ECE 559 NEURAL NETWORKS-HOMEWORK 7

NAME:LAKSHMI SRIDEVI UIN:668383906

The given problem was solved using MATLAB and as given in the source code. The Kernel used to solve this problem is a Gaussian Kernel which is given by



**MATLAB SOURCE CODE**

clear all

clc

close all

**%a)&b)Plotting the points based on their respective positive and negative**

**%classes**

X=1.\*rand(2,100);

hold on

for i=1:100

a=(1/5)\*sin(10\*X(1,i))+0.3;

b=(X(2,i)-0.8)^2+(X(1,i)-0.5)^2;

if (X(2,i)<a || b<0.15^2)

d(1,i)=1;

C1(:,i)=X(:,i);

plot(C1(1,i),C1(2,i),'\*r')

else

d(1,i)=-1;

C2(:,i)=X(:,i);

plot(C2(1,i),C2(2,i),'sqb')

end

end

xlabel('x1')

ylabel('x2')

**%c)Choosing a Gaussian Kernel**

for j=1:100

for i=1:100

H(i,j)=(d(1,i)\*d(1,j)\*(exp(-4\*(X(:,i)-X(:,j))'\*(X(:,i)-X(:,j)))));

Kernel(i,j)=(exp(-4\*(X(:,i)-X(:,j))'\*(X(:,i)-X(:,j))));

end

end

f=-1\*ones(100,1);

A=[];

b=[];

lb=zeros(100,1);

ub=inf\*ones(100,1);

**%finding the values of alpha using quadratic optimization technique**

alpha=quadprog(H,f,A,b,d,0,lb,ub);

m=0;

l=alpha>10^(-4);

k=find(l);

**%Plotting the support vectors**

hold on

for i=1:length(k)

plot(X(1,k(i,1)),X(2,k(i,1)),'Ok')

end

for i=1:length(k)

temp1=0;

for j=1:100

temp=alpha(j,1)\*d(1,j)\*Kernel(j,k(i));

temp1=temp1+temp;

end

temp2(i)=d(1,k(i))-temp1;

end

theta=sum(temp2)/length(k);

**%Storing the co-ordinates of the Support Vectors**

for i=1:length(k)

SV(:,i)=X(:,k(i));

end

**%Testing the model with test points to check if**

**%SVM is able to perfectly separate the two classes with no exceptions of**

**%misclassied input patterns**

x1=0.01:0.01:1;

x2=0.01:0.01:1;

v=0;

for i=1:100

for j=1:100

v=v+1;

test(:,v)=vertcat(x1(i),x2(j));

end

end

for j=1:10000

for i=1:length(k)

Kernel1(i,j)=(exp(-4\*(X(:,k(i))-test(:,j))'\*(X(:,k(i))-test(:,j))));

end

end

temp3=0;

temp4=0;

for j=1:length(test)

temp3=0;

for i=1:length(k)

temp4=alpha(k(i),1)\*d(1,k(i))\*Kernel1(i,j);

temp3=temp3+temp4;

end

g(j)=temp3+theta;

end

[y1,y2]=meshgrid(x1,x2);

z=vec2mat(g,100);

contour(y1,y2,z',[0 0],'g')

contour(y1,y2,z'-1,[0 0],'r')

contour(y1,y2,z'+1,[0 0],'b')

OUTPUT:

**Plot for data classification along with Decision Boundary**

