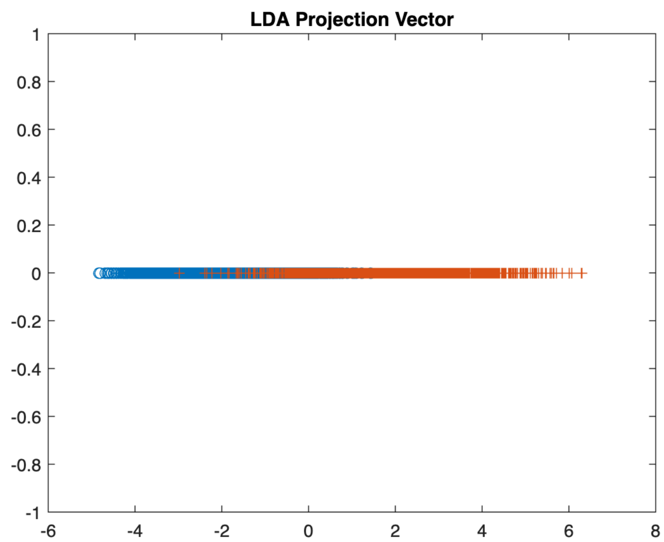


FIGURE 5

The optimal threshold of $\gamma = 0.1286$, with the minimum probability error $p_e = 3.5\,\%$. The performance is better than Naive-Bayes classifier but worse than ERM classifier.

Figure 6 is the ROC curve generated using LDA classifier.

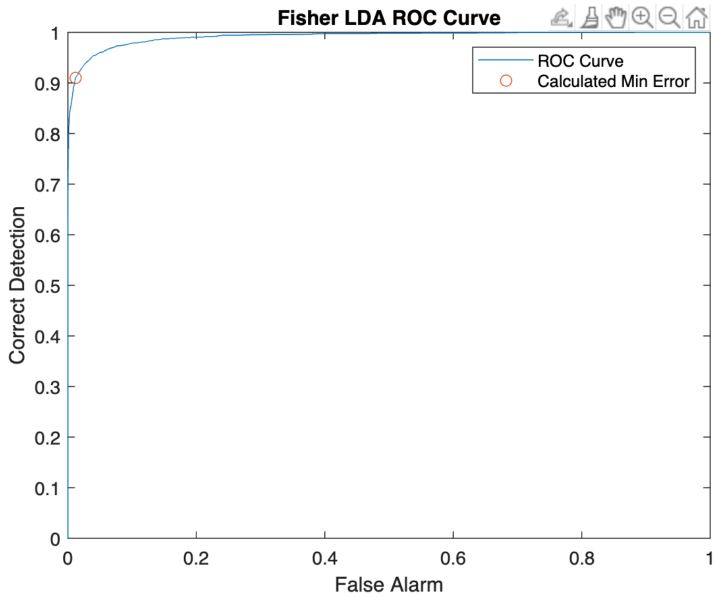


FIGURE 6

Comparing to part A, the result is worse because LDA uses projection of the data distribution, the original 4-dimension distribution is compressed into 1-dimension distribution, lost information caused the worse performance. However, since the original distribution is enough ‘separate’ (comparing distance between mean and covariance), projection on weight vector can also generate good predictions.

Question 2:
Part A:

1. Figure 7 shows 10000 samples from the data distribution that projected on 2-dimension.

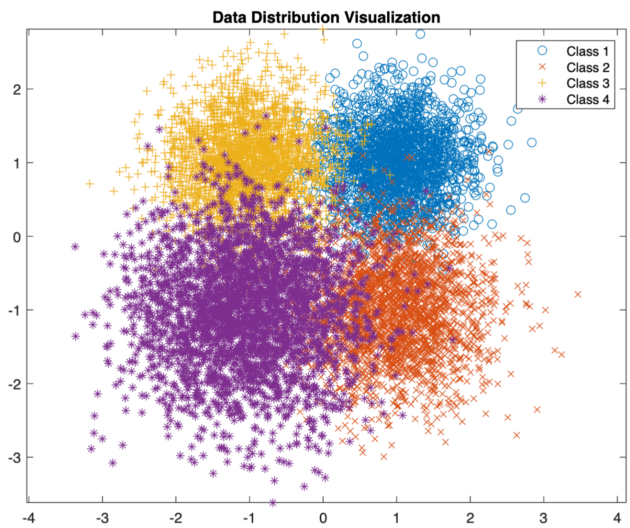


FIGURE 7

2. The decision rule that achieves minimum probability error is:

$$\underset{}{argmin}\begin{matrix} R(D=1|x) \\ R(D=2|x) \\ R(D=3|x) \\ R(D=4|x) \end{matrix} = \Lambda \begin{matrix} P(L=1|x) \\ P(L=2|x) \\ P(L=3|x) \\ P(L=4|x) \end{matrix}$$

Using the 0-1 Loss, the loss matrix $\Lambda = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$, implement this classifier with generated 10000 samples, count the samples corresponding to each decision label, summarized in the confusion matrix below:

Average Minimal Expected Risk			0.1053
Confusion Matrix			
0.9306	0.0335	0.0346	0.0013
0.0358	0.8862	0.0008	0.0771
0.0231	0.0008	0.9165	0.0595
0.0046	0.0687	0.0540	0.8727

3. Figure 8 shows visualization of the data and the prediction result:

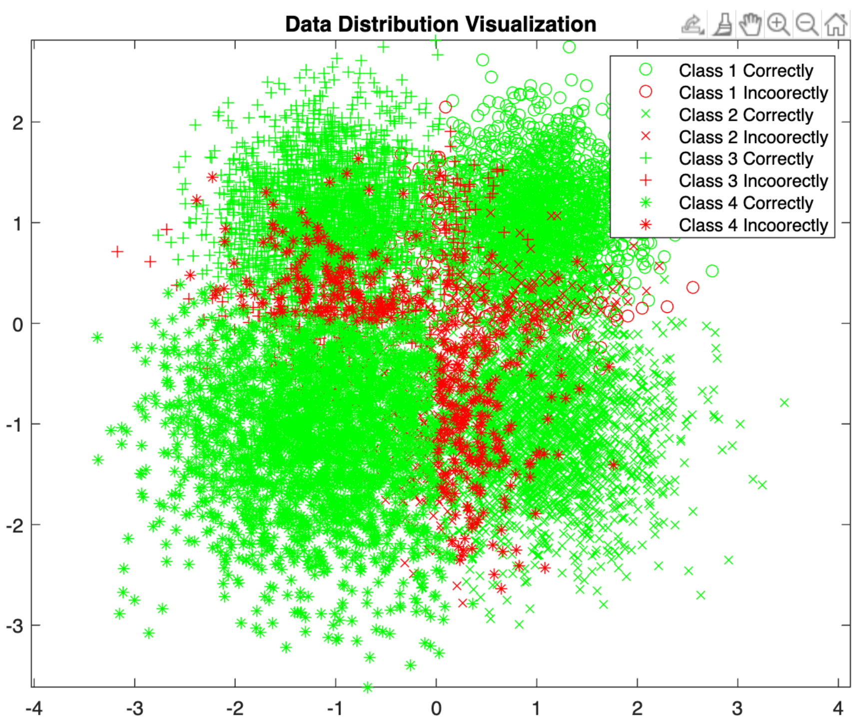


FIGURE 8

Part B

Figure 9 shows visualization of the data and the prediction result:

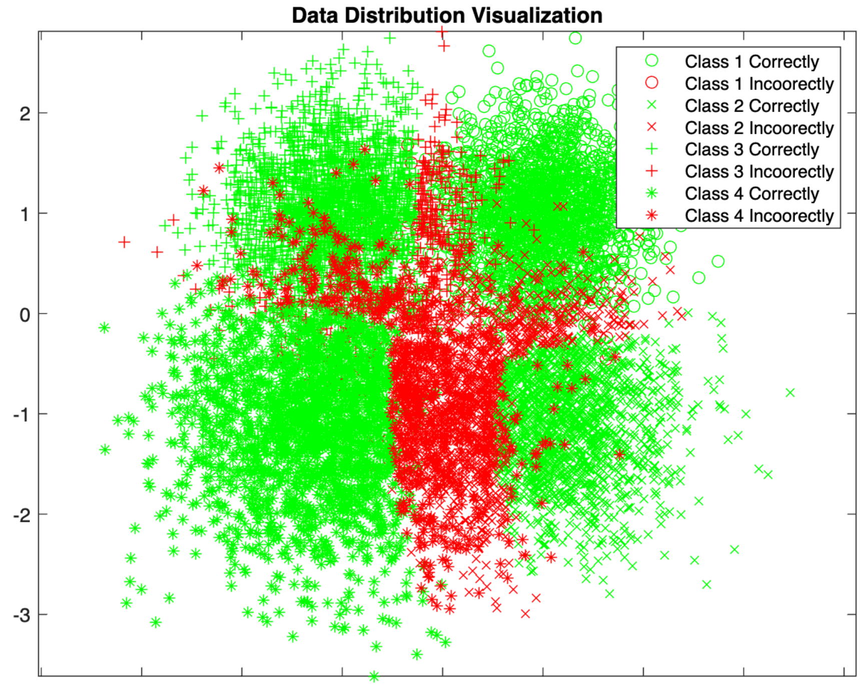


FIGURE 9

The Confusion matrix:

Average Minimal Expected Risk			0.4787
Confusion Matrix			
0.9847	0.03356	0.0957	0.1433
0.0061	0.6433	0	0.0125
0.0092	0.0131	0.8619	0.1244
0	0.007	0.0425	0.7198

The average minimal expected risk is higher in part B than part A, because of the loss values in costs matrix in part B is larger.

Comparing to the cost matrix in part A, we could found the cost of mis-classify class 1 is large(10, 20, 30), while the cost of mis-classify class 3 is tiny(2, 5, 1), which means the prediction accuracy of class 1 is more important than class 3, the result also shows the prediction accuracy in part B(0.9847) is higher than in part A(0.9306), the accuracy of class 3 has dropped from 0.9165 in part A to 0.8619 in part B.

Appendix

Question 1 Code:

```
%%===========Question 1 Setup =============%%
clear all; close all; clc;

n = 4;
N = 10000;

% Class mean and covariance
mu(:,1) = [-1; -1; -1; -1];
Sigma(:,:,1) = [2 -0.5 0.3 0; -0.5 1 -0.5 0; 0.3 -0.5 1 0; 0 0 0 2];
mu(:,2) = [1; 1; 1; 1];
Sigma(:,:,2) = [1 0.3 -0.2 0; 0.3 2 0.3 0; -0.2 0.3 1 0; 0 0 0 3];

%
p = [0.7, 0.3];

label = rand(1, N) >=p(1);
Nc = [sum(label==0), sum(label ==1)];

% Dran samples from each class pdf
x = zeros(n,N);
x(:, label ==0) = mvnrnd(mu(:,1),Sigma(:,:,1), Nc(1))';
x(:, label ==1) = mvnrnd(mu(:,2),Sigma(:,:,2),Nc(2))';

%plot true class labels
figure(1);
plot(x(1,label == 0),x(2,label ==0),'o',x(1, label ==1),x(2,label
==1),'+');
legend('Class 0','Class 1');


%%================ Part 1==============%%
Score = log(eGaussian(x,mu(:,2),Sigma(:,:,2)) ./ eGaussian(x,
mu(:,1),Sigma(:,:,1)) );

tau = log(sort(Score(Score >=0)));
mid_tau = [tau(1)-100 tau(1:end-1) + diff(tau)./2 tau(length(tau))+100];

for i = 1:length(mid_tau)
decision = (Score>=mid_tau(i));
pFA(i) = sum(decision==1 & label== 0)/Nc(1);
pCD(i) = sum(decision==1 & label== 1)/Nc(2);
pE(i) = pFA(i)*p(1)+(1-pCD(i))*p(2);
end

figure(2)
plot(pFA,pCD,'-')
title('Minimum Expected Risk ROC Curve');
xlabel('False Alarm');
ylabel('Correct Detection');

[min_error, min_index] = min(pE);
min_decision = (Score >= mid_tau(min_index));
```

```
min_FA = pFA(min_index );
min_CD = pCD(min_index);

% find the theoretical minimum err
ideal_decision = (Score >= log(p(1)/p(2)));
ideal_pFA = sum(ideal_decision==1 & label==0)/Nc(1);
ideal_pCD = sum(ideal_decision==1 & label==1)/Nc(2);
ideal_error = ideal_pFA*p(1)+(1-ideal_pCD)*p(2);

figure(3)
plot(pFA,pCD,'-',min_FA,min_CD,'o',ideal_pFA,ideal_pCD,'g+');
title('Minimum Expected Risk ROC Curve');
legend('ROC Curve','Calculated Min Error','Theoretical Min Error');
xlabel('False Alarm');
ylabel('Correct Detection');

%resultp
fprintf('Theoretial Result')
ideal_error*100 % Minimum probability of error
p(1)/p(2) % Threshold Value
fprintf('Calculated Result')
min_error*100
exp(mid_tau(min_index))

%==============Part B======================
Sigma_nb =[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

%
Score_nb = log(eGaussian(x,mu(:,2),Sigma_nb) ./ eGaussian(x,
mu(:,1),Sigma_nb) );
tau_nb = log(sort(Score_nb(Score_nb >=0)));
mid_tau_nb = [tau_nb(1)-100 tau_nb(1:end-1) + diff(tau_nb)./2
tau_nb(length(tau_nb))+100];


for i = 1:length(mid_tau_nb)
decision_nb = (Score_nb>mid_tau_nb(i));
pFA_nb(i) = sum(decision_nb==1 & label== 0)/Nc(1);
pCD_nb(i) = sum(decision_nb==1 & label== 1)/Nc(2);
pE_nb(i) = pFA_nb(i)*p(1)+(1-pCD_nb(i))*p(2);
end

%find minimun error
[min_error_nb,min_index_nb] = min(pE_nb);
min_decision_nb = (Score_nb >= mid_tau_nb(min_index_nb));
min_FA_nb = pFA_nb(min_index_nb);
min_CD_nb = pCD_nb(min_index_nb);

% find the theoretical minimum err
ideal_decision_nb = (Score_nb >= log(p(1)/p(2)));
ideal_pFA_nb = sum(ideal_decision_nb==1 & label==0)/Nc(1);
ideal_pCD_nb = sum(ideal_decision_nb==1 & label==1)/Nc(2);
ideal_error_nb = ideal_pFA_nb*p(1)+(1-ideal_pCD_nb)*p(2);

figure(4);
```

```
plot(pFA_nb,pCD_nb,'-',min_FA_nb,min_CD_nb,'o',ideal_pFA_nb,ideal_pCD_nb
,'g+');
title('Minimum Expected Risk ROC Curve');
legend('ROC Curve','Calculated Min Error','Theoretical Min Error');
xlabel('False Alarm');
ylabel('Correct Detection');

%resultp
fprintf('NB Theoretial Result')
ideal_error_nb*100 % Minimum probability of error
p(1)/p(2) % Threshold Value
fprintf('NB Calculated Result')
min_error_nb*100
exp(mid_tau_nb(min_index))

%=================LDA=============
Sb = (mu(:,1)-mu(:,2))*(mu(:,1)-mu(:,2))';
Sw = Sigma(:,:,1)+Sigma(:,:,2);

[V,D ] = eig(inv(Sw)*Sb);
[value,ind]= sort(diag(D),'descend');
wLDA = V(:,ind(1));
yLDA = wLDA'*x;

figure(5);
plot(yLDA(find(label ==0)),zeros(1,Nc(1)),'o',yLDA(find(label
==1)),zeros(1,Nc(2)),'+');
title('LDA Projection Vector');

sorted_yLDA = sort(yLDA);
mid_tau_yLDA = [sorted_yLDA(1)-1
sorted_yLDA(1:end-1)+diff(sorted_yLDA)./2 sorted_yLDA(end)+1];

for i = 1:length(mid_tau_yLDA)
    decision_yLDA = (yLDA >= mid_tau_yLDA(i));
    pFA_yLDA(i) = sum(decision_yLDA==1 & label== 0)/Nc(1);
    pCD_yLDA(i) = sum(decision_yLDA==1 & label== 1)/Nc(2);
    pE_yLDA(i) = pFA_yLDA(i)*p(1)+(1-pCD_yLDA(i))*p(2);

end

[min_error_yLDA,min_index_yLDA] = min(pE_yLDA);
min_decision_yLDA = (Score >= mid_tau_yLDA(min_index_yLDA));
min_FA_yLDA = pFA_yLDA(min_index_yLDA);
min_CD_yLDA = pCD_yLDA(min_index_yLDA);

figure(6)
plot(pFA_yLDA, pCD_yLDA,'-',min_FA_yLDA,min_CD_yLDA,'o')
title('Fisher LDA ROC Curve');
legend('ROC Curve','Calculated Min Error');
xlabel('False Alarm');
ylabel('Correct Detection');

%resultp
fprintf('LDA Result')
min_error_yLDA
```

```
mid_tau_yLDA(min_index_yLDA)

function g = eGaussian(x, mu, Sigma)
% Evaluates the Gaussian pdf N(mu,Sigma) at each coumn of X
[n,N] = size(x);
C = ((2*pi)^n * det(Sigma))^(-1/2);
E = -0.5*sum((x-repmat(mu,1,N)).*(inv(Sigma)*(x-repmat(mu,1,N))),1);
g = C*exp(E);
end
```

Question 2 Code

```
clear all, close all,

C = 4;
N = 10000; % Number of samples
n = 3; % Data dimensionality
p = [0.2 0.25 0.25 0.3]; %Class priors

Mu = [1 1 0; 1 -1 0; -1 1 0; -1 -1 0]';
cov_factor = [0.25 0.4 0.3 0.6]; %factor to change overlap
for i = 1:C
Sigma(:,:,i) = cov_factor(i)*eye(3);
end

% generate data from gmm
u = rand(1,N);
threshold = cumsum(p);
x = zeros(n,N);
label = zeros(1,N);

for i = 1:C
index = find(u<=threshold(i));
Nc = length(index);
label(1,index) = i * ones(1,Nc);
u(1,index) = u(1,index)+2; %these samples will not be selected again
x(:,index) = mvnrnd(Mu(:,i),Sigma(:,:,i),Nc)';
end

% Calculate px
for i = 1:C
pxgiven(i,:) = eGaussian(x,Mu(:,i),Sigma(:,:,i));

end
px = p*pxgiven;
classposteriors = pxgiven.*repmat(p',1,N)./repmat(px,C,1);
%ERM
lossmatrix = ones(C,C)-eye(C);
expectedr = lossmatrix * classposteriors;
[min_expectedr,decisions] = min(expectedr,[],1);
mean(min_expectedr)
```

```
mshapes = 'ox+*.';
mcolors = 'rckmy';

figure(1)
for lab = 1:C
plot(x(1,label==lab),x(2,label==lab),strcat(mshapes(lab))), hold on,
axis equal;

end
title('Data Distribution Visualization');
legend('Class 1','Class 2','Class 3','Class 4');

figure(2)
for lb = 1:C %each label
for d = 1:C % each decision
ind = find(decisions==d & label==lb );

ConfusionMatrix(d,lb) = sum(ind)/sum(find(label==lb));
end

ind1 = find(decisions==lb & label==lb);
plot(x(1,ind1),x(2,ind1),strcat(mshapes(lb),'g')), hold on, axis equal;
    ind2 = find(decisions~=lb & label==lb);
    plot(x(1,ind2),x(2,ind2),strcat(mshapes(lb),'r')), hold on, axis
equal;
end
title('Data Distribution Visualization');
legend('Class 1 Correctly','Class 1 Incoorectly','Class 2
Correctly','Class 2 Incoorectly','Class 3 Correctly','Class 3
Incoorectly','Class 4 Correctly','Class 4 Incoorectly');

ConfusionMatrix


%================PartB=================
lossmatrix = [0 1 2 3;10 0 5 10; 20 10 0 1;30 20 1 0];
expectedr = lossmatrix * classposteriors;
[min_expectedr1,decisions] = min(expectedr,[],1);
mean(min_expectedr1)

figure(3)
for lb = 1:C %each label
    for d = 1:C % each decision
        ind = find(decisions==d & label==lb );

        ConfusionMatrix1(d,lb) = sum(ind)/sum(find(label==lb));
    end

    ind1 = find(decisions==lb & label==lb);
    plot(x(1,ind1),x(2,ind1),strcat(mshapes(lb),'g')), hold on, axis
equal;

    ind2 = find(decisions~=lb & label==lb);
    plot(x(1,ind2),x(2,ind2),strcat(mshapes(lb),'r')), hold on, axis
equal;
```

```
end
title('Data Distribution Visualization');
legend('Class 1 Correctly','Class 1 Incoorectly','Class 2
Correctly','Class 2 Incoorectly','Class 3 Correctly','Class 3
Incoorectly','Class 4 Correctly','Class 4 Incoorectly');

ConfusionMatrix1

function g = eGaussian(x, mu, Sigma)
% Evaluates the Gaussian pdf N(mu,Sigma) at each coumn of X
[n,N] = size(x);
C = ((2*pi)^n * det(Sigma))^(-1/2);
E = -0.5*sum((x-repmat(mu,1,N)).*(inv(Sigma)*(x-repmat(mu,1,N))),1);
g = C*exp(E);

end
```