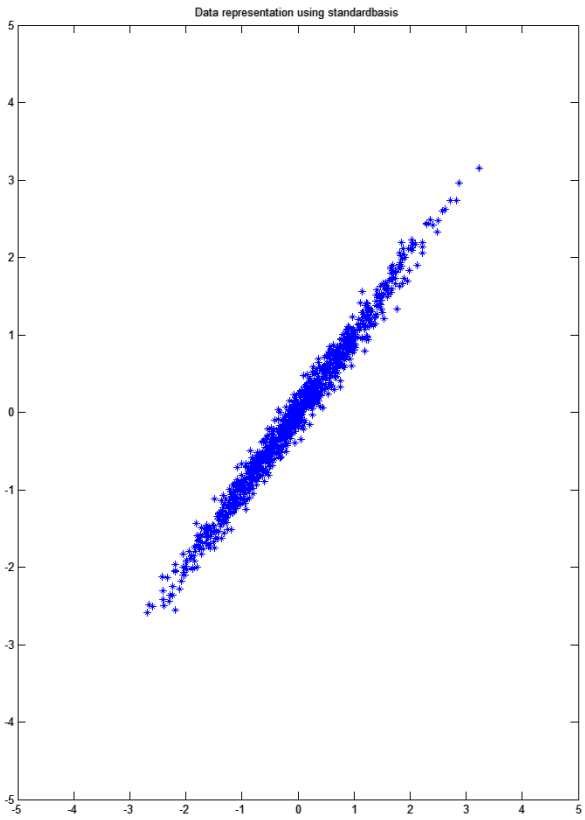
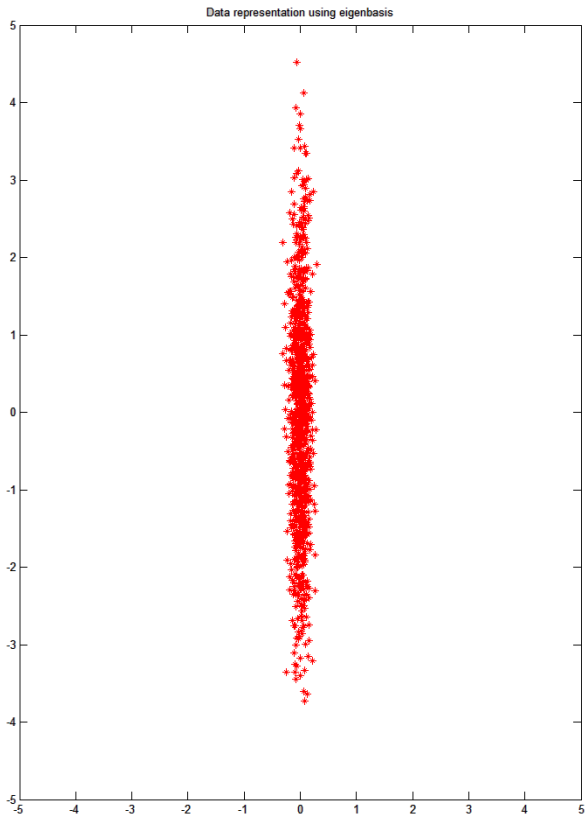
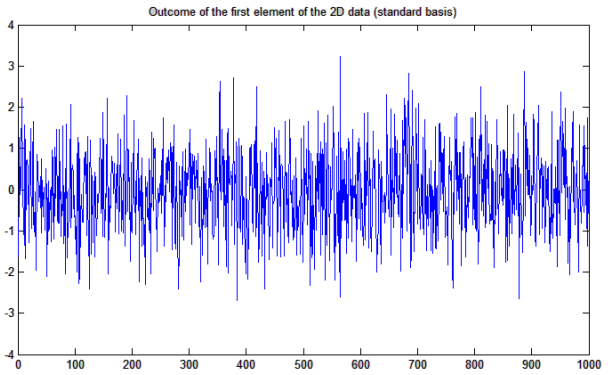
EXPERIMENT 1:PRINCIPAL COMPONENT ANALYSIS

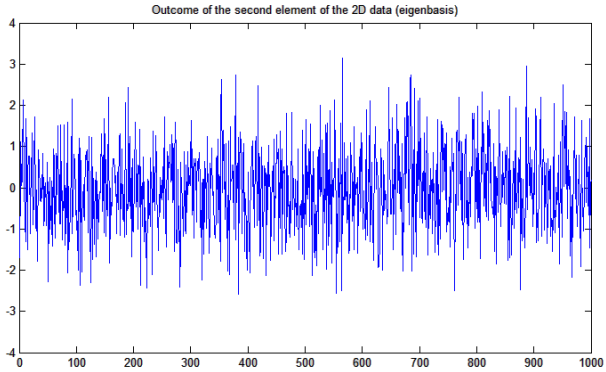
```
%Data generation
CX=[1 0.99;0.99 1];
[E,diag]=eig(CX);
X=randn(2,1000);
X=E*diag^(1/2)*X;
%PCA
CX=cov(X');
[E,D]=eig(CX);
%Projection
Y=E'*X;
figure(1)
subplot(1,2,1)
plot(X(1,:),X(2,:),'*')
title('Data representation using standardbasis')
axis([-5 5 -5 5])
subplot(1,2,2)
plot(Y(1,:),Y(2,:),'*r')
title('Data representation using eigenbasis')
axis([-5 5 -5 5])

figure(2)
subplot(2,2,1)
plot(X(1,:))
title('Outcome of the first element of the 2D data
(standard basis)')
axis([0 1000 -4 4])
subplot(2,2,2)
plot(X(2,:))
title('Outcome of the second element of the 2D data
(eigenbasis)')
axis([0 1000 -4 4])
subplot(2,2,3)
plot(Y(1,:))
title('Outcome of the first element of the 2D data
(eigenbasis)')
axis([0 1000 -4 4])
subplot(2,2,4)
plot(Y(2,:))
title('Outcome of the second element of the 2D data
(eigenbasis)')
axis([0 1000 -4 4])
```

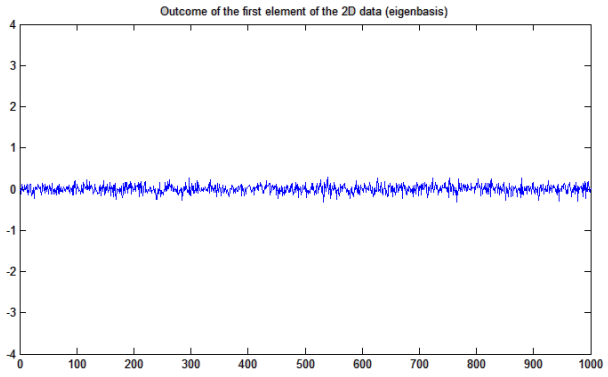
```
%Date reconstruction
Z=[zeros(1,size(Y,2));Y(2,:)];
X1=E*Z;
figure(3)
subplot(2,1,1)
plot(X(1,:),X(2,:),'b*')
title('Original data')
subplot(2,1,2)
plot(X1(1,:),X1(2,:),'r*')
title('Reconstructed data using the significant eigen
vector')
```

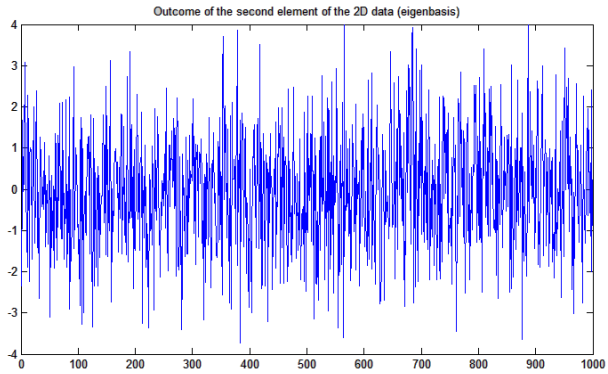


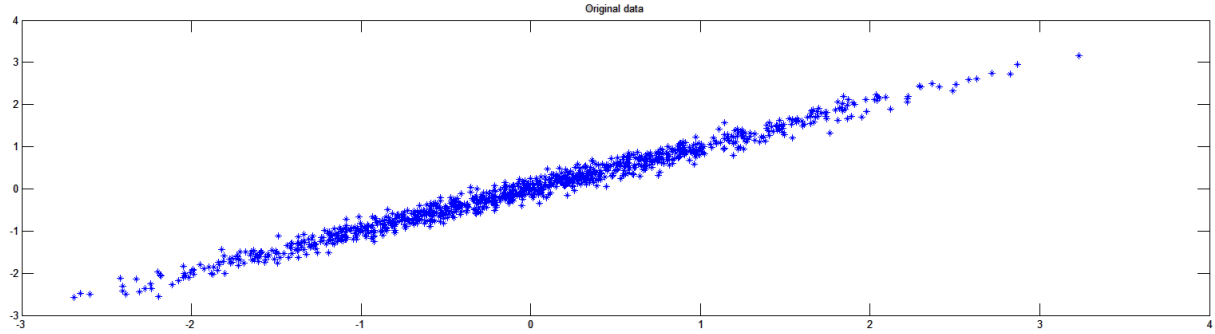


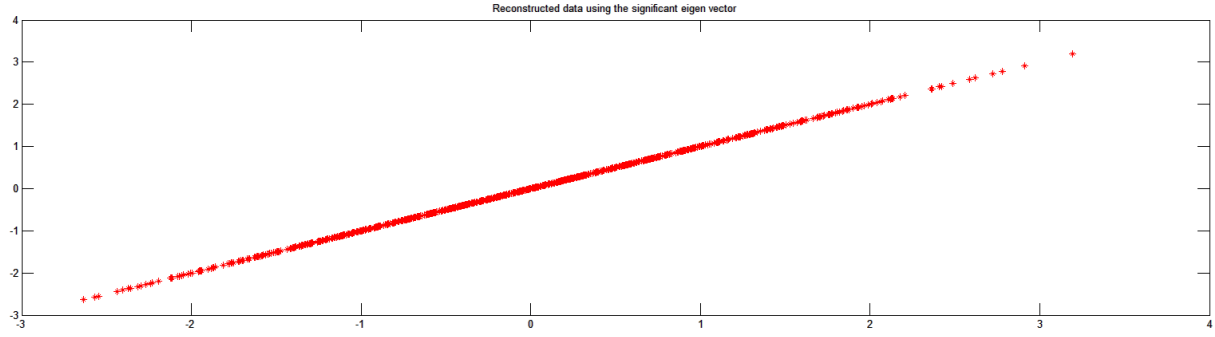

Outcome of the first element of the 2D data (standard basis)


Outcome of the second element of the 2D data (eigenbasis)


Outcome of the first element of the 2D data (eigenbasis)


Outcome of the second element of the 2D data (eigenbasis)


Original data


Reconstructed data using the significant eigen vector

EXPERIMENT 2: LINEAR DISCRIMINANT ANALYSIS

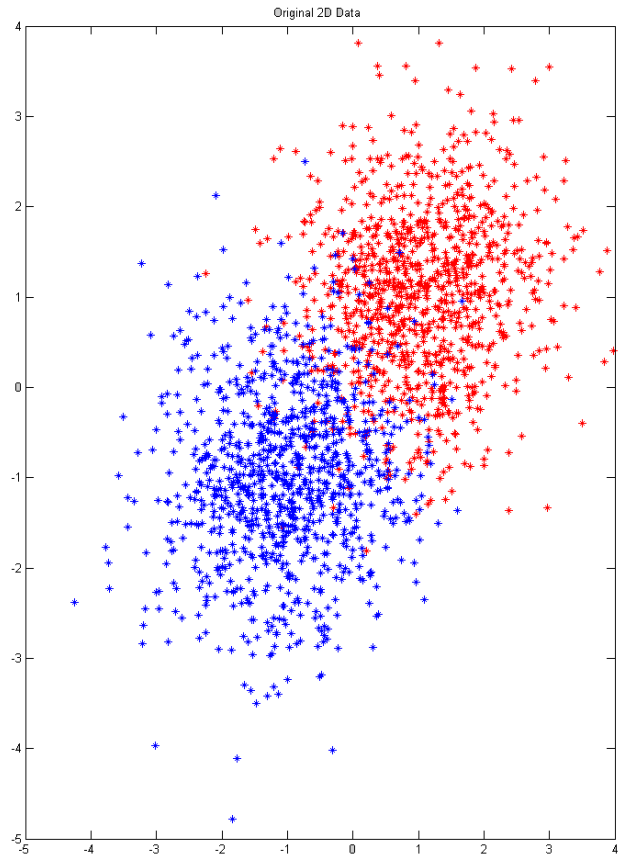
```
%Generate DATA
M1=[1 1];
CX1=[0.9 0.1;0.1 0.9];
[E,diag]=eig(CX1);
X=randn(2,1000);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));

M2=[-1 -1];
CX2=[0.9 0.1;0.1 0.9];
[E,diag]=eig(CX1);
X=randn(2,1000);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));

%LDA
%Between-class scatter matrix
MEAN1=mean(X1');
MEAN2=mean(X2');
SB=cov([MEAN1;MEAN2]);
%Withing-class scatter matrix
SW=(1/2)*(cov(X1')+cov(X2'))
[E,diag]=eig(pinv(SW)*SB);
Y1=E'*X1;
Y2=E'*X2;

figure(1)
subplot(1,2,1)
plot(X1(1,:),X1(2,:),'r*')
title('Original 2D Data ')
hold on
plot(X2(1,:),X2(2,:),'b*')


subplot(1,2,2)
plot(Y1(1,:),Y1(2,:),'r*')
title('2D Data after subjected to LDA ')
hold on
plot(Y2(1,:),Y2(2,:),'b*')

figure(2)
subplot(2,1,1)
plot(X1(1,:),zeros(1,length(X1)),'r*')
```

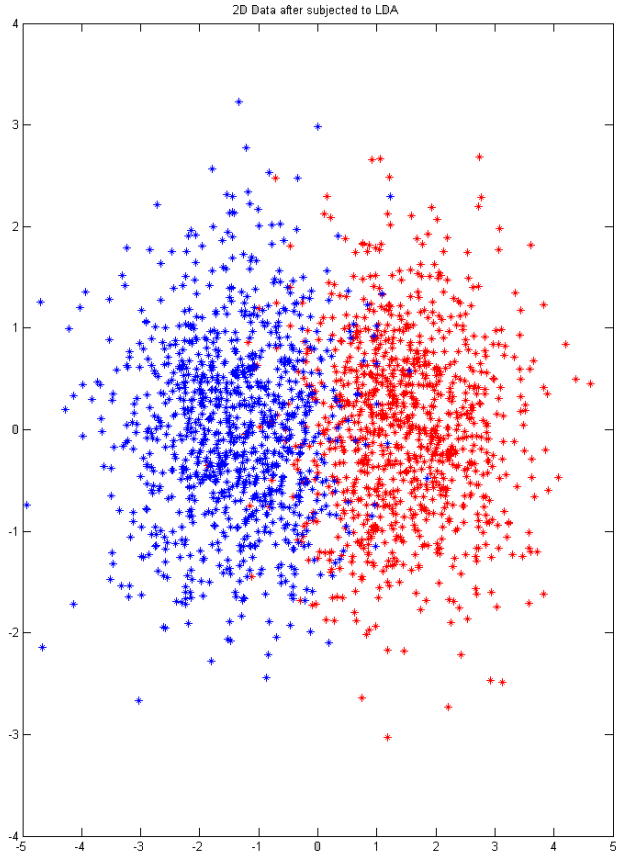
```
axis([-6 6 -1 1])
title('Range of the first element using the standard
basis ')
hold on
plot(X2(1,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])


subplot(2,1,2)
plot(X1(2,:),zeros(1,length(X1)),'r*')
title('Range of the second element using the standard
basis ')
axis([-6 6 -1 1])
hold on
plot(X2(2,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])

figure(3)
subplot(2,1,1)
plot(Y1(1,:),zeros(1,length(Y1)),'r*')
title('Range of the first element using LDA basis ')
axis([-6 6 -1 1])
hold on
plot(Y2(1,:),zeros(1,length(Y1)),'b*')
axis([-6 6 -1 1])

subplot(2,1,2)
plot(Y1(2,:),zeros(1,length(X1)),'r*')
title('Range of the second element using LDA basis ')
axis([-6 6 -1 1])
hold on
plot(Y2(2,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])
```

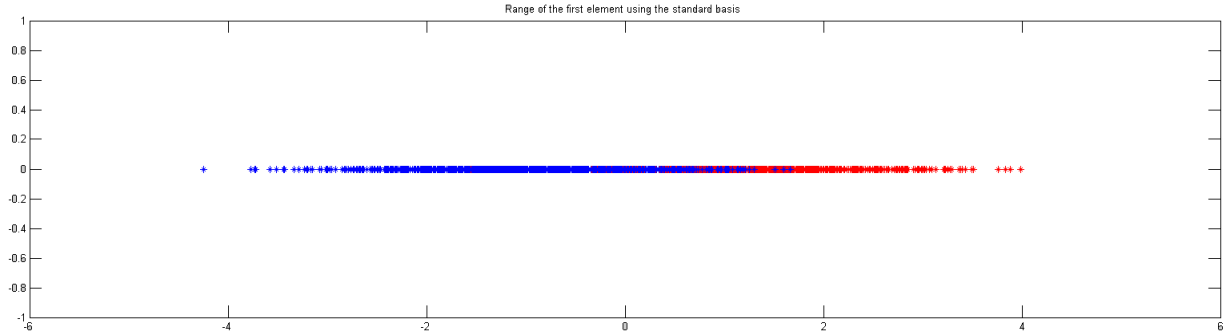
EXPERIMENT 3

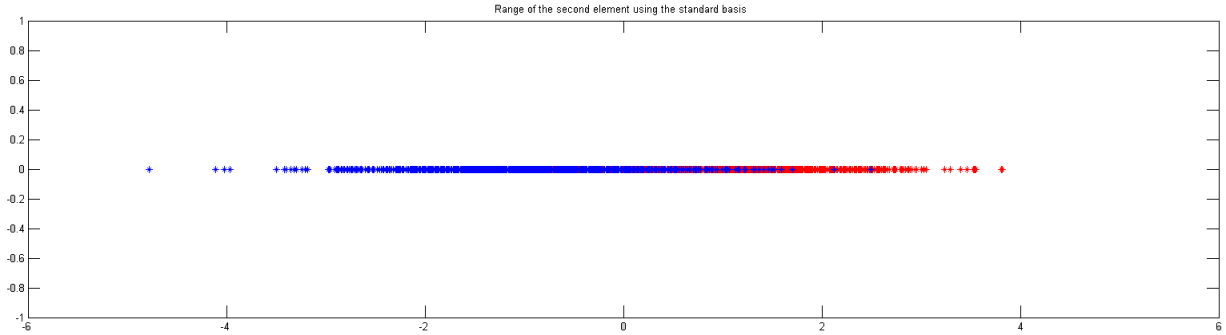
```
%Generate DATA
M1=[1 1];
CX1=[0.9 0.1;0.1 0.9];
[E,diag]=eig(CX1);
X=randn(2,1000);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));

M2=[-1 -1];
CX2=[0.9 0.1;0.1 0.9];
[E,diag]=eig(CX1);
X=randn(2,1000);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));

%LDA
%Between-class scatter matrix
MEAN1=mean(X1');
MEAN2=mean(X2');
SB=cov([MEAN1;MEAN2]);
%Withing-class scatter matrix
SW=(1/2)*(cov(X1')+cov(X2'))
[E,diag]=eig(pinv(SW)*SB);
Y1=E'*X1;
Y2=E'*X2;

figure(1)
subplot(1,2,1)
plot(X1(1,:),X1(2,:),'r*')
title('Original 2D Data ')
hold on
plot(X2(1,:),X2(2,:),'b*')


subplot(1,2,2)
plot(Y1(1,:),Y1(2,:),'r*')
title('2D Data after subjected to LDA ')
hold on
plot(Y2(1,:),Y2(2,:),'b*')
```

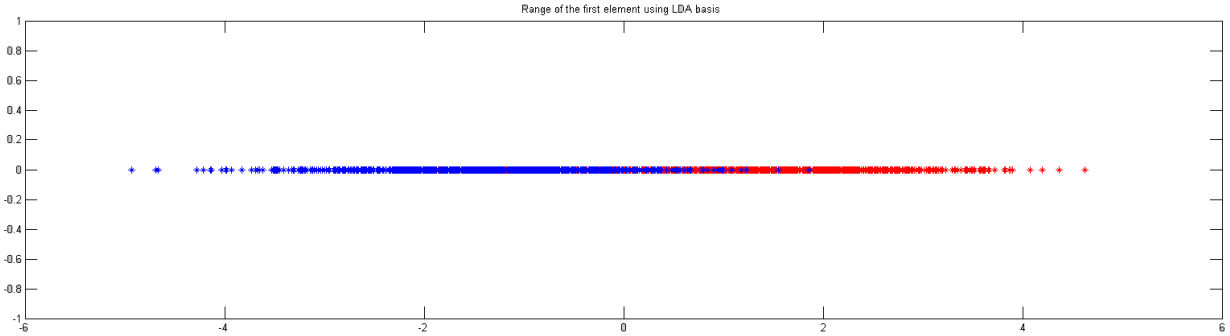
```
figure(2)
subplot(2,1,1)
plot(X1(1,:),zeros(1,length(X1)),'r*')
axis([-6 6 -1 1])
title('Range of the first element using the standard
basis ')
hold on
plot(X2(1,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])


subplot(2,1,2)
plot(X1(2,:),zeros(1,length(X1)),'r*')
title('Range of the second element using the standard
basis ')
axis([-6 6 -1 1])
hold on
plot(X2(2,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])

figure(3)
subplot(2,1,1)
plot(Y1(1,:),zeros(1,length(Y1)),'r*')
title('Range of the first element using LDA basis ')
axis([-6 6 -1 1])
hold on
plot(Y2(1,:),zeros(1,length(Y1)),'b*')
axis([-6 6 -1 1])

subplot(2,1,2)
plot(Y1(2,:),zeros(1,length(X1)),'r*')
title('Range of the second element using LDA basis ')
axis([-6 6 -1 1])
hold on
plot(Y2(2,:),zeros(1,length(X1)),'b*')
axis([-6 6 -1 1])
```

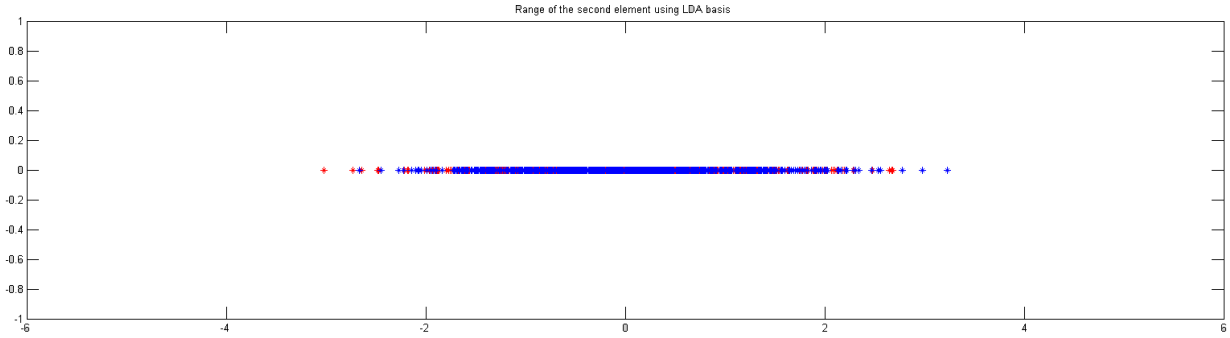

Original 2D Data


2D Data after subjected to LDA


Range of the first element using the standard basis


Range of the second element using the standard basis





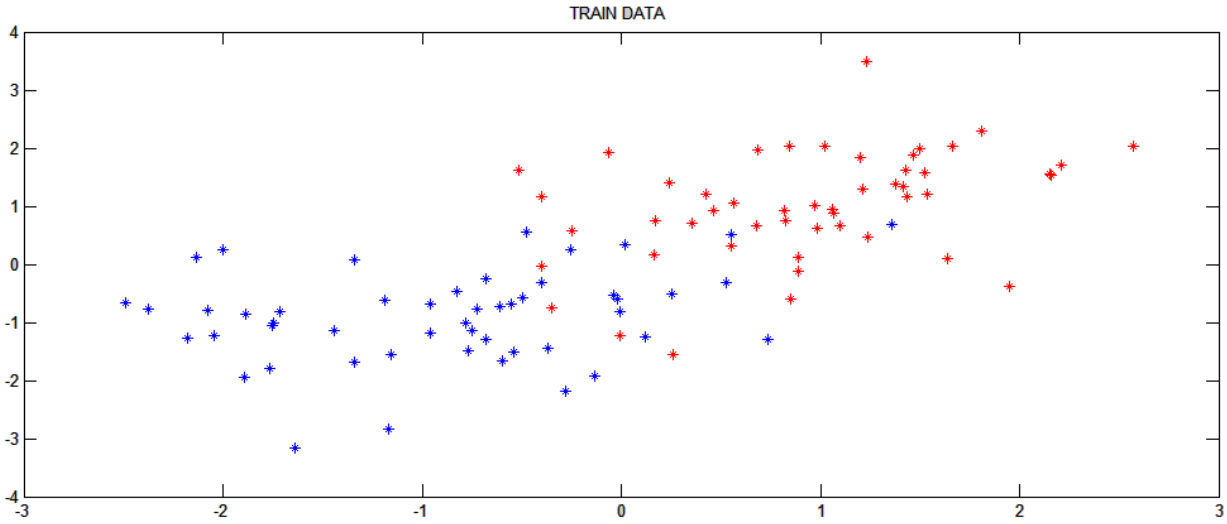
EXPERIMENT 4:KERNEL LINEAR DISCRIMINANT ANALYSIS

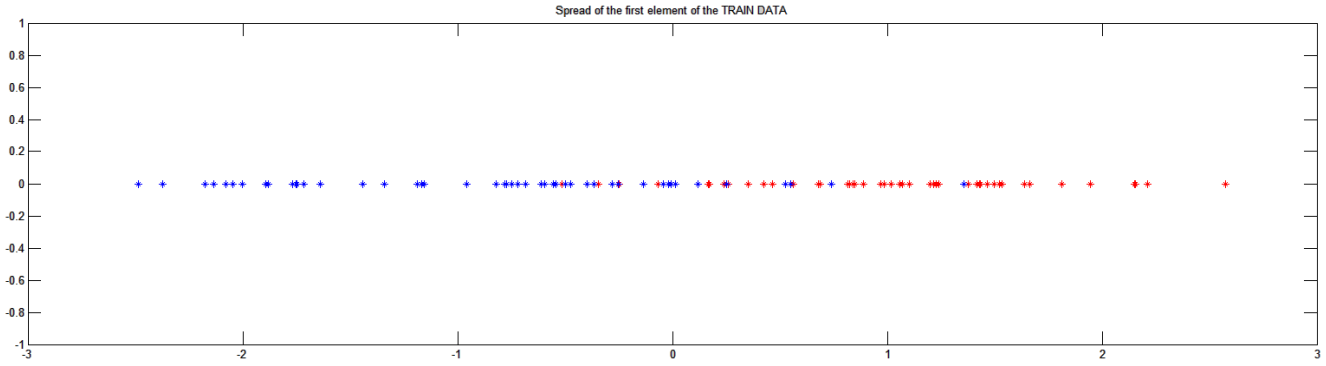
```
close all
clear all
sigma=0.1;
%Generate DATA
M1=[1 1];
CX1=[0.6 0.1;0.1 0.6];
[E,diag]=eig(CX1);
X=randn(2,100);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));

M2=[-1 -1];
CX2=[0.7 0.1;0.1 0.7];
[E,diag]=eig(CX1);
X=randn(2,100);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));

[p,q]=sort(rand(1,100));
TRAINDATA1=X1(:,q(1:1:50));
TRAINDATA2=X2(:,q(1:1:50));
TESTDATA1=X1(:,q(51:1:100));
```

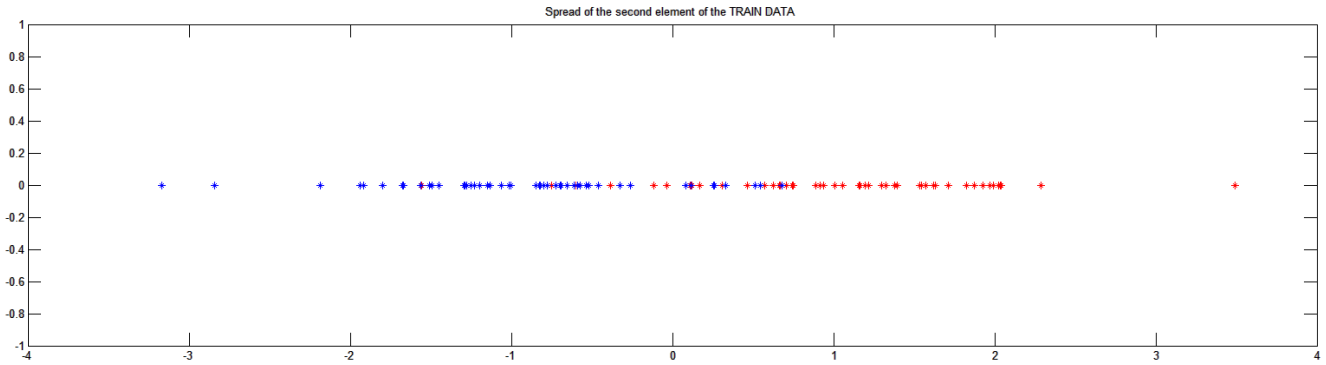
```
TESTDATA2=X2(:,q(51:1:100));
TRAINDATA=[TRAINDATA1 TRAINDATA2];
TESTDATA=[TESTDATA1 TESTDATA2];
figure(1)
subplot(2,1,1)
plot(TRAINDATA1(1,:),TRAINDATA1(2,:),'r*')
title('TRAIN DATA ')
hold on
plot(TRAINDATA2(1,:),TRAINDATA2(2,:),'b*')

subplot(2,1,2)
plot(TESTDATA1(1,:),TESTDATA1(2,:),'r*')
title('TEST DATA')
hold on
plot(TESTDATA2(1,:),TESTDATA2(2,:),'b*')

figure(2)
subplot(2,1,1)
plot(TRAINDATA1(1,:),zeros(1,length(TRAINDATA1)),'r*')
title('Spread of the first element of the TRAIN DATA')
hold on
plot(TRAINDATA2(1,:),zeros(1,length(TRAINDATA2)),'b*')
subplot(2,1,2)
plot(TRAINDATA1(2,:),zeros(1,length(TRAINDATA1)),'r*')
title('Spread of the second element of the TRAIN DATA
')
hold on
plot(TRAINDATA2(2,:),zeros(1,length(TRAINDATA2)),'b*')



figure(3)
subplot(2,1,1)
plot(TESTDATA1(1,:),zeros(1,length(TESTDATA1)),'r*')
title('Spread of the first element of the TESTDATA')
hold on
plot(TESTDATA2(1,:),zeros(1,length(TESTDATA2)),'b*')

subplot(2,1,2)
plot(TESTDATA1(2,:),zeros(1,length(TESTDATA1)),'r*')
title('Spread of the second element of the TEST DATA')
hold on
plot(TESTDATA2(2,:),zeros(1,length(TESTDATA2)),'b*')
```

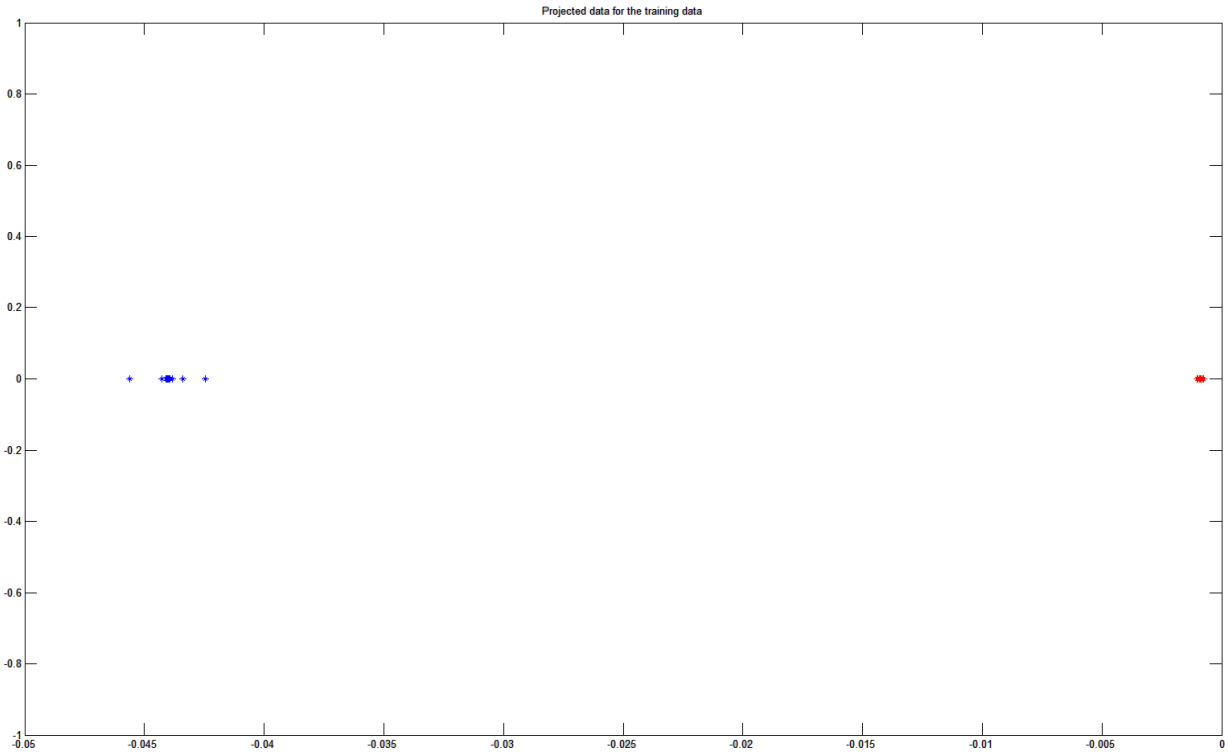
```
%KLDA
%FORMULATING GRAM-MATRIX using TRAINDATA
G=[];
for i=1:1:100
    G1=[];
    for j=1:1:100

G1=[G1;gausskernel(TRAINDATA(:,i),TRAINDATA(:,j),sigma)
];
    end
    G=[G G1];
end
X1=G(:,1:1:50);
X2=G(:,51:100);
%Between-class scatter matrix
MEAN1=mean(X1');
MEAN2=mean(X2');
SB=cov([MEAN1;MEAN2]);
%Withing-class scatter matrix
SW=(1/2)*(cov(X1')+cov(X2'))
[E,diag]=eigs(pinv(SW)*SB,1);
%Projecting the training data
Y1=E'*X1;
Y2=E'*X2;
figure(4)
plot(Y1,zeros(1,length(Y1)),'r*')
title('Projected data for the training data')
hold on
plot(Y2,zeros(1,length(Y2)),'b*')

%Projecting the testing data
PR=[];
for i=1:1:100
    temp=[];
    for j=1:1:100
    temp=[temp
gausskernel(TESTDATA(:,i),TRAINDATA(:,j),sigma)];
    end
    PR=[PR E'*temp'];
end
```

```
PR=real(PR);
figure(5)
plot(PR(1:1:50),zeros(1,length(Y2)),'r*')
title('Projected data for the testing data ')
hold on
plot(PR(51:1:100),zeros(1,length(Y2)),'b*')
```







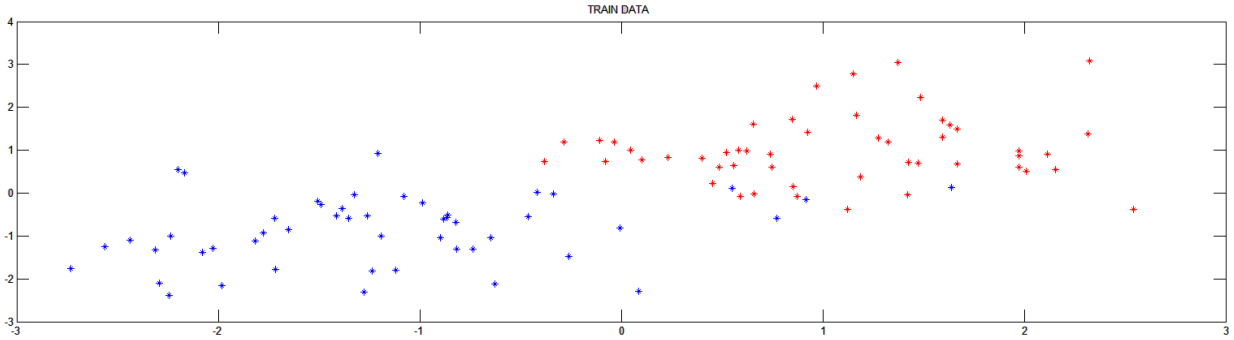


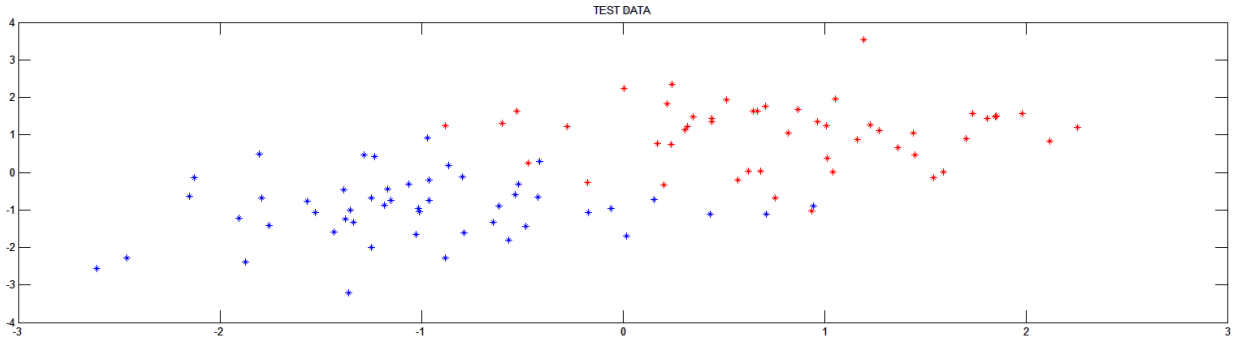
EXPERIMENT 5:SUPPORT VECTOR MACHINE USING KERNEL TRICK

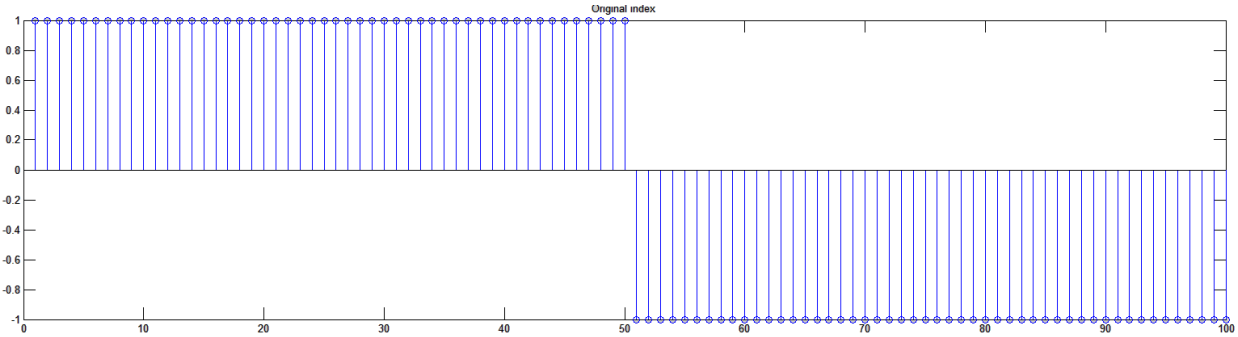
```
%SVM in the higher dimensional space
close all
clear all
sigma=0.1;
%Generate DATA
M1=[1 1];
CX1=[0.6 0.1;0.1 0.6];
[E,diag]=eig(CX1);
X=randn(2,100);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));

M2=[-1 -1];
CX2=[0.7 0.1;0.1 0.7];
[E,diag]=eig(CX1);
X=randn(2,100);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));
```

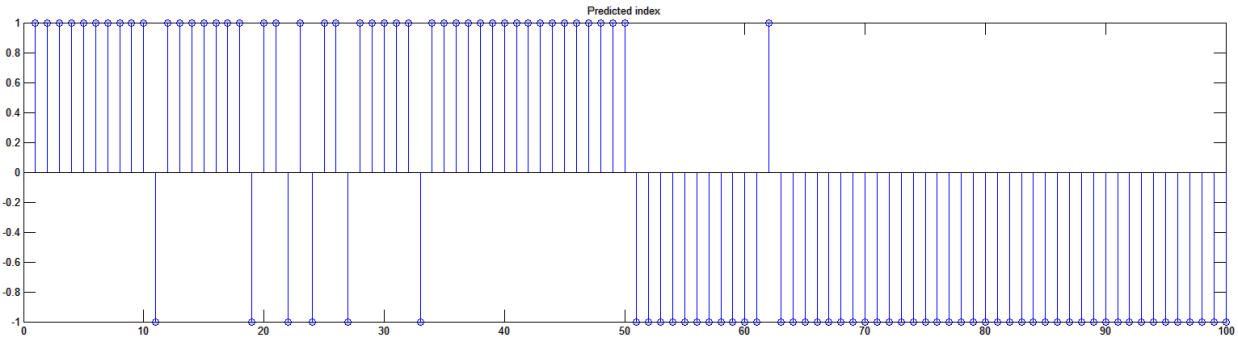
```
[p,q]=sort(rand(1,100));
TRAINDATA1=X1(:,q(1:1:50));
TRAINDATA2=X2(:,q(1:1:50));
TESTDATA1=X1(:,q(51:1:100));
TESTDATA2=X2(:,q(51:1:100));
TRAINDATA=[TRAINDATA1 TRAINDATA2];
TESTDATA=[TESTDATA1 TESTDATA2];
figure(1)
subplot(2,1,1)
plot(TRAINDATA1(1,:),TRAINDATA1(2,:),'r*')
title('TRAIN DATA ')
hold on
plot(TRAINDATA2(1,:),TRAINDATA2(2,:),'b*')

subplot(2,1,2)
plot(TESTDATA1(1,:),TESTDATA1(2,:),'r*')
title('TEST DATA')
hold on
plot(TESTDATA2(1,:),TESTDATA2(2,:),'b*')

T=[ones(1,50) ones(1,50)*(-1)];
%Construct the matrix
G=[];
for i=1:1:length(TRAINDATA)
    M=[];
for j=1:1:length(TRAINDATA)
M=[M
gausskernel(TRAINDATA(:,i),TRAINDATA(:,j),sigma)*T(i)*T
(j)];
end
G=[G;M];
end
G=[G;T];
lambda=pinv(G)*[ones(1,length(TRAINDATA)) 0]';
[p,q]=find(lambda>0);
s=0;
for i=1:1:length(p)
    s1=0
    for j=1:1:length(p)

s1=s1+lambda(p(i))*T(p(i))*gausskernel(TRAINDATA(:,p(i)
),TRAINDATA(:,p(j)),sigma);
    end
```

```
    s=s+(T(i)-s1)
end
b=s/length(p);
%Check with the testing data
res=0;
R=[];
for i=1:1:length(TESTDATA)
    res=0;
    for j=1:1:length(p)
res=res+lambda(p(j))*T(p(j))*gausskernel(TESTDATA(:,i),
TRAINDATA(:,p(j)),sigma);
    end
R=[R  res];
end
RES=[];
for i=1:1:length(R)
    if(R(i)>0)
RES=[RES 1];
    else
RES=[RES -1];
    end
end
[U,V]=find(T==RES);
POS=(length(U)/length(T))*100;
figure
subplot(2,1,1)
stem(T)
title('Original index')
subplot(2,1,2)
stem(RES)
title('Predicted index')
```


TRAIN DATA


TEST DATA

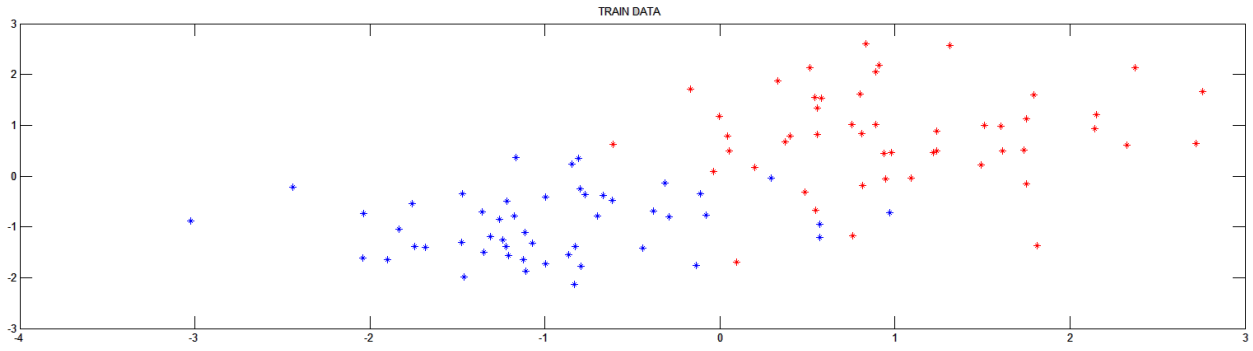

Original index

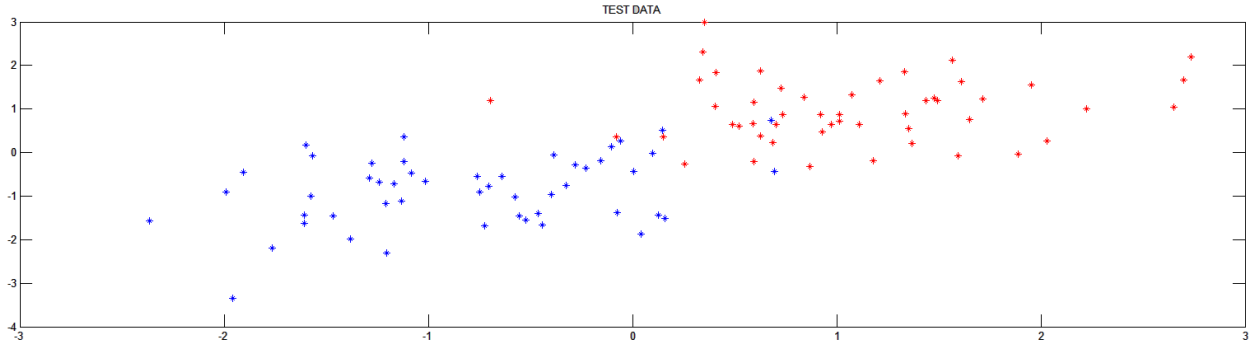

Predicted index

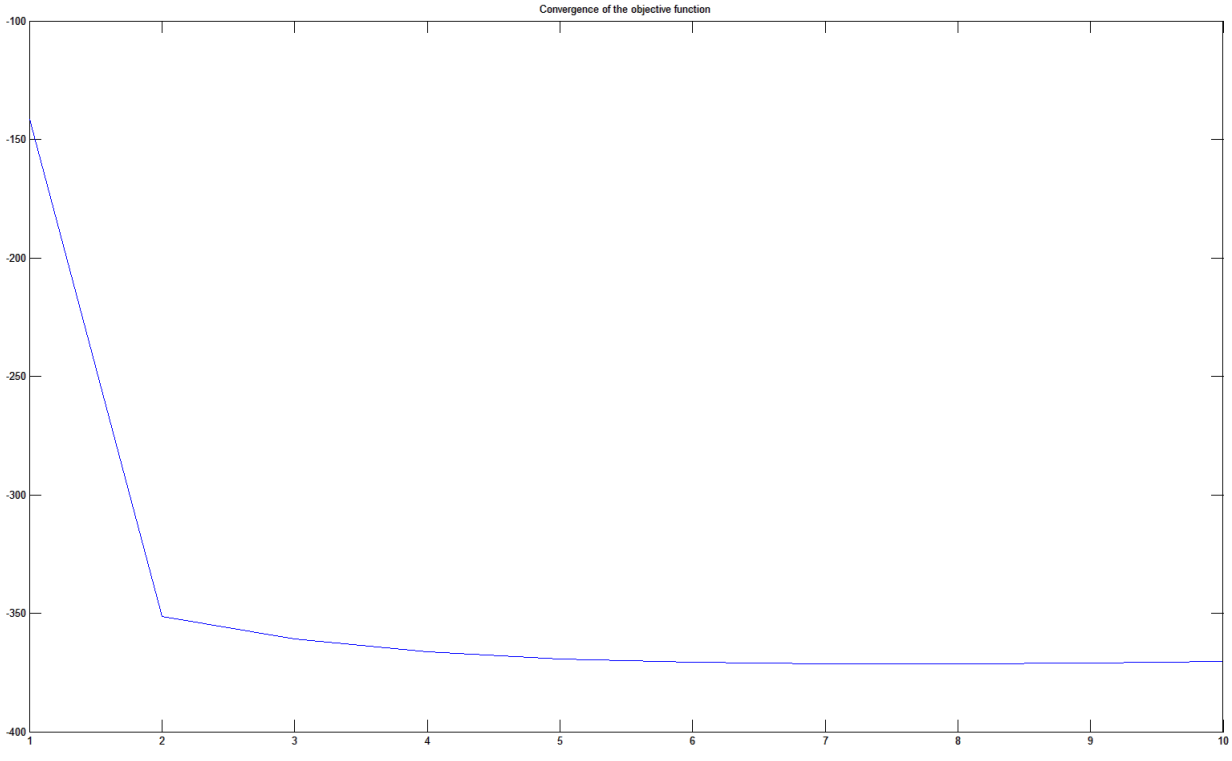
EXPERIMENT 6: LOGISTIC REGRESSION

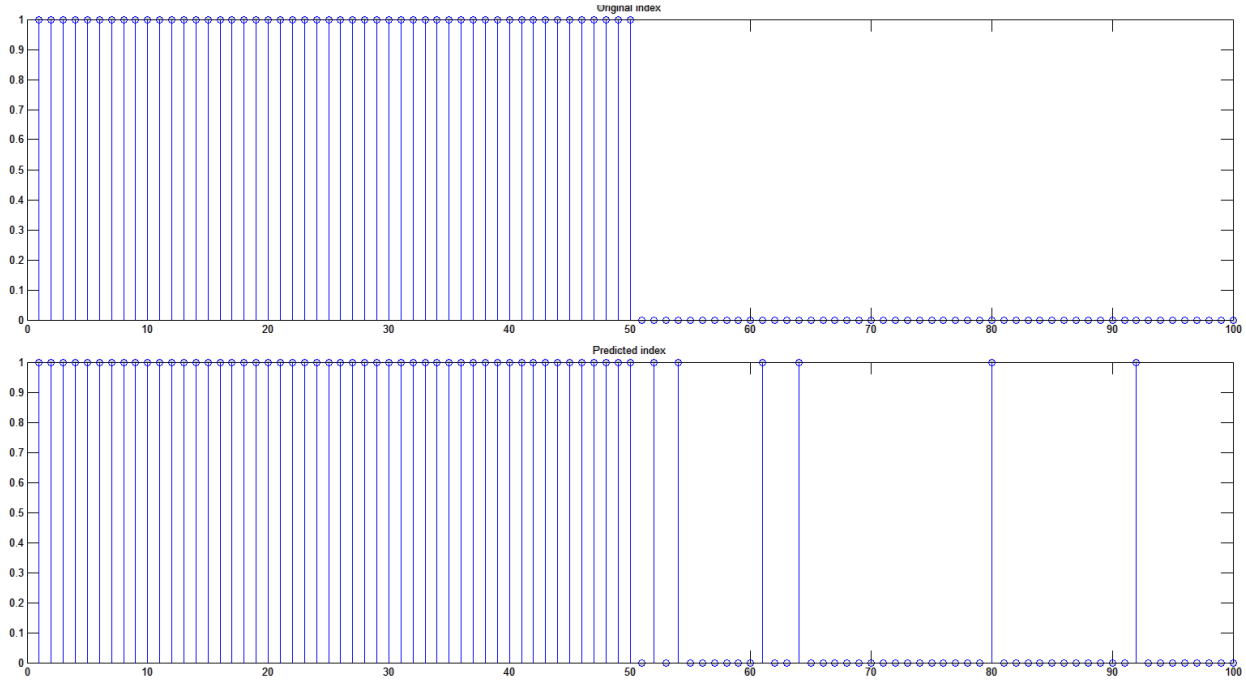
```
close all
clear all
%Generate DATA
%CLASS 1
M1=[1 1];
CX1=[0.6 0.1;0.1 0.6];
[E,diag]=eig(CX1);
X=randn(2,100);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));

%CLASS 2
M2=[-1 -1];
CX2=[0.7 0.1;0.1 0.7];
[E,diag]=eig(CX1);
X=randn(2,100);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));

t=[ones(1,50) zeros(1,50)];
%Initialize w

[p,q]=sort(rand(1,100));
TRAINDATA1=X1(:,q(1:1:50));
TRAINDATA2=X2(:,q(1:1:50));
TESTDATA1=X1(:,q(51:1:100));
TESTDATA2=X2(:,q(51:1:100));
TRAINDATA=[TRAINDATA1 TRAINDATA2];
TESTDATA=[TESTDATA1 TESTDATA2];
figure(1)
subplot(2,1,1)
plot(TRAINDATA1(1,:),TRAINDATA1(2,:),'r*')
title('TRAIN DATA ')
hold on
plot(TRAINDATA2(1,:),TRAINDATA2(2,:),'b*')

subplot(2,1,2)
plot(TESTDATA1(1,:),TESTDATA1(2,:),'r*')
title('TEST DATA')
hold on
plot(TESTDATA2(1,:),TESTDATA2(2,:),'b*')
```

```
w=rand(1,4);
COL=[];
for iteration=1:1:10
lc=0.1;
ytrain=[];
for i=1:1:100
ytrain=[ytrain 1/(1+exp(-w(1)-w(2)*TRAINDATA(1,i)-
w(3)*TRAINDATA(2,i)-
w(4)*TRAINDATA(1,i)*TRAINDATA(2,i)))];
end
J=sum(t(1:1:100).*log(ytrain(1:1:100)))+sum(1-
t(1:1:100).*log(1-ytrain(1:1:100)));
COL=[COL -1*J];
s=0;
for i=1:1:100
    s=s+(ytrain(i)-t(i))*[1 TRAINDATA(1,i)
TRAINDATA(2,i) TRAINDATA(1,i)*TRAINDATA(2,i)]
end
w=w-lc*(s);
end
ytest=[];
for i=1:1:100
ytest=[ytest 1/(1+exp(-w(1)-w(2)*TESTDATA(1,i)-
w(3)*TESTDATA(2,i)-w(4)*TESTDATA(1,i)*TESTDATA(2,i)))];
end
[t;round(ytest)]
figure(2)
plot(COL)
title('Convergence of the objective function')

[U,V]=find(t==round(ytest));
POS=(length(U)/length(t))*100;
figure(3)
subplot(2,1,1)
stem(t)
title('Original index')
subplot(2,1,2)
stem(round(ytest))
title('Predicted index')
```


TRAIN DATA
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3


TEST DATA
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3

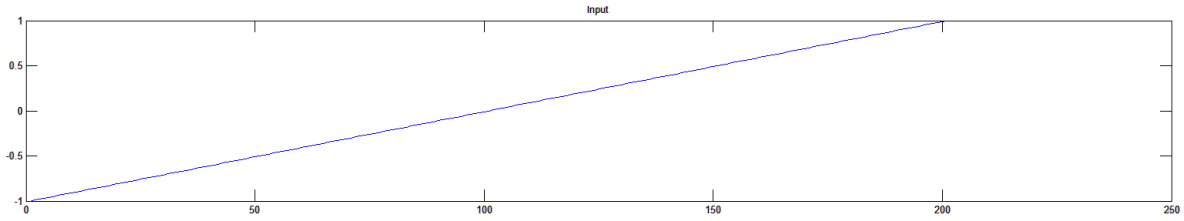

Convergence of the objective function
-100
-150
-200
-250
-300
-350
-400
1
2
3
4
5
6
7
8
9
10

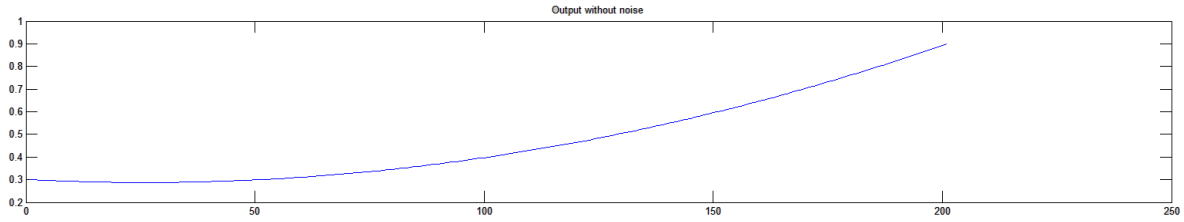


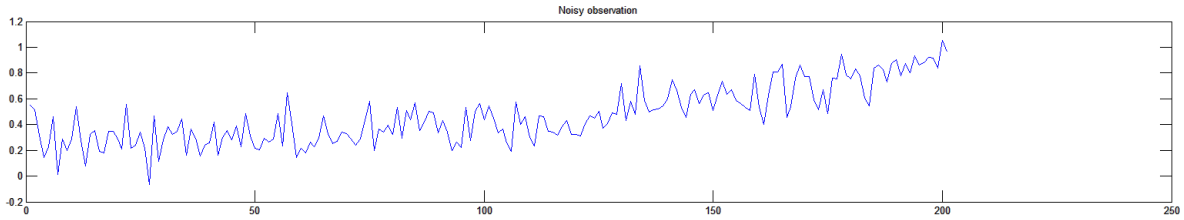
EXPERIMENT 6 LINEAR REGRESSION USING POLYNOMIAL KERNEL

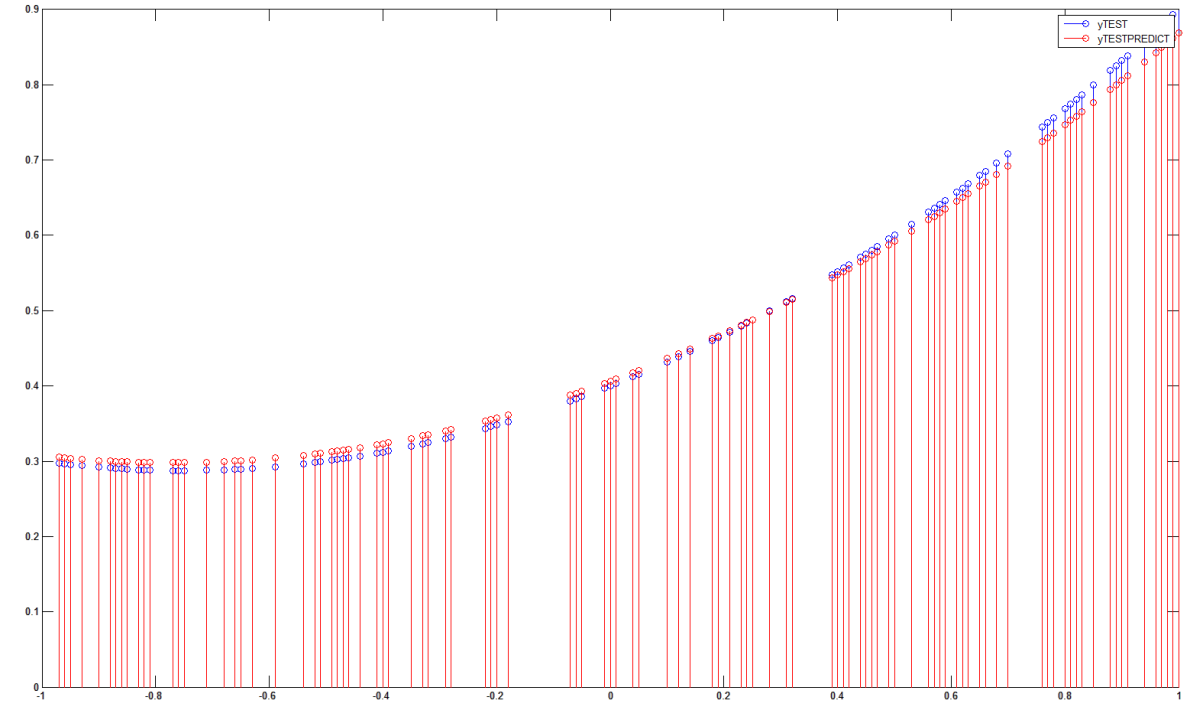
```
%LINREG (Demonstration using Polynomial kernel)
%Data generation
x=-1:0.01:1;
y=0.2*x.^2+0.3*x+0.4;
t=y+randn(1,length(x))*0.1;
figure
subplot(3,1,1)
plot(x)
title('Input')
subplot(3,1,2)
plot(y)
title('Output without noise')
subplot(3,1,3)
plot(t)
title('Noisy observation')

%TRAIN-TEST SPLIT
L=length(x);
[p,q]=sort(rand(1,L));
```

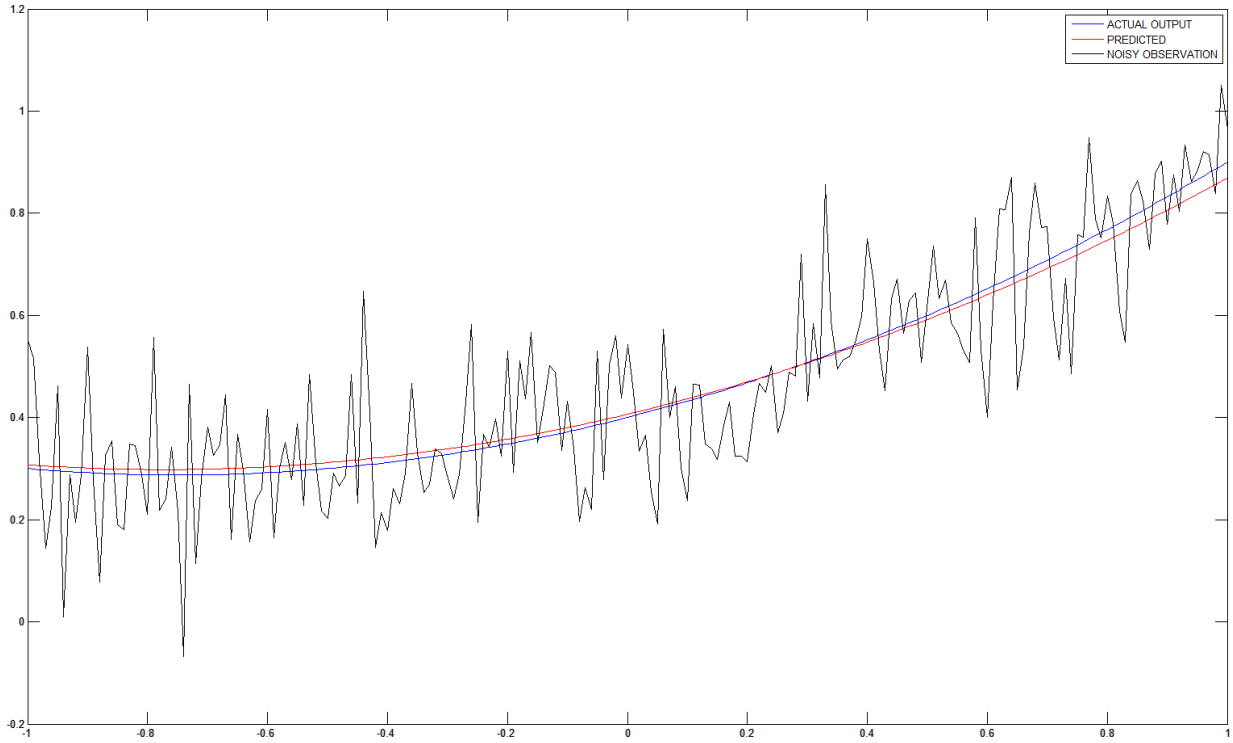
```
TRAINPOS=q(1:1:fix(L/2));
TESTPOS=q(fix(L/2)+1:1:L);

%Least square solution using Linear regression with
polynomial regression
%Constructing PHI matrix
xTRAIN=x(TRAINPOS);
tTRAIN=t(TRAINPOS);
xTEST=x(TESTPOS);
yTEST=y(TESTPOS);


PHI=[ones(1,length(TRAINPOS)); xTRAIN ;xTRAIN.^2]';
w=pinv(PHI'*PHI)*PHI'*tTRAIN';
%FIT using
yTESTPREDICT=w(1)+w(2)*xTEST+w(3)*xTEST.^2;
figure
stem(xTEST,yTEST)
hold on
stem(xTEST,yTESTPREDICT,'r')
legend('yTEST','yTESTPREDICT')
yPREDICT=w(1)+w(2)*x+w(3)*x.^2;
figure
plot(x,y)
hold on
plot(x,yPREDICT,'r')
plot(x,t,'k')
legend('ACTUAL OUTPUT','PREDICTED','NOISY OBSERVATION')
```


Input
1
0.5
0
-0.5
-1
0
50
100
150
200
250


Output without noise
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
50
100
150
200
250


Noisy observation
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
50
100
150
200
250

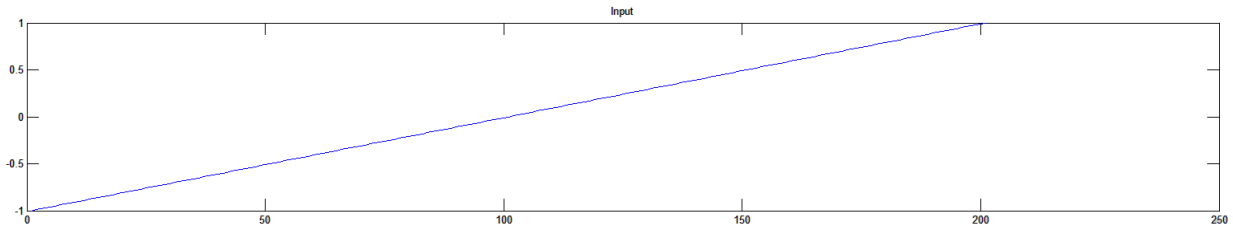

yTEST
yTESTPREDICT
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

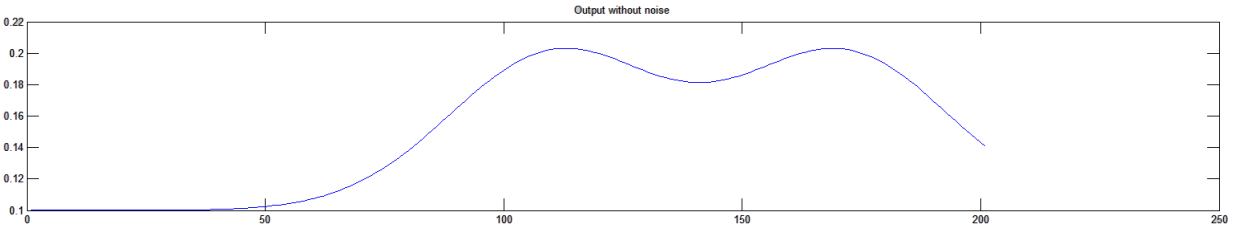


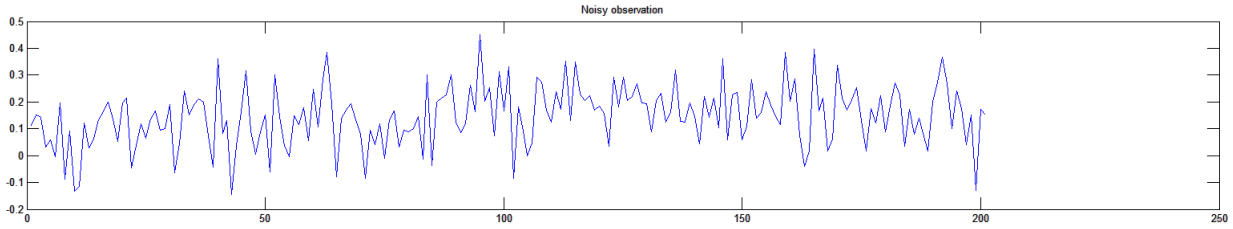
EXPERIMENT 7: LINEAR REGRESSION USING GAUSSIAN KERNEL

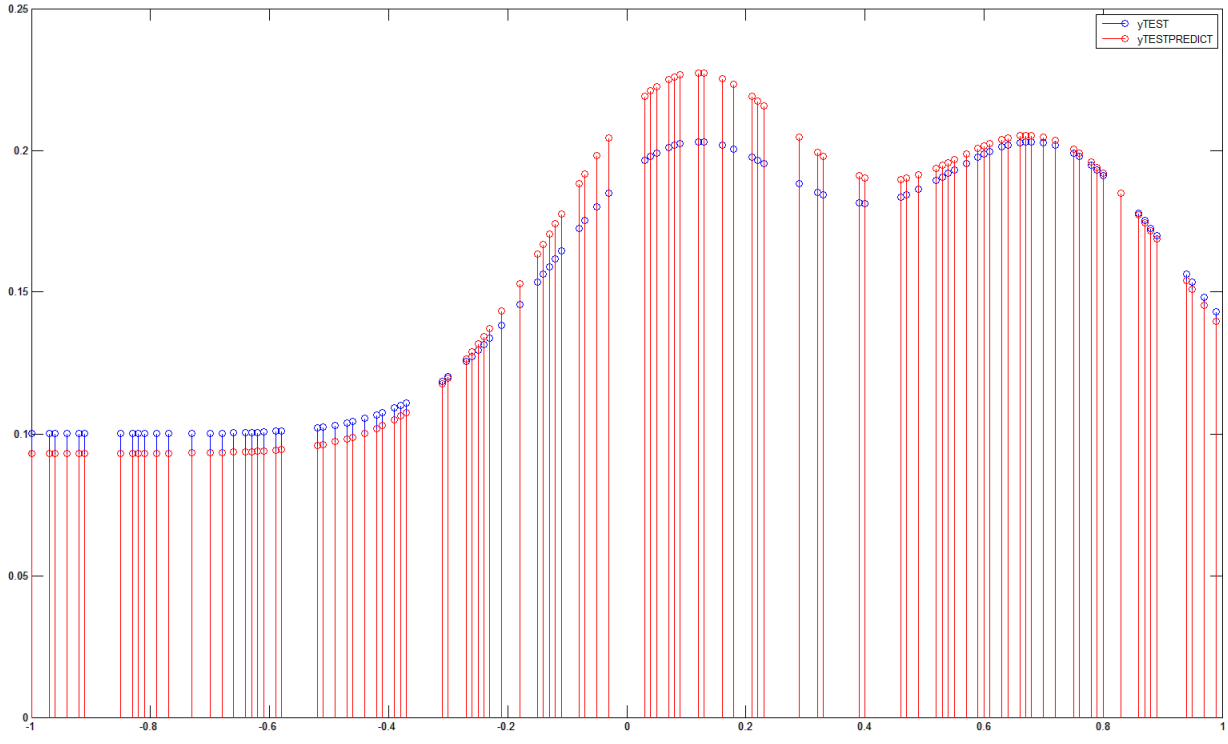
```
%LINREG (Demonstration using Gaussian kernel)
%Data generation
x=-1:0.01:1;
y=0.1*exp(-(x-0.1).^2/0.1)+0.1*exp(-(x-
0.7).^2/0.1)+0.1;
t=y+randn(1,length(x))*0.1;

figure
subplot(3,1,1)
plot(x)
title('Input')
subplot(3,1,2)
plot(y)
title('Output without noise')
subplot(3,1,3)
plot(t)
title('Noisy observation')

%TRAIN-TEST SPLIT
L=length(x);
[p,q]=sort(rand(1,L));
```

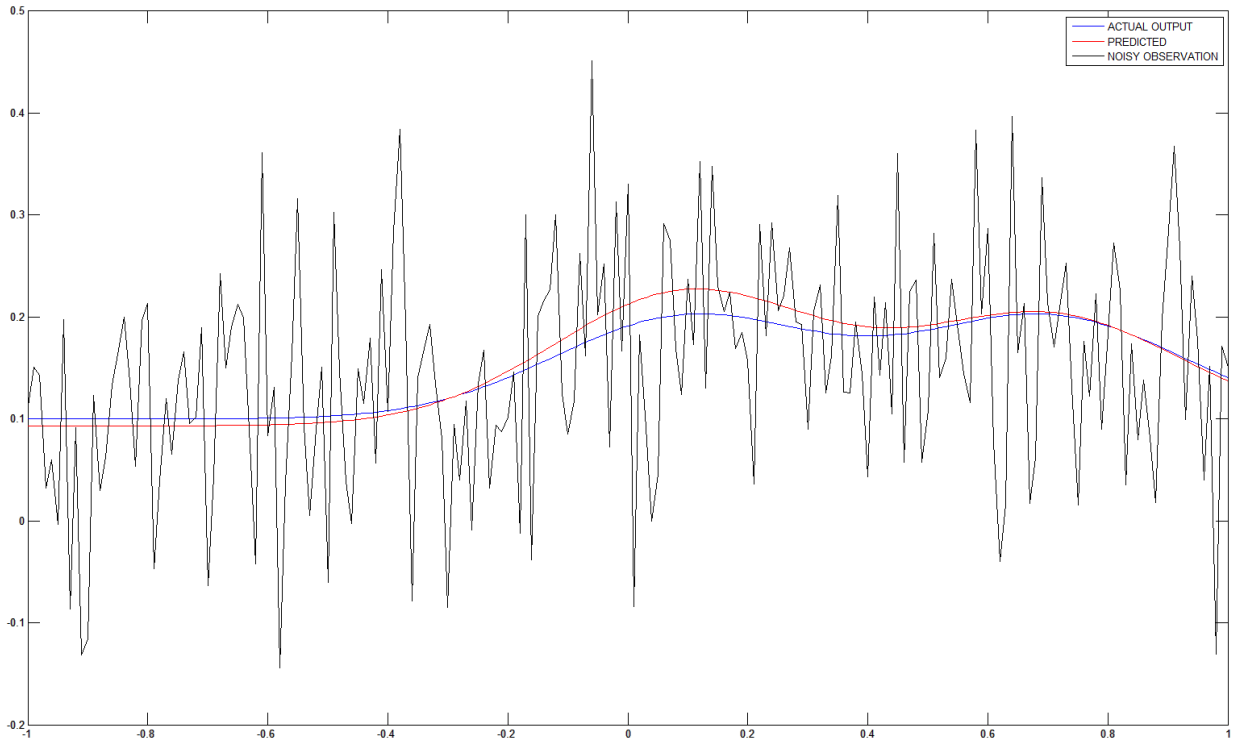
```
TRAINPOS=q(1:1:fix(L/2));
TESTPOS=q(fix(L/2)+1:1:L);

%Least square solution using Linear regression with
polynomial regression
%Constructing PHI matrix
xTRAIN=x(TRAINPOS);
tTRAIN=t(TRAINPOS);
xTEST=x(TESTPOS);
yTEST=y(TESTPOS);
PHI=[ones(1,length(TRAINPOS)); exp(-(xTRAIN-
0.1).^2/0.1) ;exp(-(xTRAIN-0.7).^2/0.1)]';
w=pinv(PHI'*PHI)*PHI'*tTRAIN';
%FIT using
yTESTPREDICT=w(1)+w(2)*exp(-(xTEST-
0.1).^2/0.1)+w(3)*exp(-(xTEST-0.7).^2/0.1);

figure
stem(xTEST,yTEST)
hold on
stem(xTEST,yTESTPREDICT,'r')
legend('yTEST','yTESTPREDICT')
yPREDICT=w(1)+w(2)*exp(-(x-0.1).^2/0.1)+w(3)*exp(-(x-
0.7).^2/0.1);figure
plot(x,y)
hold on
plot(x,yPREDICT,'r')
plot(x,t,'k')
legend('ACTUAL OUTPUT','PREDICTED','NOISY OBSERVATION')
```


Input
1
0.5
0
-0.5
-1
0
50
100
150
200
250


Output without noise
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0
50
100
150
200
250


Noisy observation
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
50
100
150
200
250

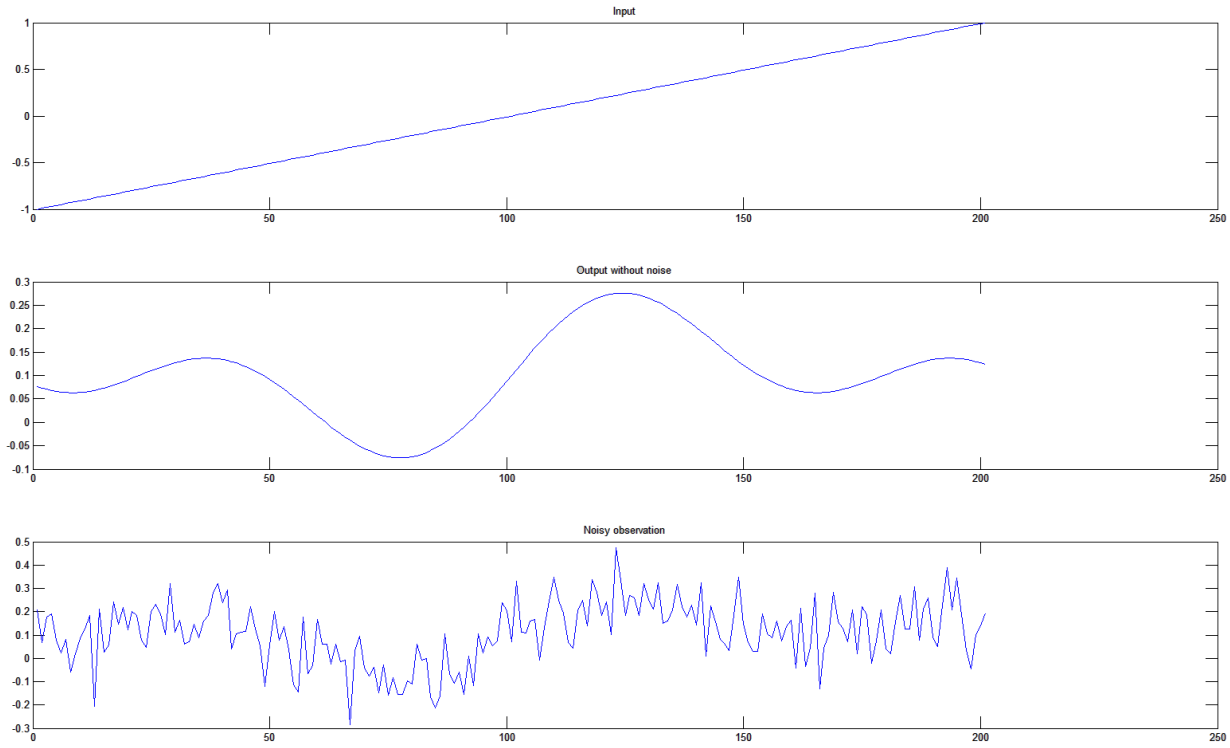

yTEST
yTESTPREDICT
0.25
0.2
0.15
0.1
0.05
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

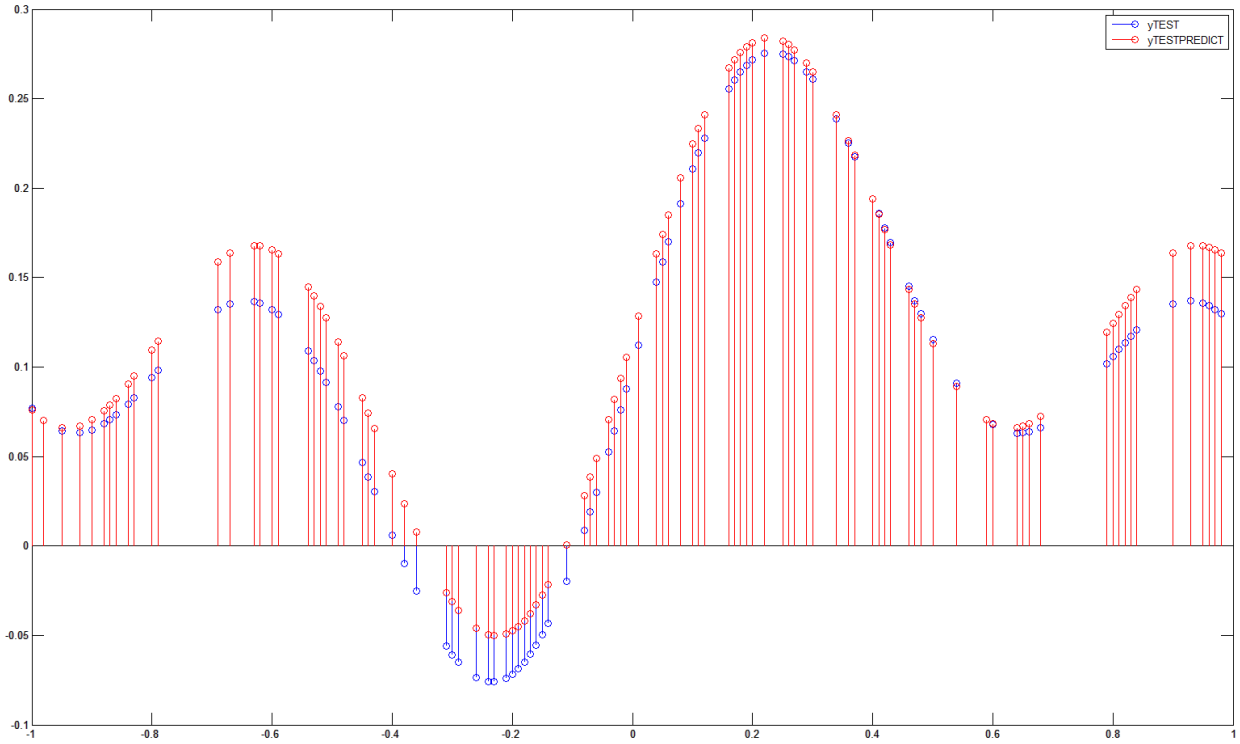


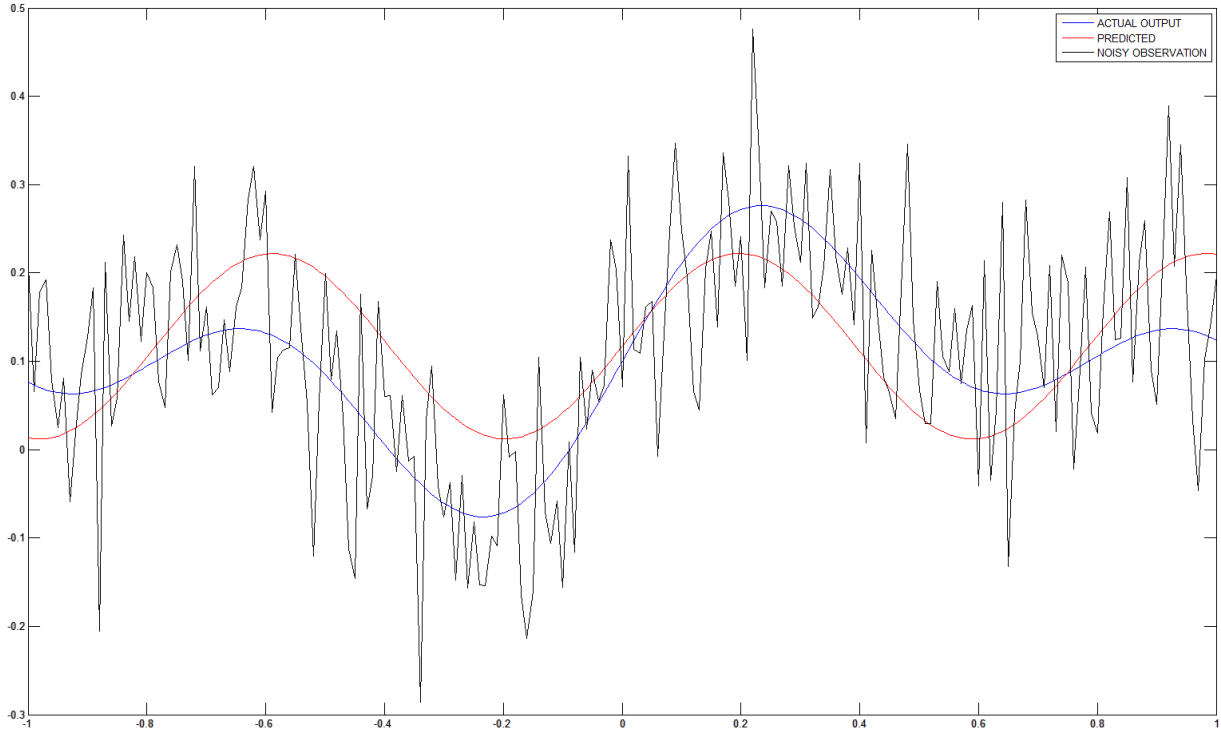
EXPERIMENT 8: LINEAR REGRESSION USING TRIGNOMETRY EXPANSION

```
%LINREG (Demonstration using TRIGNOMETRY)
%Data generation
x=-1:0.01:1;
y=0.1*sin(8*x)+0.1*sin(4*x)+0.1;
t=y+randn(1,length(x))*0.1;
figure
subplot(3,1,1)
plot(x)
title('Input')
subplot(3,1,2)
plot(y)
title('Output without noise')
subplot(3,1,3)
plot(t)
title('Noisy observation')
%TRAIN-TEST SPLIT
L=length(x);
[p,q]=sort(rand(1,L));
TRAINPOS=q(1:1:fix(L/2));
TESTPOS=q(fix(L/2)+1:1:L);
```

```
%Least square solution using Linear regression with
polynomial regression
%Constructing PHI matrix
xTRAIN=x(TRAINPOS);
tTRAIN=t(TRAINPOS);
xTEST=x(TESTPOS);
yTEST=y(TESTPOS);
PHI=[ones(1,length(TRAINPOS));
sin(8*xTRAIN);sin(4*xTRAIN)]';
w=pinv(PHI'*PHI)*PHI'*tTRAIN';
%FIT using
yTESTPREDICT=w(1)+w(2)*sin(8*xTEST)+w(3)*sin(4*xTEST);
figure
stem(xTEST,yTEST)
hold on
stem(xTEST,yTESTPREDICT,'r')
legend('yTEST','yTESTPREDICT')
yPREDICT=w(1)+w(2)*sin(8*x);+w(3)*sin(4*x);
figure
plot(x,y)
hold on
plot(x,yPREDICT,'r')
plot(x,t,'k')
legend('ACTUAL OUTPUT','PREDICTED','NOISY OBSERVATION')
```



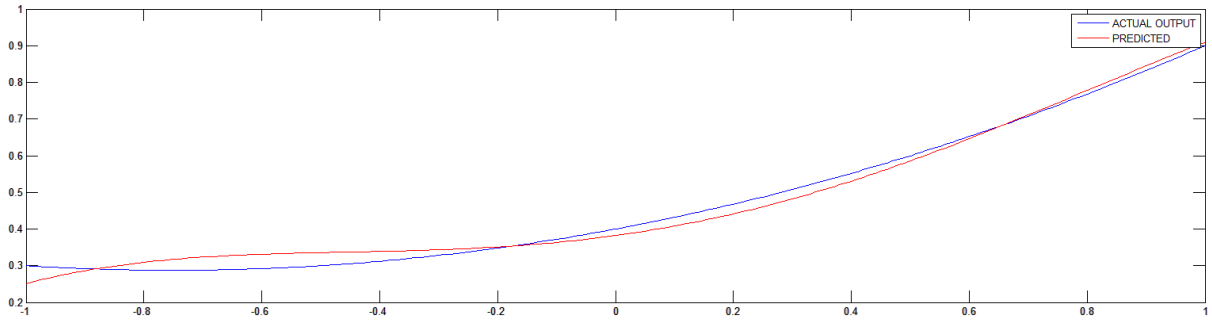

yTEST
yTESTPREDICT
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

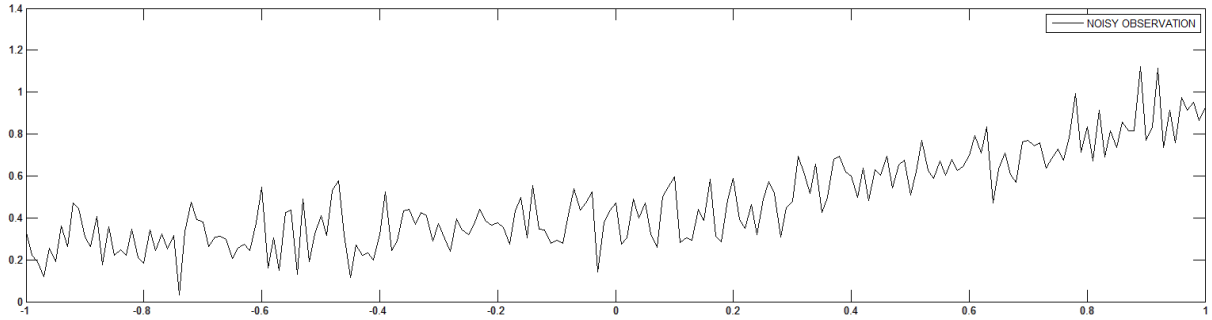

ACTUAL OUTPUT
PREDICTED
NOISY OBSERVATION
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

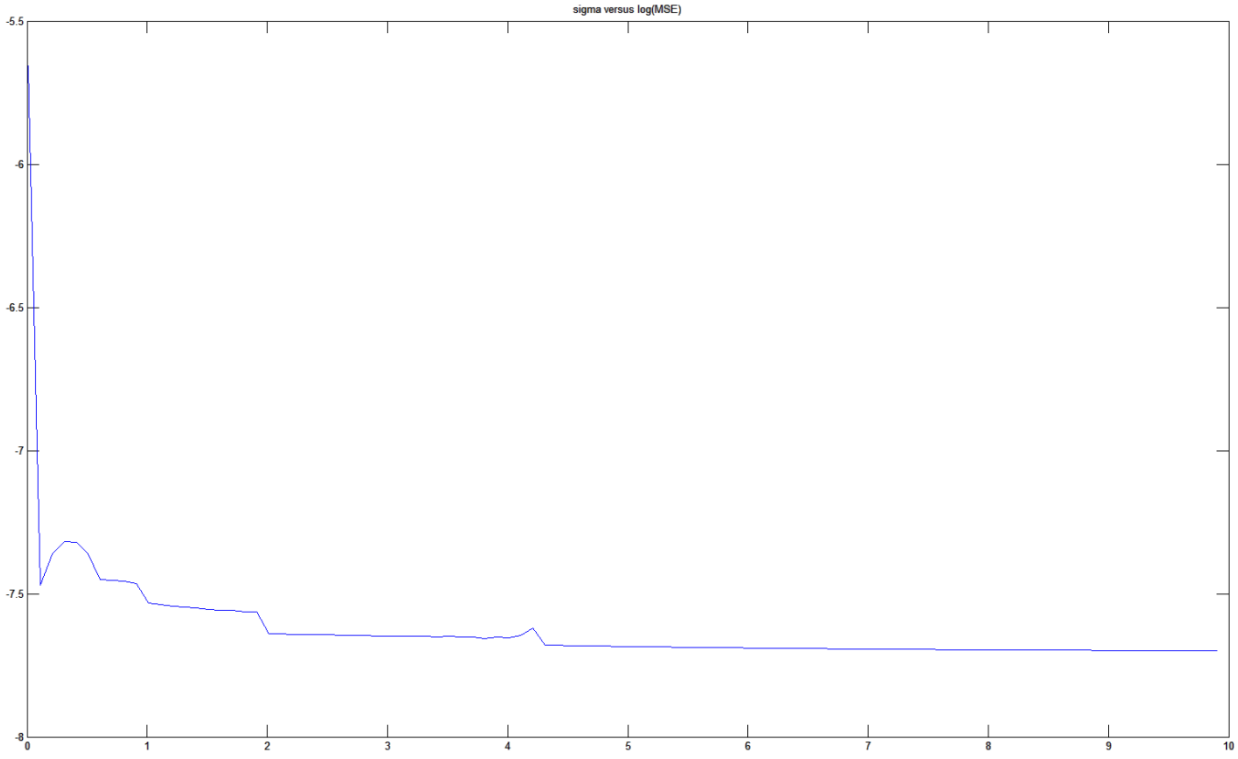
EXPERIMENT 9:GAUSSIAN KERNEL

```
sigma=0.01:0.1:10;
load REGDATA
t=t1;
y=y1;
%TRAIN-TEST SPLIT
L=length(x);
[p,q]=sort(rand(1,L));
TRAINPOS=q(1:1:fix(L/2));
TESTPOS=q(fix(L/2)+1:1:L);
%Least square solution using Linear regression with
polynomial regression
%Constructing PHI matrix
xTRAIN=x(TRAINPOS);
tTRAIN=t(TRAINPOS);
xTEST=x(TESTPOS);
yTEST=y(TESTPOS);
MSE=[];
for i=1:1:length(sigma)
PHI=[ones(1,length(TRAINPOS)); exp(-(xTRAIN-
0.1).^2/sigma(i)) ; exp(-(xTRAIN-0.2).^2/sigma(i))
;exp(-(xTRAIN-0.3).^2/sigma(i)) ;exp(-(xTRAIN-
0.4).^2/sigma(i)) ;...
    exp(-(xTRAIN-0.5).^2/sigma(i));exp(-(xTRAIN-
0.6).^2/sigma(i));exp(-(xTRAIN-0.7).^2/sigma(i)) ;exp(-
(xTRAIN-0.8).^2/sigma(i));exp(-(xTRAIN-
0.9).^2/sigma(i))]';
w=pinv(PHI'*PHI)*PHI'*tTRAIN';
weight{i}=w;
%FIT using
yTESTPREDICT=w(1)+w(2)*exp(-(xTEST-
0.1).^2/sigma(i))+w(3)*exp(-(xTEST-
0.2).^2/sigma(i))+w(4)*exp(-(xTEST-
0.3).^2/sigma(i))+w(5)*exp(-(xTEST-0.4).^2/sigma(i))...
    +w(6)*exp(-(xTEST-0.5).^2/sigma(i))+w(7)*exp(-
(xTEST-0.6).^2/sigma(i))+w(8)*exp(-(xTEST-
0.7).^2/sigma(i))+w(9)*exp(-(xTEST-
0.8).^2/sigma(i))+w(10)*exp(-(xTEST-0.9).^2/sigma(i));
MSE=[MSE sum((yTESTPREDICT-yTEST).^2)/length(yTEST)];
end
[p,q]=min(MSE);
w=weight{q};
```

```
yPREDICT=w(1)+w(2)*exp(-(x-0.1).^2/sigma(q))+w(3)*exp(-
(x-0.2).^2/sigma(q))+w(4)*exp(-(x-
0.3).^2/sigma(q))+w(5)*exp(-(x-0.4).^2/sigma(q))...
    +w(6)*exp(-(x-0.5).^2/sigma(q))+w(7)*exp(-(x-
0.6).^2/sigma(q))+w(8)*exp(-(x-
0.7).^2/sigma(q))+w(9)*exp(-(x-
0.8).^2/sigma(q))+w(10)*exp(-(x-0.9).^2/sigma(q));
figure
subplot(2,1,1)
plot(x,y)
hold on
plot(x,yPREDICT,'r')
legend('ACTUAL OUTPUT','PREDICTED');
subplot(2,1,2)
plot(x,t,'k')
legend('NOISY OBSERVATION')
figure
plot(sigma,log(MSE))
title('sigma versus log(MSE)')
```


ACTUAL OUTPUT
PREDICTED

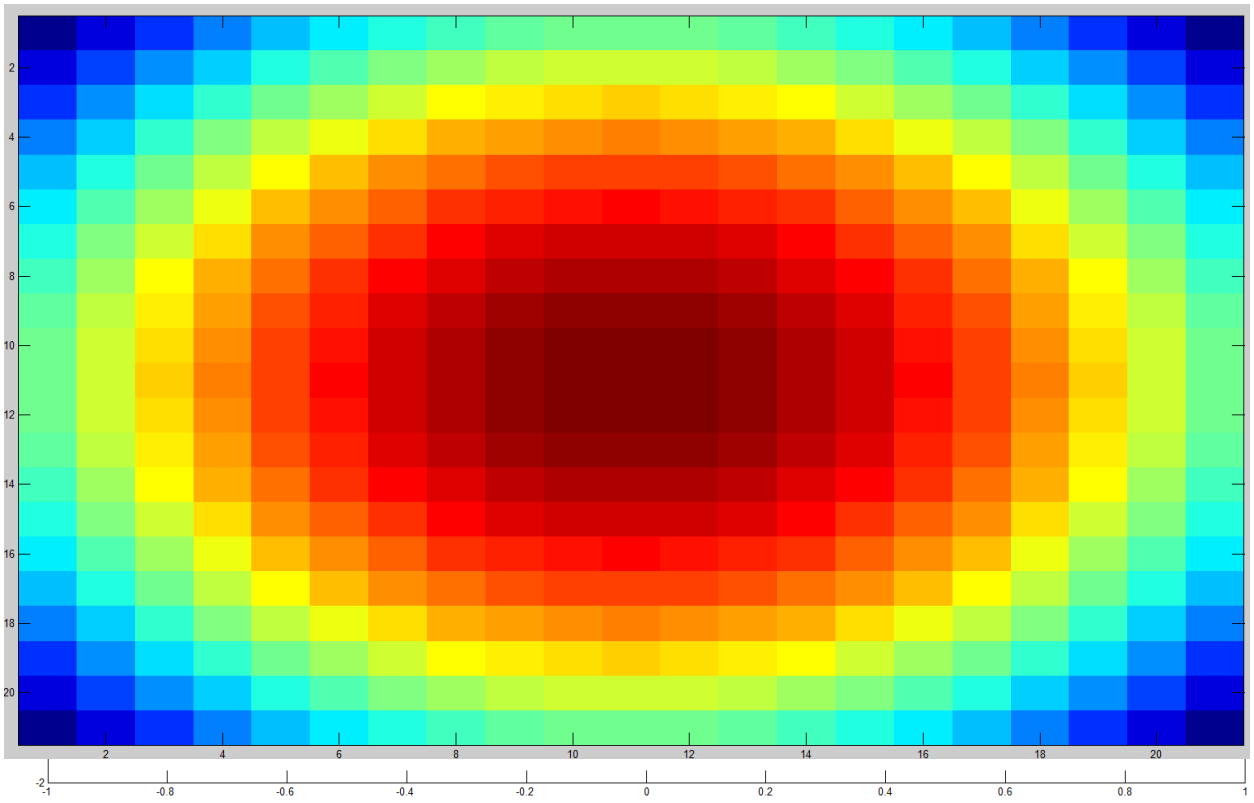

NOISY OBSERVATION


sigma versus log(MSE)

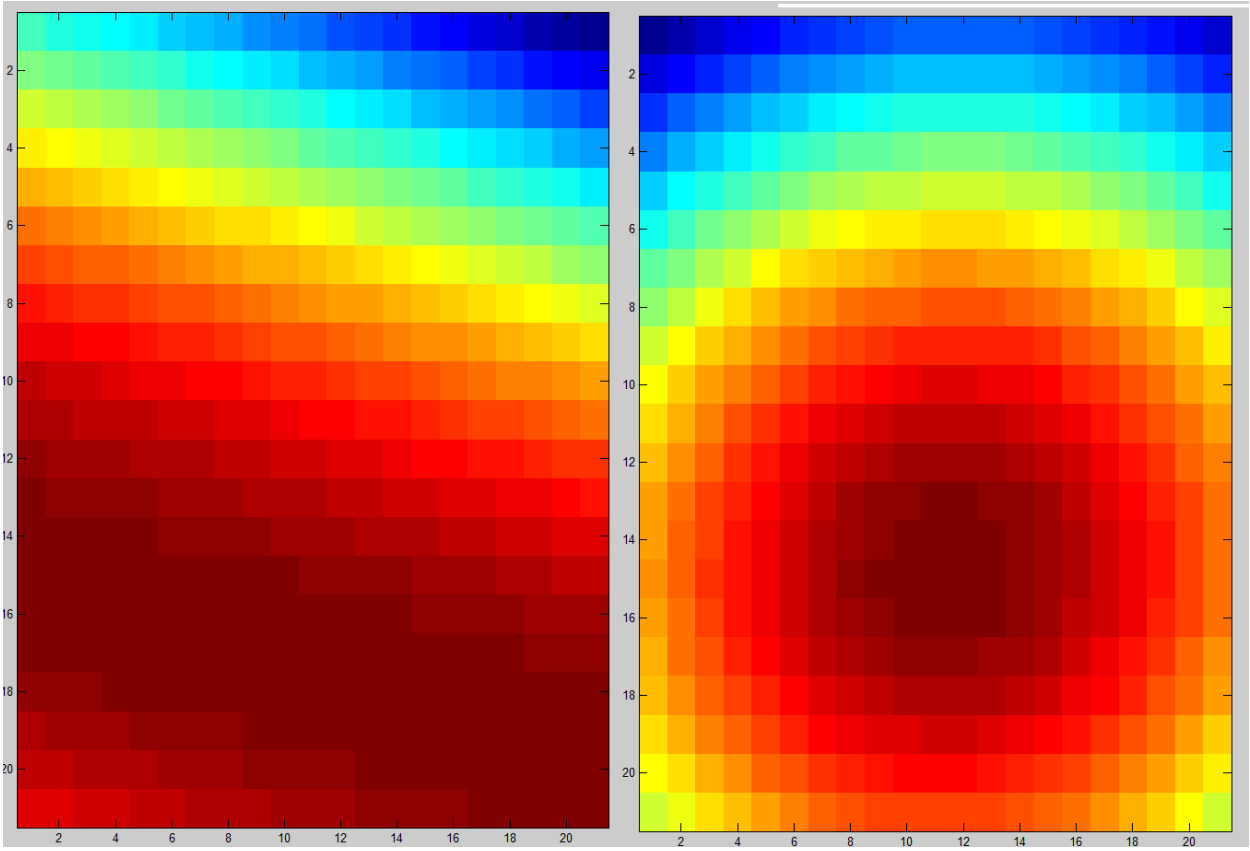
EXPERIMENT 9: BAYES REGRESSION MODEL

```
%Demonstrating Bayes technique
%Data generation
x=-1:0.01:1;
y=0.3*x+0.4;
t=y+randn(1,length(x))*sqrt(0.1);
plot(x,t)
%Linear regression model y=w(1)+w(2) x
%Prior density function of [w1 w2] with mean zero and
co-variance matrix
%0.2 I, I is the Identity matrix
for i=1:1:10
w=randn(1,2)*sqrt(0.2);
ypred=w(1)*x+w(2)
plot(x,ypred)
hold on
end
wrange=-1:0.1:1;
%Prior density plot
for u=1:1:length(wrange)
for v=1:1:length(wrange)
PRIOR(u,v)=(1/sqrt(2*pi*0.1))*exp(-
(wrange(u)^2+wrange(v)^2)/2*0.2);
end
end
figure(1)
imagesc(PRIOR)
POSTERIOR=PRIOR;
[p,q]=sort(rand(1,length(x)));
%Likelihood function based on the first observation
for d=1:1:length(x)
for u=1:1:length(wrange)
for v=1:1:length(wrange)
LIKELIHOOD(u,v)=(1/sqrt(2*pi*0.1))*exp(-(t(q(d))-
wrange(u)-wrange(v)*x(q(d)))^2/2*0.1);
end
end
POSTERIOR=POSTERIOR.*LIKELIHOOD;
POSTERIOR=POSTERIOR/sum(sum(POSTERIOR));
[wpred]=pickoutbest(POSTERIOR);
figure(2)
hold off
```

```
plot(x,y,'r')
hold on
ypred=wpred(1)*x+wpred(2);
plot(x,ypred,'k')

xlim([-1 1])
ylim([-2 2])
pause(0.3)
figure(3)
subplot(1,2,1)
imagesc(LIKELIHOOD)
subplot(1,2,2)
imagesc(POSTERIOR)
pause(5)
end
```



Prior density of the vector w

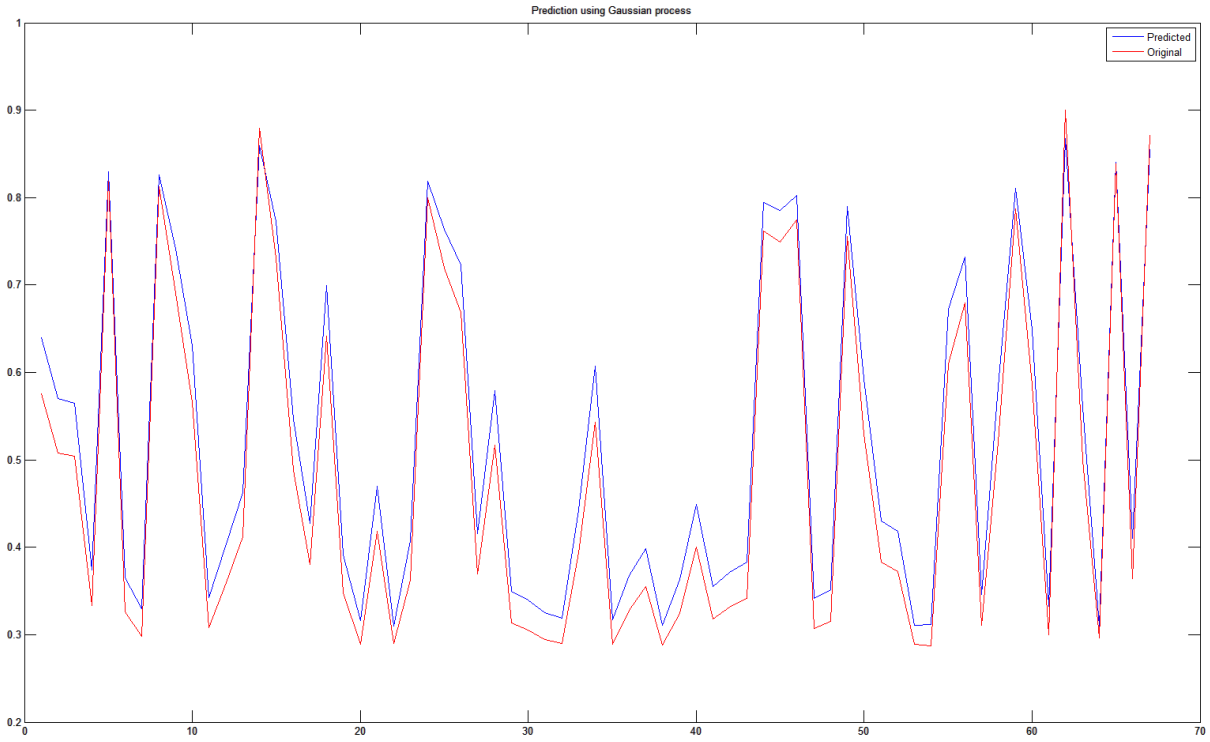


Likelihood and Aposterior density function of the vector w

EXPERIMENT 10:Gaussian smoothing

```
load REGDATA
t=t1;
y=y1;
%TRAIN-TEST SPLIT
L=length(x);
[p,q]=sort(rand(1,L));
TRAINPOS=q(1:1:fix(L/3));
TESTPOS=q(fix(L/3)+1:1:fix(2*L/3));
VALPOS=q(fix(2*L/3)+1:1:L);
%Least square solution using Linear regression with
polynomial regression
%Constructing PHI matrix
xTRAIN=x(TRAINPOS);
tTRAIN=t(TRAINPOS);
xTEST=x(TESTPOS);
yTEST=y(TESTPOS);
```

```
tTEST=t(TESTPOS);
xVAL=x(TESTPOS);
yVAL=y(TESTPOS);
COL=[];
sigmalist=[0.1:0.1:1 2:1:10 11:10:99 100:100:1000];
for s=1:1:length(sigmalist)
sigma=sigmalist(s)
    %Predicting values corresponding to xVAL pr
error=0;
for i=1:1:length(TESTPOS)
    COEF=[];
    for j=1:1:length(TRAINPOS)
   COEF=[COEF gausskernel(xTRAIN(j),xTEST(i),sigma)];
    end
    COEF=COEF/sum(COEF);
    temp=sum(COEF.*tTRAIN);
    error=error+(temp-tTEST(i))^2;
end
COL=[COL error];
end
[p,q]=min(COL);
optsigma=sigmalist(q);
%Validating the constructed regression
RES=[];
for i=1:1:length(xVAL)
RES=[RES
predict_using_gausskernel(xTRAIN,tTRAIN,xVAL(i),optsigm
a)];
end
plot(RES)
hold on
plot(yVAL,'r')
legend('Predicted','Original')
title('Prediction using Gaussian smoothing')
```

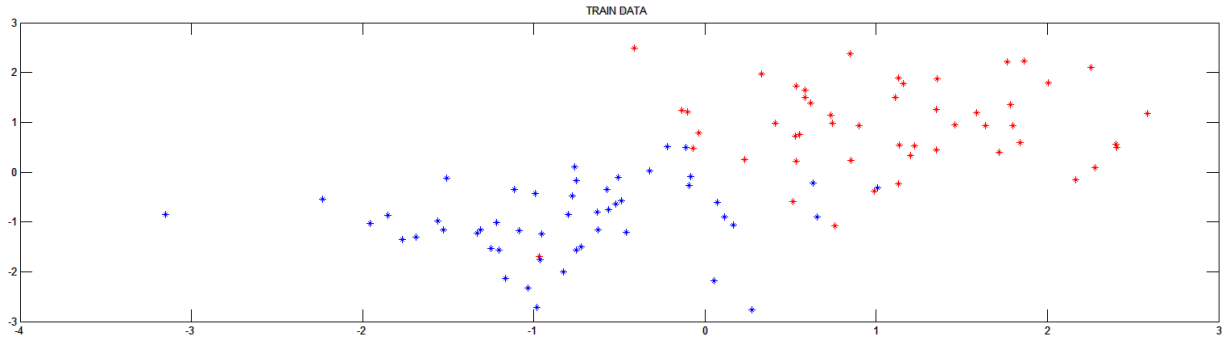


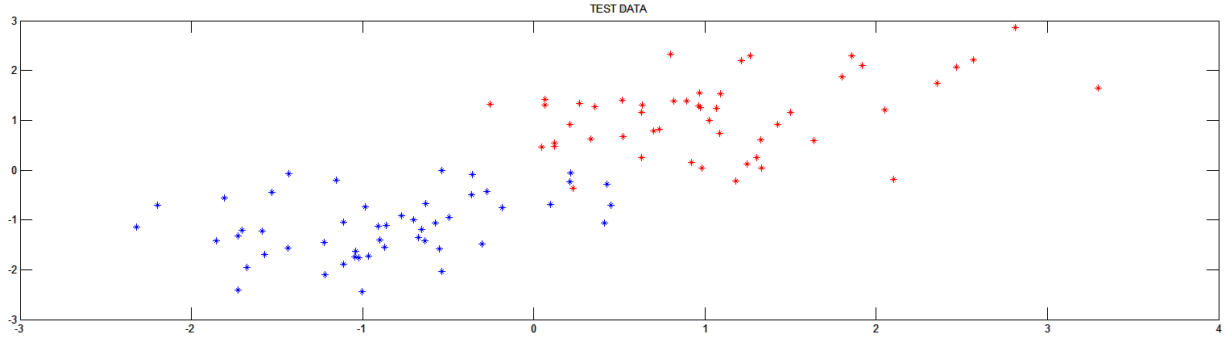
EXPERIMENT 11: ARTIFICIAL NEURAL NETWORK

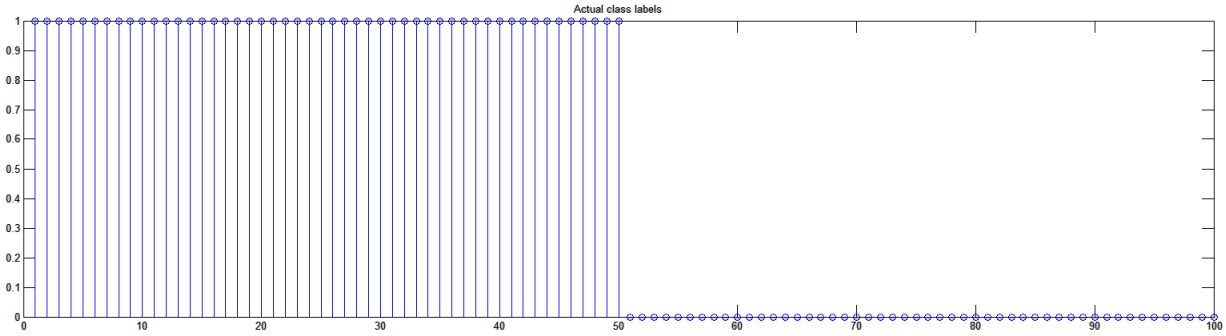
```
close all
clear all
M1=[1 1];
CX1=[0.6 0.1;0.1 0.6];
[E,diag]=eig(CX1);
X=randn(2,100);
X1=E*diag^(1/2)*X+repmat(M1',1,length(X));
%CLASS 2
M2=[-1 -1];
CX2=[0.7 0.1;0.1 0.7];
[E,diag]=eig(CX1);
X=randn(2,100);
X2=E*diag^(1/2)*X+repmat(M2',1,length(X));
t=[ones(1,50) zeros(1,50)];
%Initialize w

[p,q]=sort(rand(1,100));
TRAINDATA1=X1(:,q(1:1:50));
TRAINDATA2=X2(:,q(1:1:50));
TESTDATA1=X1(:,q(51:1:100));
```

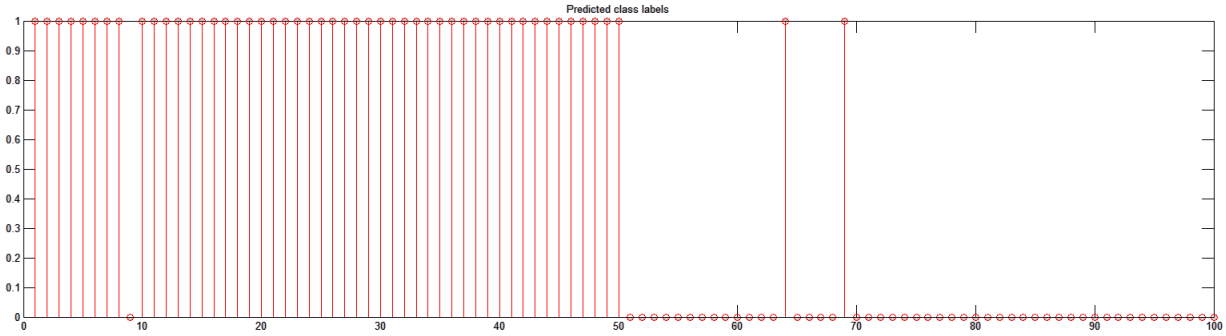
```
TESTDATA2=X2(:,q(51:1:100));
TRAINDATA=[TRAINDATA1 TRAINDATA2];
TESTDATA=[TESTDATA1 TESTDATA2];
figure(1)
subplot(2,1,1)
plot(TRAINDATA1(1,:),TRAINDATA1(2,:),'r*')
title('TRAIN DATA ')
hold on
plot(TRAINDATA2(1,:),TRAINDATA2(2,:),'b*')

subplot(2,1,2)
plot(TESTDATA1(1,:),TESTDATA1(2,:),'r*')
title('TEST DATA')
hold on
plot(TESTDATA2(1,:),TESTDATA2(2,:),'b*')



t=[ones(1,50) zeros(1,50)];
% Constructing ANN with 2X2X1

%Initializing the weights
W=rand(3,2);
V=rand(3,1);
S=[];
eta=0.01;
for iteration=1:1:100
y=logsig(W(1:2,:)'*TRAINDATA+repmat(W(3,:)',1,100));
z=V(1:2,:)'*y+repmat(V(3,1)',1,100);
error=t-z;
V(1,1)=V(1,1)+eta*sum((error.*y(1,:)));
V(2,1)=V(2,1)+eta*sum((error.*y(2,:)));
V(3,1)=V(3,1)+eta*sum((error));
W(1,1)=W(1,1)+eta*sum((error.*y(1,:).*(1-
y(1,:)).*TRAINDATA(1,:)));
W(1,2)=W(1,2)+eta*sum((error.*y(2,:).*(1-
y(2,:)).*TRAINDATA(1,:)));
W(2,1)=W(2,1)+eta*sum((error.*y(1,:).*(1-
y(1,:)).*TRAINDATA(2,:)));
W(2,2)=W(2,2)+eta*sum((error.*y(2,:).*(1-
y(2,:)).*TRAINDATA(2,:)));
W(3,1)=W(3,1)+eta*sum((error.*y(1,:).*(1-y(1,:))));
W(3,2)=W(3,2)+eta*sum((error.*y(2,:).*(1-y(2,:))));
```

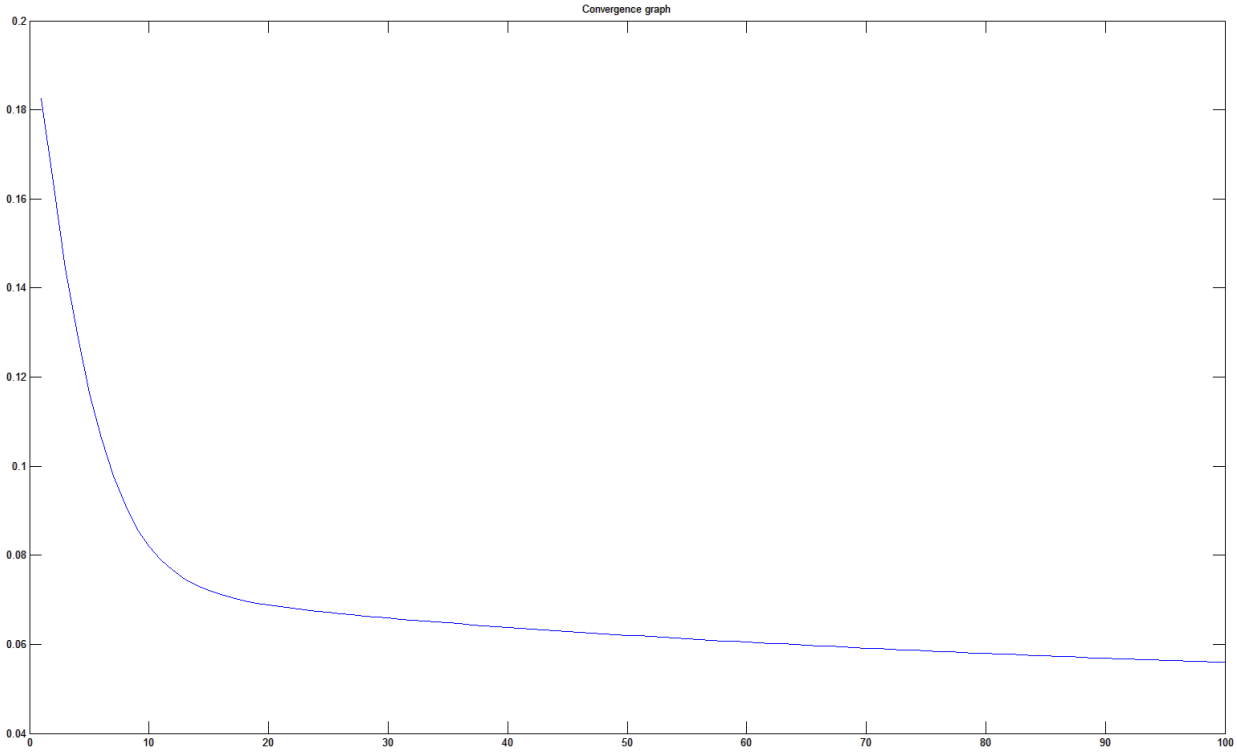
```
S=[S sum(error.^2)/100];
end
INDEX=[];
y=logsig(W(1:2,:)'*TESTDATA+repmat(W(3,:)',1,100));
z=V(1:2,:)'*y+repmat(V(3,1)',1,100);
for i=1:1:length(y)
if(z(i)>0.5)
    INDEX=[INDEX 1];
else
    INDEX=[INDEX 0];
end
end
[p,q]=find(INDEX==t);
POS=length(p);
figure(2)
subplot(2,1,1)
stem(t,'b')
title('Actual class labels')
subplot(2,1,2)
stem(INDEX,'r')
title('Predicted class labels')
figure(3)
plot(S)
title('Convergence graph')
```






Actual class labels
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100


Predicted class labels
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100


Convergence graph
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0
10
20
30
40
50
60
70
80
90
100