clc

clear

load('iris.mat')

figure(1);

gscatter(sepalLength, petalLength, class, 'rgb', '^v^', [], 'off');

count\_setosa = sum(class == "Iris-setosa");

count\_versicolor = sum(class == "Iris-versicolor");

count\_virginica = sum(class == "Iris-virginica");

total\_data = count\_setosa + count\_versicolor + count\_virginica;

%prior probabilities

prior = [ count\_setosa/total\_data, count\_versicolor/total\_data, count\_virginica/total\_data];

setosa\_data = [];

versicolor\_data = [];

virginica\_data = [];

%segregating data based on classes

for i = 1 : total\_data

if class(i) == "Iris-setosa"

setosa\_data = [setosa\_data [sepalLength(i); petalLength(i)]];

end

if class(i) == "Iris-versicolor"

versicolor\_data = [versicolor\_data [sepalLength(i); petalLength(i)]];

end

if class(i) == "Iris-virginica"

virginica\_data = [virginica\_data [sepalLength(i); petalLength(i)]];

end

end

%calculating mean and covariance

setosa\_data\_mean = [ mean(setosa\_data(1, :)); mean(setosa\_data(2, :)) ];

versicolor\_data\_mean = [ mean(versicolor\_data(1, :)); mean(versicolor\_data(2, :)) ];

virginica\_data\_mean = [ mean(virginica\_data(1, :)); mean(virginica\_data(2, :)) ];

setosa\_data\_cov = cov(setosa\_data');

versicolor\_data\_cov = cov(versicolor\_data');

virginica\_data\_cov = cov(virginica\_data');

sigma(:,:,1) = setosa\_data\_cov;

sigma(:,:,2) = versicolor\_data\_cov;

sigma(:,:,3) = virginica\_data\_cov;

mu = [setosa\_data\_mean'; versicolor\_data\_mean'; virginica\_data\_mean'];

%liklihood fn

liklihood = @(x,class) mvnpdf(x, mu(class, :), sigma(:,:,class));

%cost matrix

cost = [ 0, 1, 1; 1, 0 ,1; 1, 1, 0];

%expected cost calculation

R1 = @(x) (cost(1,1) \* liklihood(x, 1)\*prior(1)) + (cost(1,2)\*liklihood(x,2)\*prior(2)) + (cost(1,3)\*liklihood(x,3)\*prior(3));

R2 = @(x) (cost(2,1)\*liklihood(x, 1)\*prior(1)) + (cost(2,2)\*liklihood(x,2)\*prior(2)) + (cost(2,3)\*liklihood(x,3)\*prior(3));

R3 = @(x) (cost(3,1)\*liklihood(x, 1)\*prior(1)) + (cost(3,2)\*liklihood(x,2)\*prior(2)) + (cost(3,3)\*liklihood(x,3)\*prior(3));

%Decision rule

classfy = @(x) min([R1(x),R2(x) ,R3(x) ]);

%finding the regions by brute force

figure(2);

for h = 4:0.1:9

for k = 0:0.1:8

[expected\_min\_risk, label] = classfy([h, k]);

hold on;

if(label == 1)

scatter(h,k, 50,[0.9,0.9, 0.9],'filled');

end

if(label == 2)

scatter(h,k, 50,[0.4,0.4, 0.4],'filled');

end

if(label == 3)

scatter(h,k, 50,[0.7,0.7, 0.7],'filled');

end

hold off;

end

end

hold on;

gscatter(sepalLength, petalLength, class, 'rgb', '^v^', [], 'off');

%axis equal;

hold off;