Pattern Recognition Lab

CSE 4214

Implementing Minimum Error Rate Classifier

28 July 2018

Submitted By:

Md. Mukitul Islam 14.02.04.076

Implementing Minimum Error Rate Classifier

Md. Mukitul Islam (*140204076*)
Department of Computer Science and Engineering,
Ahsanullah University of Science and Technology.

***Objectives*—the objective of this experiment is to classify some sample points using the posterior probabilities which uses Gaussian distribution to calculate the likelihood probabilities. The objective of this type of classifier is to minimize the error rate during classification. So this classifier takes decision based on the most posterior probabilities. This classifier is also known as Bayes classifier with minimum error.**

***Keywords*—discriminant functions; pattern recognition; liklihood probabilities ratio; MATLAB code; Bayesian classifier;**

I. INTRODUCTION

Minimum error rate classifier is a classifier and its objective is to minimize the error rate. In this experiment we are given six sample data, we have to classify those. The likelihood probabilities of a sample is given by the normal distribution. Any normal distribution can be express with two parameter- sigma and mean. In this experiment these parameters are given. As Bayesian classifier works with posterior probabilities the decision rule is as follows-

If p(w1 | x) > p(w2 | x) then x € w1
If p(w1 | x) < p(w2 | x) then x € w2

The posterior probabilities can be calculated with the help of likelihood probabilities. This can be written as-

$P(w_i \mid x) = P(x \mid w_i)\, P(w_i)$
$= \text{Ln } P(w_i \mid x) = \text{Ln } P(x \mid w_i)\, P(w_i)$
$= \text{Ln } P(x \mid w_i) + \text{Ln } P(w_i)$ (i)

Now the likelihood probabilities is the Gaussian distribution here and in 1D Gaussian distribution can be written as-

$N(x) = 1/ \sqrt{2}\prod \partial \text{ x } e^{-1/2(x-\check{\ })\hat{\ }2/\partial\hat{\ }2}$

Since here the data is 2D so we have to use the following equation-

$N(x) = 1/ (2\prod)^{d/2} |\sum|^{½} \text{ x } e^{-1/2\,(x-\check{\ })t \sum -1\, (x-\check{\ })}$

So $g_i(x) = w^t x + w_0 = \text{Ln } P(x \mid w_i) + \text{Ln } P(w_i)$
$= -d/2 \text{ Ln } 2\prod - 1/2 \text{ Ln } | \sum | - 1/2\, (x - \check{\ })^t \sum^{-1} (x - \check{\ }) + \text{Ln } p(w_i)$

Here $\sum$ and ˇ are class specific and d=2 for our experiment because all the data are 2D.

II. IMPLEMENTATION

A. Plotting the sample points with different markers

We simply calculate the value of g(x) for each sample points with the two given Gaussian distributions and check for the following conditions g1(x) > g2(x). If the above condition is true then the sample point, x belongs to the corresponding regions of the gaussian distributions.

The value of g1(x) greater than g2(x) means the sample point likelihood probabilities close to the used gaussian distribution so we can assign this sample point to that region.

After plotting all the sample values the output is as follows-

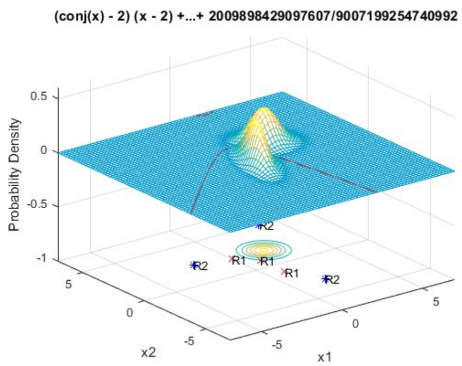

Figure 1: plotting the given sample points

Now that we plot all the sample to their expected class now we have to draw a decision boundary to divide the whole space in two region.

B. Drawing decision boundary

To draw the decision boundary we have to obtain the equation of the decision boundary and we know that at decision boundary g1(x)-g2(x) = 0 so we use symbolic variable in the code for making the equation and use a build in function in MATLAB called ezplot to draw the boundary. To see the decision boundary more clearly we can rotate the graph and see it clearly in the following figure-

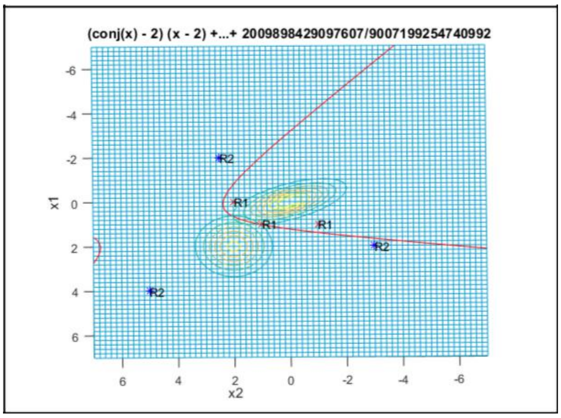

Figure 2: Drawing the decision boundary

Now we can see the decision boundary more clearly in the above figure.

After drawing the decision boundary we have to change the parameter value of Gaussian distribution and we have to observer that what changes. We will discuss it in the Result analysis section.

III. RESULT ANALYSIS

As we know that minimum error rate classifier tries to minimize the error. So if we change the parameter of Gaussian distribution the sample values also can be shifted to another class because the likelihood probabilities can also be shifted towards another class.

For example if we change the mean of first distribution then it will simply shifted to another place in the graph. Thus now all the sample points will shift as needed. The output is given below –

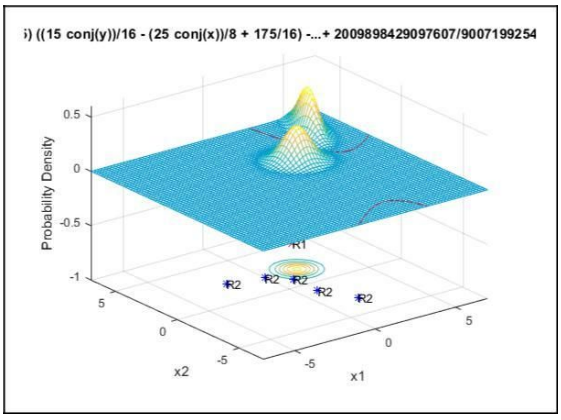

Figure 3: changing the mean of first distribution

class 1. Now different output for different parameter values are given below –

Changing Sigma values of first distribution mu1 = [5 5];

Sigma1 = [.1 .8;.8 9];

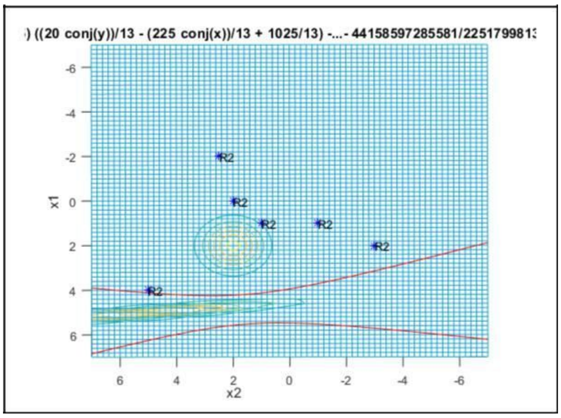

Figure 4: Different outputs

Changing the Sigma and mean values of second distribution

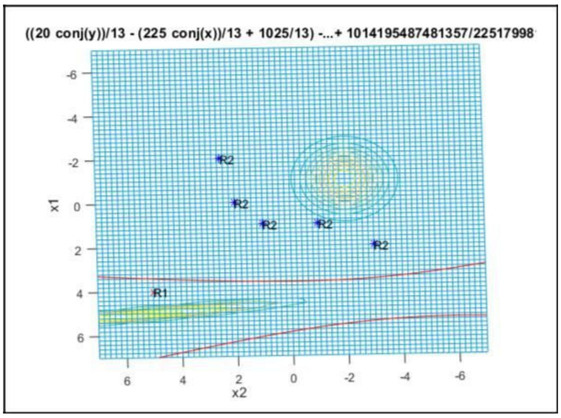

Figure 5: Different outputs

From the above figure we can see that now 5 sample values are classifies as class 2 and only 1 sample value is classified as

mu2 = [-1 -2];
Sigma2 = [.8 0;0 .8];

From the above figures we observed that sigma denotes the Gaussian distribution scatterness and and meand is the mean of all points in the Gaussian distribution. And here the error rate is zero in every case. So our design to minimum error rate classification is accomplished.

MATLAB CODE:

```
x1 = -7:0.2:7; x2 =
-7:0.2:7; [X1,X2] =
meshgrid(x1,x2);

mu1 = [5 5];

Sigma1 = [.1 .8;.8 9];

F1 = mvnpdf([X1(:)
X2(:)],mu1,Sigma1); %multi variate
normal probability density function

%returns the density of the
multivariate normal distribution

%with mean mu1 and covariance
Sigma1,

%evaluated at each row of X1 X2

F1 =

reshape(F1,length(x2),length(x1));%r
eshap es F1 into a length(x2) by
length(x1) array

%

where length(x2) and length(x1)
indicates

the size of each dimension.
%surfc(x1,x2,F1);
meshc(X1,X2,F1);

%draws a wireframe mesh and a
contour plot under it

%with color determined by F1
                                     %A

contour plot displays isolines of

%matrix F1

axis([-7 7 -7 7 -1.0
0.6]); %sets the axis
limit

xlabel('x1');
ylabel('x2');
zlabel('Probability
Density');

hold on;
mu2 = [-1 -2];

Sigma2 = [.8 0;0 .8];
F2 = mvnpdf([X1(:)
X2(:)],mu2,Sigma2);

F2 =
reshape(F2,length(x2),length(x1));
%surfc(x1,x2,F2);
meshc(X1,X2,F2);

axis([-7 7 -7 7 -1.0
0.6]); xlabel('x1');
ylabel('x2');
zlabel('Probability
Density'); hold on;

%assigning color to the axis
caxis([min(F2(:))-.5*ra
nge(F2(:)),max(F2(:))])
;

% Write Your CODE here
samples=[1 1;1 -1;4 5;-2 2.5;0 2;2
-3];

for n=1:6

for m=1:2
if(m==1)
g1=-log(2*pi)-

0.5*log(det(Sigma1))-0.5*(samples(
n,:)'-mu1')'*inv(Sigma1)*(samples(
n,:)'-mu1')+log(0.5);
elseif(m==2)
g2=-log(2*pi)-

0.5*log(det(Sigma2))-0.5*(samples(
n,:)'-mu2')'*inv(Sigma2)*(samples(
n,:)'-mu2')+log(0.5);
end
end

if(g1>g2)
plot3(samples(n,1),samples(n,2),-
1.0,'rx');
text(samples(n,1),samples(n,2),-
1.0,'R1');
else

        plot3(samples(n,1),samples(
n,2),-1.0,'b*');
```

```
        text(samples(n,1),samples(
n,2),-1.0,'R2' );

end

end

syms x y;

eq1
=-log(2*pi)-0.5*log(det(Sigma1)
)-0.5*([x;y]-mu1')'*inv(Sigma1)
*([x;y]-mu1')+log(0.5);

eq2
=-log(2*pi)-0.5*log(det(Sigma2)
)-0.5*([x;y]-mu2')'*inv(Sigma2)
*([x;y]-mu2')+log(0.5);
eq = eq1 - eq2;    %generating the

equation

con=ezplot(eq, [[-7,7],[
-7,7]]);
set(con,'Color','red');
xlabel('x1');
ylabel('x2');
zlabel('Probability
Density');

hold off;
```

IV. CONCLUSION

In this experiment we came to know that how a minimum error rate classifier works and what does it mean by sigma and mean of Gaussian distribution. However there are some limitations of this classifier. This classifier fully depends on probability and the Gaussian distribution should be known.