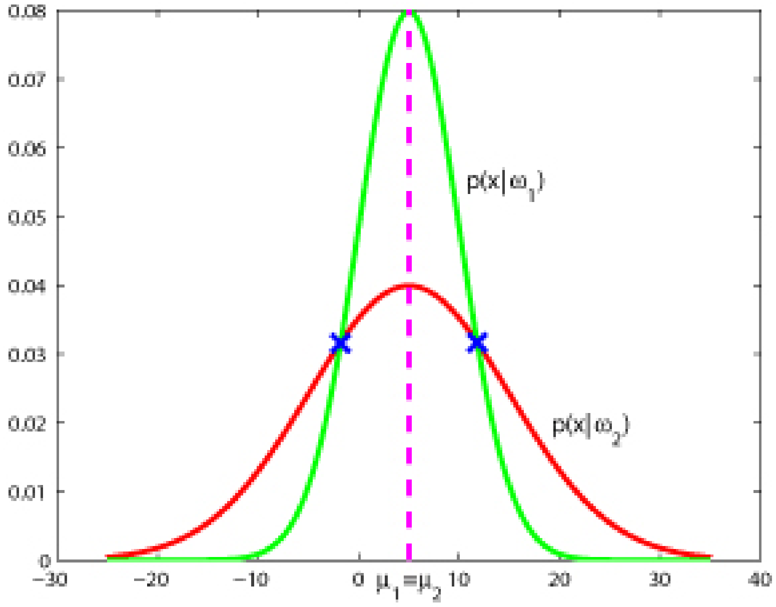
ECE 662 : Pattern Recognition and Decision Making Processes

Mini-Project 1

Debasmit Das
das35@purdue.edu

February 12, 2016



Bayesian Decision Rule
Courtesy : Project Rhea

Contents

1 Introduction **2**

2 Experiment and Results **3**
2.1 Effect of distance between the Mean of Gaussians on Error Rate . 3
2.1.1 Case 1 3
2.1.2 Case 2 3
2.1.3 Case 3 3
2.1.4 Case 4 3
2.1.5 Comments 6
2.2 Effect of Number of Sample Points on Error Rate 6
2.2.1 Case 1 6
2.2.2 Case 2 6
2.2.3 Case 3 6
2.2.4 Case 4 6
2.2.5 Comments 6
2.3 Effect of Dimension on Error Rate 9
2.3.1 Case 1 9
2.3.2 Case 2 9
2.3.3 Comments 10

3 Conclusion **10**

4 Code **11**
4.1 Function to randomly generate data from 2-Component Gaussian Mixture 11
4.2 Code to return Bayes Error using Monte Carlo 12
4.3 Empirical Error using Discriminant Function 13
4.4 Effect of Factors on Error Rate 14

1 Introduction

This project is about finding how the **error rate** of our classifier varies with changing factors. The classifier in our case is a discriminant function which we will be introducing shortly. We will also be comparing our discriminant function against Bayes decision rule and see how our error rate trends with the Bayes error. Before we go any further we want to discuss the Bayesian Decision Rule.

Consider a model where there are a number of classes. For our case, we consider the trivial example of 2 classes w_1 and w_2. Now, we assume that data is generated from these models. The probability that the class w_i can generate the data is given by $P(w_i)$ and $i = 1, 2$. This probability is called the **prior**. Now, given the fact that the data is from a particular class, the probability of occurrence of the data depends on the model we choose. For our case, the distribution of data in each class is Gaussian. Mathematically, it can be written as $P(x \mid w_i) = \mathcal{N}(\mu_i, \Sigma_i)$. The term $P(x|w_i)$ is known as **likelihood**. Now, from Bayes Theorem, we know that the **posterior** is proportional to the product of the likelihood and the prior. Mathematically, $P(w_i \mid x) \propto P(x \mid w_i)P(w_i)$ for $i = 1, 2$. After finding the posterior, we compare the posterior probability of the two classes. If $P(w_1 \mid x) > P(w_2 \mid x)$ then we assign point x to class 1 else if $P(w_1 \mid x) < P(w_2 \mid x)$ then we assign x to class 2. This how Bayes Decision Rule for two classes works and is explained well in Chapter 3 of [2].

Now, the question is how we should assign a performance metric to this classification rule. Error rate is a good idea. But there should be a theoretical measure rather than an empirical one. So we define Bayes Error as the probability of misclassification in the model. This can be visualized in the figure given in Fig. 3.1 of [1] .

The Bayes error is difficult to compute and therefore a sampling technique is used to compute it. The Monte Carlo Sampling technique has been used to find the Bayes Error with number of samples around a million. The implementation has been given in the code section. We also want to compare our classifier with the Bayesian one. The classifier we use is a discriminant function. Before describing the discriminant function we would like to look at the parameters of our model. So we have two classes represented by 2 Gaussians. Each Gaussian i has parameters $\boldsymbol{\mu_i} \in \mathbb{R}^d$ (mean), $\boldsymbol{\Sigma_i} \in \mathbb{R}^{d\times d}$ (covariance), and $p_i \in \mathbb{R}$ (prior) for $i = 1, 2$. Given a d dimensional data $\mathbf{x} \in \mathbb{R}^d$ then our discriminant $g(x)$ is given by -

$$g(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_1})^T\boldsymbol{\Sigma_1}^{-1}(\mathbf{x}-\boldsymbol{\mu_1}) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_2})^T\boldsymbol{\Sigma_2}^{-1}(\mathbf{x}-\boldsymbol{\mu_2}) + \frac{1}{2}\log(\frac{|\boldsymbol{\Sigma_1}|}{|\boldsymbol{\Sigma_2}|}) - \log(\frac{p_1}{p_2}) \tag{1}$$

The classification rule is that if $g(\mathbf{x}) > 0$ then $\mathbf{x}$ should belong to Class 2 otherwise Class 1. Now how do we compute the error of our classifier ? The following steps need to be followed -

- Generate N number of samples from the 2-component Gaussian Mixture Distribution along with their ground true class labels.

- Apply the discriminant rule to the N samples and find the number of misclassified (M) examples. The error rate should be $e = M/N$. This is called the empirical error rate.
- Repeat the above experiment around a 100 times to get the the average error $\bar{e}$ and also find its variance. These are plotted on the graph with respect to some parameters that we are changing (We will get to the parameters later)
- We have to plot the $\bar{e}$ by varying the parameter. We also have to plot the Bayes Error by changing the parameter of interest and then compare and analyze.

Now, let us talk about the parameters that we are changing in the experimental section

2 Experiment and Results

Here we will describe the experiments. We will visualize the effect of various factors on the error rate of our classifier. We will also look at how it trends with Bayes error. **Priors** of both classes are set to **0.5**

2.1 Effect of distance between the Mean of Gaussians on Error Rate

d, the variable parameter is the distance between the means. $\mathbf{M}(a, b, c)$ is a $c \times c$ matrix with a as the diagonal elements and b are rest of the elements of the matrix. $N = 1000$ is the number of samples used for calculating the empirical error. Number of Monte Carlo samples =1000 for plotting the Bayes Error.

2.1.1 Case 1

2-D Data | $\boldsymbol{\mu}_1 = [0, 0]^T$ | $\boldsymbol{\mu}_2 = [d, 0]^T$ | $\boldsymbol{\Sigma}_1 = [1, 0.5; 0.5, 1]^T$ | $\boldsymbol{\Sigma}_2 = [1, 0.5; 0.5, 1]^T$

2.1.2 Case 2

2-D Data | $\boldsymbol{\mu}_1 = [0, 0]^T$ | $\boldsymbol{\mu}_2 = [d, 0]^T$ | $\boldsymbol{\Sigma}_1 = [1, 0.5; 0.5, 1]^T$ | $\boldsymbol{\Sigma}_2 = [1, 0; 0, 1]^T$

2.1.3 Case 3

10-D Data | $\boldsymbol{\mu}_1 = \mathbf{0}_{10\times 1}$ | $\boldsymbol{\mu}_2 = d\mathbf{I}_{10\times 1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{M}(1, 0.5, 10)$ | $\boldsymbol{\Sigma}_2 = \mathbf{M}(1, 0.5, 10)$

2.1.4 Case 4

10-D Data | $\boldsymbol{\mu}_1 = \mathbf{0}_{10\times 1}$ | $\boldsymbol{\mu}_2 = d\mathbf{I}_{10\times 1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{M}(1, 0.5, 10)$ | $\boldsymbol{\Sigma}_2 = \mathbf{I}_{10\times 10}$

Figure 1: Effect of distance between the Mean of Gaussians on Error Rate (Case 1)

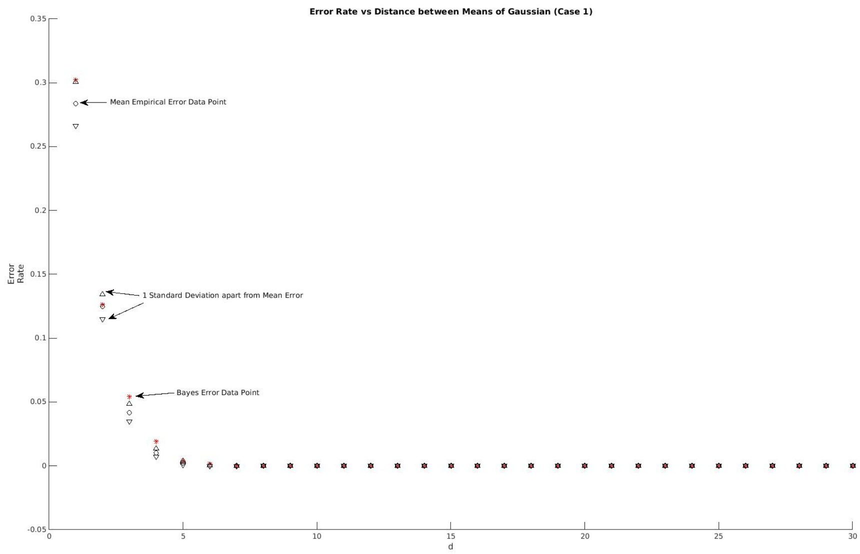


Figure 2: Effect of distance between the Mean of Gaussians on Error Rate (Case 2)

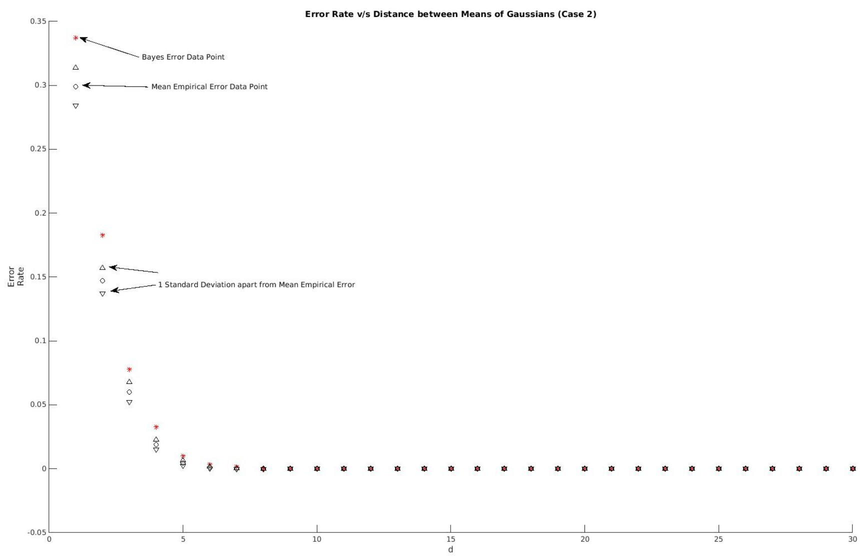


Figure 3: Effect of distance between the Mean of Gaussians on Error Rate (Case 3)

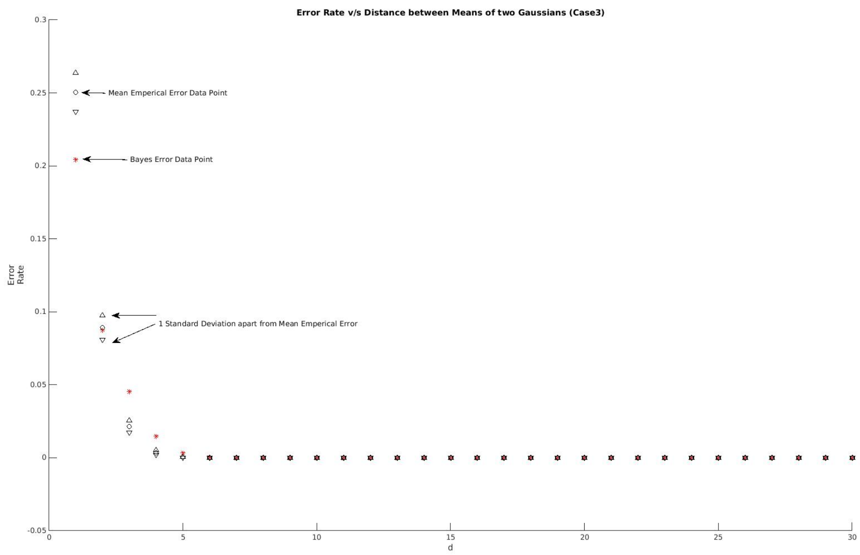
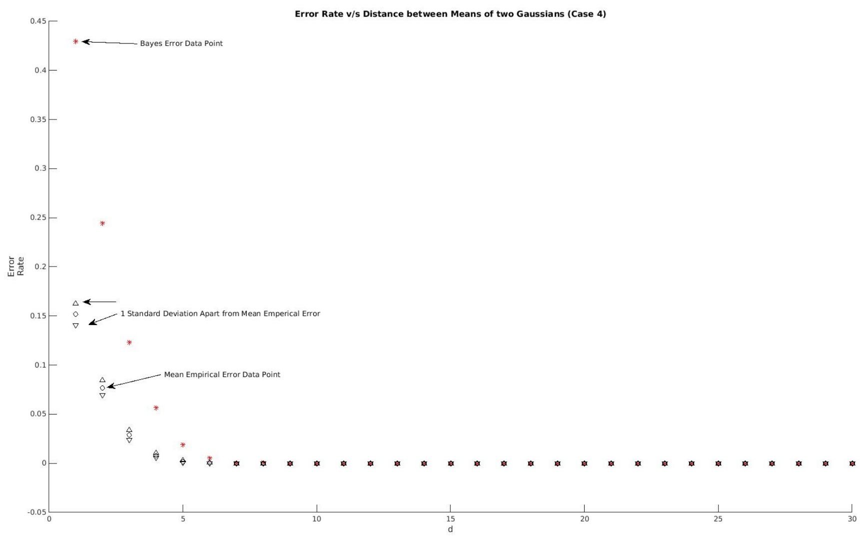


Figure 4: Effect of distance between the Mean of Gaussians on Error Rate (Case 4)



2.1.5 Comments

As expected we see that the Bayes error decreases with the increase in distance between the means. This is mainly because the overlap region between the 2 gaussians decrease as the gaussians go far apart. The overlap region represents the probability of mis-classification. This is similar for our discriminant function which simulates the Bayes Decision Rule except the fact that it represents empirical error rather than the theoritical error. As we increase dimensionality to 10 the difference between the empirical error and the theoritical error increases because the sample points remain the same and therefore the model underfits and the difference is magnified. When the covariance matrix are equal the decision boundary is a hyperplane and when the covariance matrix are not equal the decision boundary is highly non-linear and it may overfit the data. As a result error rate may increase and gap between Bayes Error and Empirical Error is increased as is seen between Fig. 1 and Fig. 2 and also between Fig. 3 and Fig. 4

2.2 Effect of Number of Sample Points on Error Rate

N, the variable parameter is the number of samples when calculating empirical error. $\mathbf{M}(a, b, c)$ is a $c \times c$ matrix with a as the diagonal elements and b are rest of the elements of the matrix. $d = 1$ is the distance between the means. Number of Monte Carlo samples =1000 for plotting the Bayes Error. The number of monte-carlo samples is kept constant.

2.2.1 Case 1

2-D Data | $\boldsymbol{\mu}_1 = [0, 0]^T$ | $\boldsymbol{\mu}_2 = [1, 0]^T$ | $\boldsymbol{\Sigma}_1 = [1, 0.5; 0.5, 1]^T$ | $\boldsymbol{\Sigma}_2 = [1, 0.5; 0.5, 1]^T$

2.2.2 Case 2

2-D Data | $\boldsymbol{\mu}_1 = [0, 0]^T$ | $\boldsymbol{\mu}_2 = [d, 0]^T$ | $\boldsymbol{\Sigma}_1 = [1, 0.5; 0.5, 1]^T$ | $\boldsymbol{\Sigma}_2 = [1, 0; 0, 1]^T$

2.2.3 Case 3

10-D Data | $\boldsymbol{\mu}_1 = \mathbf{0}_{10\times1}$ | $\boldsymbol{\mu}_2 = d\mathbf{I}_{10\times1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{M}(1, 0.5, 10)$ | $\boldsymbol{\Sigma}_2 = \mathbf{M}(1, 0.5, 10)$

2.2.4 Case 4

10-D Data | $\boldsymbol{\mu}_1 = \mathbf{0}_{10\times1}$ | $\boldsymbol{\mu}_2 = d\mathbf{I}_{10\times1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{M}(1, 0.5, 10)$ | $\boldsymbol{\Sigma}_2 = \mathbf{I}_{10\times10}$

2.2.5 Comments

Here the number of classification examples or the number of sample points for empirical error calculation is varied. It is seen that the mean empirical error rate is more or less same except the fact that its variance increases with the number of sample points. The reason may be that for very less number of sample

Figure 5: Effect of Number of Classification Points on Error Rate (Case 1)

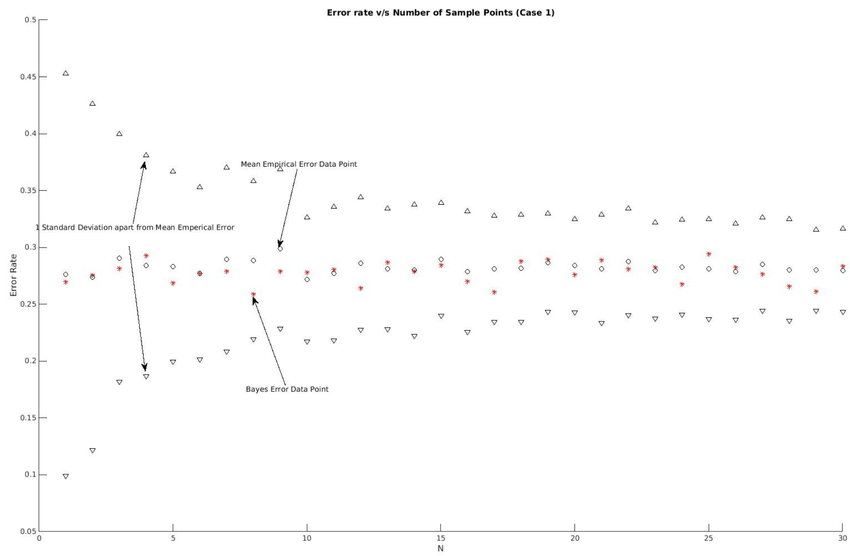


Figure 6: Effect of Number of Sample Points on Error Rate (Case 2)

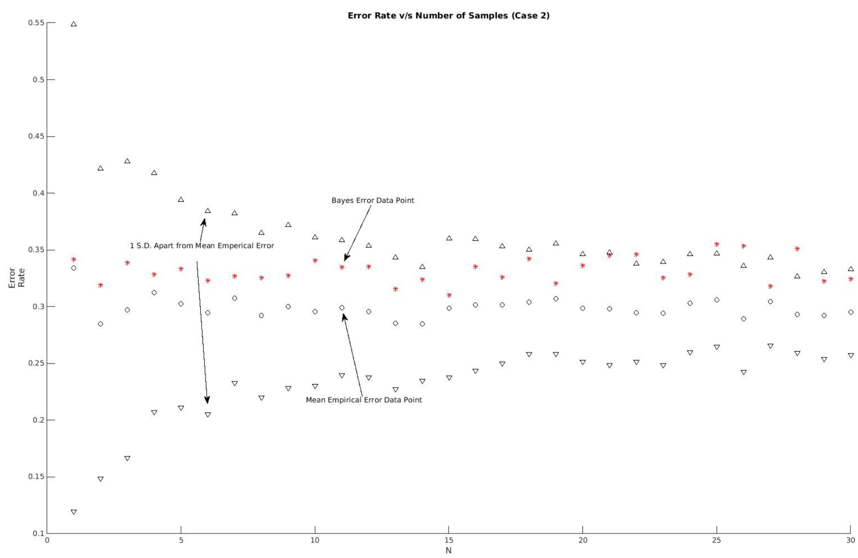


Figure 7: Effect of Number of Sample Points on Error Rate (Case 3)

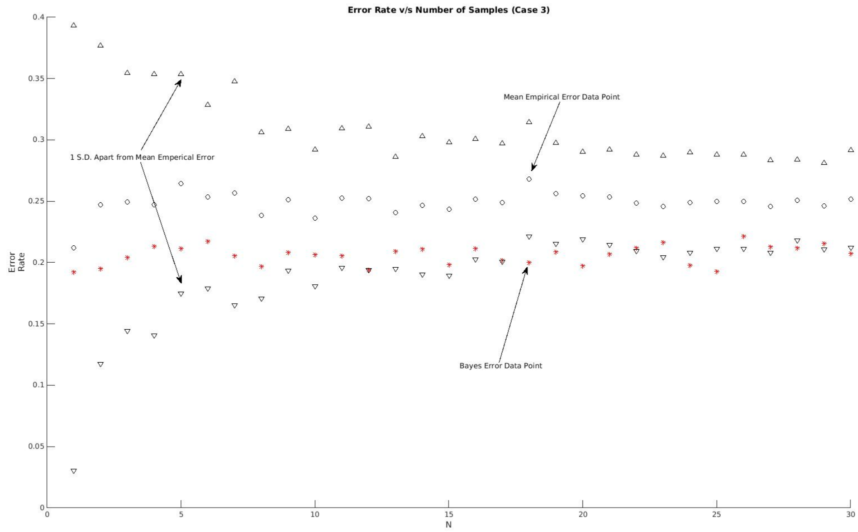
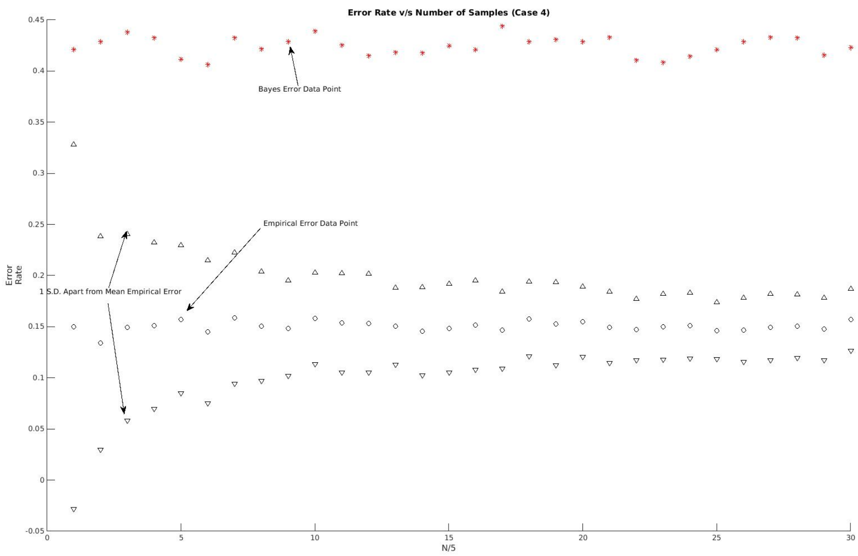


Figure 8: Effect of Number of Sample Points on Error Rate (Case 4)



points the under-fitting phenomenon take place and there is high variance and less bias. Moreover, we see sometimes as in Fig.8 that the Bayes Error becomes more than Empirical one. This is mainly because of curse of dimensionality. As dimension increases the number of monte-carlo samples need to increase as well so the approximation for Bayes Error is not that good. For, infinite monte-carlo samples the Bayes error would always have lied below the empirical error.

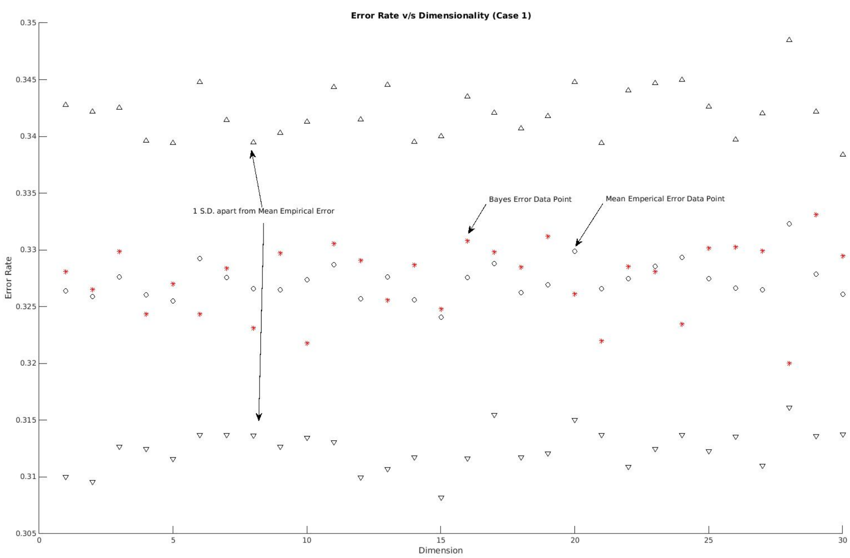
2.3 Effect of Dimension on Error Rate

n, the variable parameter is the dimension of data. Number of Monte Carlo samples =10000 for plotting the Bayes Error. The number of monte-carlo samples is kept constant. The number of classification points is kept at 1000. Two different cases are analysed one when covariance matrix are equal and when they are not equal. Care is taken as to keep similar pattern between means and covariances as dimension is varied.

2.3.1 Case 1

Unequal Covariances | $\boldsymbol{\mu}_1 = \mathbf{0}_{n\times n}$ | $\boldsymbol{\mu}_2 = \mathbf{I}_{n\times 1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{I}_{n\times n}$ | $\boldsymbol{\Sigma}_2 = diag(2, 1, 1, ...)$

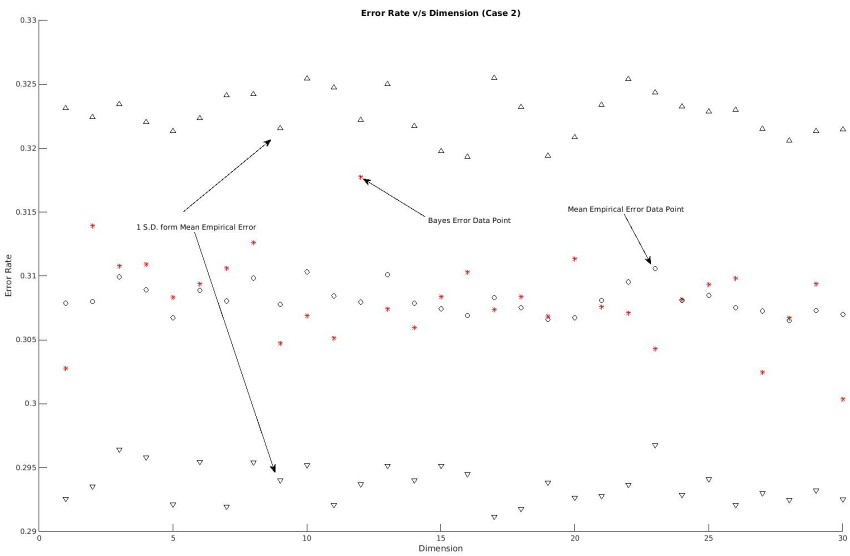
Figure 9: Effect of Dimension on Error Rate (Case 1)



2.3.2 Case 2

Equal Covariances | $\boldsymbol{\mu}_1 = \mathbf{0}_{n\times n}$ | $\boldsymbol{\mu}_2 = \mathbf{I}_{n\times 1}$ | $\boldsymbol{\Sigma}_1 = \mathbf{I}_{n\times n}$ | $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_1$

Figure 10: Effect of Dimension on Error Rate (Case 2)



2.3.3 Comments

As Dimensionality increase and number of classification points as well as distance between means remain same the error rate approaches a wavy nature (see Fig. 9,10). It is difficult to say whether the error rate increases or decreases for both equal and unequal covariance matrix. For equal covariance matrix the decision boundary will be a hyperplane while for unequal covariance matrix it will be non-linear.

3 Conclusion

In this project we have basically compared the discriminant function corresponding to Bayes Decision rule and Bayes Decision Rule itself. For situatuion 1 we see that the error rate decreases with increase in separation of the two gaussians in question because of the fact the overlap hypervolume (mis-classification rate) decreases with it.
Also as the number of classification points increase the mean empirical error remains almost same but the variance decreases which is explained pretty well in the comments section.
The effect of dimensionality on error is still not understood as the error fluctuates a lot showing no trend of increasing or decreasing but very much follows the Bayes Error. The question is to know , When will our classifier work and when will it fail ?

Firstly, when the number of classification points is less and dimensionality is more the classifier will perform very badly as is shown on Fig.8.
Secondly, we cannot assign any class label when $g(\mathbf{x}) = 0$.
Thus we have discussed situations when our discriminant function does not work. It works in the situation that we have plotted in the figures. (**I have included most of the remarks in the comments section rather than the conclusion**)

4 Code

4.1 Function to randomly generate data from 2-Component Gaussian Mixture

Written by Alex Gheith

```
function [ samples , l] = genranddatafu(p1,mu1,sigma1,p2,mu2,sigma2,sampleCount)
N = size(mu1,1);


        % Dataset and labels.
gaussianSamples = zeros([sampleCount, N]);
sampleLabels = zeros([sampleCount, 1]);

        % Control via Uniform PDF.
uniform = rand([sampleCount, 1]);

              % Class 1 Probability. (Threshold)
        % Proper implementation: Logical Indexing.
class1Mask = uniform <= p1;
class1Count = sum(class1Mask);

gaussianSamples(class1Mask, :) = mvnrnd(mu1, sigma1, class1Count);
sampleLabels(class1Mask) = 0;

class2Mask = uniform > p1;
class2Count = sum(class2Mask);

gaussianSamples(class2Mask, :) = mvnrnd(mu2, sigma2, class2Count);
sampleLabels(class2Mask) = 1;

%disp(['Class 1 = ', int2str(class1Count), ', Class 2 = ', int2str(class2Count)]);
samples=gaussianSamples;
l=sampleLabels;
```

4.2 Code to return Bayes Error using Monte Carlo

Written by Joe Zhou

```
function [ z ] = discFunc(x,P1,M1,Sigma1,P2,M2,Sigma2, CFlag)
%compute  and return number of samples that are misclassified
%Written by J. Zhou

[N,M] = size(x); %M samples

invSigma1 = inv(Sigma1);
invSigma2 = inv(Sigma2);
temp = 0.5*log(det(Sigma1)/det(Sigma2)) - log(P1/P2);

y = zeros(1,M);

for i = 1:M

        if  0.5*(x(:,i)-M1)'*invSigma1*(x(:,i)-M1) - ...
                0.5*(x(:,i)-M2)'*invSigma2*(x(:,i)-M2) + temp > 0
            if CFlag == 1
                y(i) = 1;
            else
                y(i) = 0;
            end
        else
            if CFlag == 1
                y(i) = 0;
            else
                y(i) = 1;
            end
        end

end

z = sum(y);

end

-----------------------------------------------------------------------

function [ e ] = BayesError( n, P1, M1, Sigma1, P2, M2, Sigma2, N )
%Compute Bayes Error using Monte-Carlo method
%Written by J. Zhou
```

```
% ....Input Parameters
% n --- data dimension
% P1 --- prior probability of class 1
% M1 --- mean of class 1 data
% Sigma1 --- covariance of class 1 data
% p2 --- prior probability of class 2
% M2 --- mean of class 2 data
% Sigma2 --- covariance of class 2 data
% N --- number of samples using in Monte-Carlo

%....Output Parameters
% e --- Bayes Error

x1 = randn(n,N);
y1 = discFunc(x1, P1, M1, Sigma1, P2, M2, Sigma2, 1);
e1 = y1/N;


temp = randn(n,N);
x2 = sqrt(Sigma2)*temp + repmat(M2, 1, N);
y2 = discFunc(x2, P1, M1, Sigma1, P2, M2, Sigma2, 2);
e2 = y2/N;

% Bayes Error e
e = P1*e1+P2*e2;


end
```

4.3 Empirical Error using Discriminant Function

Written by me

```
function [ z ] = discErr(x,P1,M1,Sigma1,P2,M2,Sigma2, label)
%return fraction of samples that are misclassified
%Modified by Debasmit Das
%0-implies first class ; 1-implies second class


[M,N] = size(x); %M samples

invSigma1 = inv(Sigma1);
```

```
invSigma2 = inv(Sigma2);
temp = 0.5*log(det(Sigma1)/det(Sigma2)) - log(P1/P2);

y = zeros(M,1);

for i = 1:M

        if  0.5*(x(:,i)-M1)'*invSigma1*(x(:,i)-M1) - ...
                0.5*(x(:,i)-M2)'*invSigma2*(x(:,i)-M2) + temp > 0
            if label(i) == 0
                y(i) = 1;
            else
                y(i) = 0;
            end
        else
            if label(i) == 0
                y(i) = 0;
            else
                y(i) = 1;
            end
        end

end

z = sum(y)/M;
```

4.4 Effect of Factors on Error Rate

Written by me

```
%Effect of distance between mean of Gaussians on Error Rate
Prior1=0.5;
Prior2=0.5;
%Dimension of Data Defined
Dimension=2;
Mean1=zeros(Dimension,1);

%Covariance Matrix are set here
Cov1=[1 0.5 ;0.5 1];
Cov2=eye(Dimesnion,Dimension);

plot([],[])
hold on
for d=1:30
```

```
    Mean2=d*eye(Dimension,1);
    e=zeros(100,1);
for j=1:100
    [a,b]= genranddatafu(Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,1000);
    e(j)=discErr(a,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2, b);
    %plot(d,z(i),'gx'); For Plotting the 100 emperical error points
end
    plot(d, BayesError(Dimension,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,1000),'r*')
    plot(d,mean(e),'ko');
    plot(d,mean(e)+sqrt(var(e)),'k^')
    plot(d,mean(e)-sqrt(var(e)),'kv')
end

--------------------------------------------------------------------------------

%Effect of No. of Classification examples on Error Rate
Prior1=0.5;
Prior2=0.5;
dimension=2;

%2D Data
Mean1=zeros(dimension,1);
Mean2=eye(dimension,1);
%Covariance Matrices are same
Cov1=[1, 0.5 ; 0.5 ,1];
Cov2=eye(dimension,dimension);
z=zeros(100,1);
plot([],[])
hold on
for s=1:30

for i=1:100
    [a,b]= genranddatafu(Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,s*5);
    z(i)=discErr(a,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2, b);
    %plot(d,z(i),'gx');
end
    plot(s, BayesError(dimension,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,1000),'r*')
    plot(s,mean(z),'ko');
    plot(s,mean(z)+sqrt(var(z)),'k^')
    plot(s,mean(z)-sqrt(var(z)),'kv')
end
----------------------------------------------------------------------------------
```

```
%Effect of dimensionality on Error Rate
Prior1=0.5;
Prior2=0.5;
n=2;
d=1;
z=zeros(100,1);
plot([],[])
hold on
for dim=1:30
    Mean1=zeros(dim,1);
    Mean2=eye(dim,1);
    %The Means are chosen like this to make sure the euclidean distance is
    %same
    Cov1=eye(dim,dim);
    Cov2=Cov1+eye(dim,1)*eye(dim,1)';
for i=1:100
    [a,b]= genranddatafu(Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,1000);
    z(i)=discErr(a,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2, b);
    %plot(d,z(i),'gx'); % You can use this to plot the emperical error but
    %plot becomes cluttered
end
    plot(dim, BayesError(dim,Prior1,Mean1,Cov1,Prior2,Mean2,Cov2,10000),'r*')
    plot(dim,mean(z),'ko');
    plot(dim,mean(z)+sqrt(var(z)),'k^')
    plot(dim,mean(z)-sqrt(var(z)),'kv')
end
```

References

[1] Fukunaga, Keinosuke. Introduction to statistical pattern recognition. Academic press, 2013.

[2] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley and Sons, 2012.