**Sayan Banerjee**

EE890: mini-project 5 report

# **SVM: Liner Kernel**

clear all;

load('SVM\_data.mat','x','y') ;

[m, n] = size(x);

figure()

plot(x(1:m/2,1),x(1:m/2,2),'+');

hold on

plot(x((m/2)+1:m,1),x((m/2)+1:m,2),'rO');

pbaspect([1 1 1])

target=y;

X\_T=x;

X=x';

Q=((y\*y').\*(X\_T\*X));

one=-1.\*(ones(40,1));

AD=[y';(-1).\*y';eye(40,40)];

zero=zeros(42,1);

[alpha,fval] = quadprog(Q,one,-AD,zero,[],[]);

for i=1:40

if alpha(i,1)<0.00001

alpha(i,1)=0.0;

end

end

W=double(zeros(1,2));

for i= 1:40

W=W+(y(i,1)\*alpha(i,1))\*x(i,:);

end

b=y(3,1)-W\*x(3,:)';

plot(x(3,1),x(3,2), '-.k\*');

plot(x(26,1),x(26,2), '-.k\*');

plot(x(27,1),x(27,2), '-.k\*');

x = linspace(-4,4);

w1=W(1,1);

w2=W(1,2);

y=(-b-w1\*x)/w2;

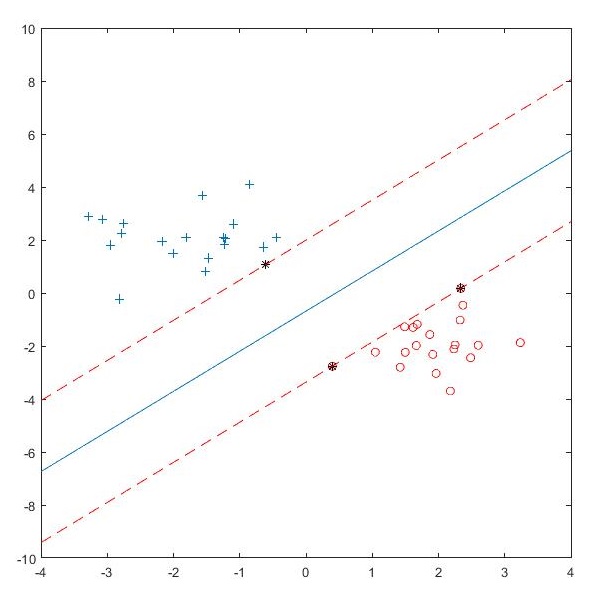
line(x,y)

y=(1-b-w1\*x)/w2;

line(x,y,'Color','red','LineStyle','--')

y=(-1-b-w1\*x)/w2;

line(x,y,'Color','red','LineStyle','--')



# **SVM Kernel Trick (Polynomial Degree 3)**

**Kernel: K=** (1+XT\*X)3

clear all ;

load('SVM\_data\_nonlinear.mat','x','y') ;

[m, n] = size(x);

figure()

plot(x(1:m/2,1),x(1:m/2,2),'+');

hold on

plot(x((m/2)+1:m,1),x((m/2)+1:m,2),'rO');

pbaspect([1 1 1])

X\_T=x;

X=x';

K=(X\_T\*X);

K=1+K;

K=K.^3;

Q=((y\*y').\*K);

one=-1.\*(ones(60,1));

AD=[y';(-1).\*y';eye(60,60)];

zero=zeros(62,1);

[alpha,fval] = quadprog(Q,one,-AD,zero,[],[]);

for i=1:60

if alpha(i,1)<0.00001

alpha(i,1)=0.0;

end

end

for i=1:60

if alpha(i,1)~=0

plot(x(i,1),x(i,2), '-.k\*');

end

end

SV=X(:,6);

bias=0.0;

for i=1:60

if alpha(i,1)~=0

K\_b=(X\_T(i,:)\*SV);

K\_b=1+K\_b;

K\_b=K\_b^3;

bias=bias+(alpha(i,1)\*y(i,1)\*K\_b);

end

end

length=1;

boundary=zeros(100,2);

while length<101

random\_x1=(3-(-5)).\*rand(1,1)+(-5);

random\_x2=(2-(-3)).\*rand(1,1)+(-3);

p=[random\_x1;random\_x2];

g=0.0;

for i=1:60

if alpha(i,1)~=0

K\_p=(X\_T(i,:)\*p);

K\_p=1+K\_p;

K\_p=K\_p.^3;

g=g+(alpha(i,1)\*y(i,1)\*K\_p);

end

end

g=g+bias;

if abs(g)<0.00001

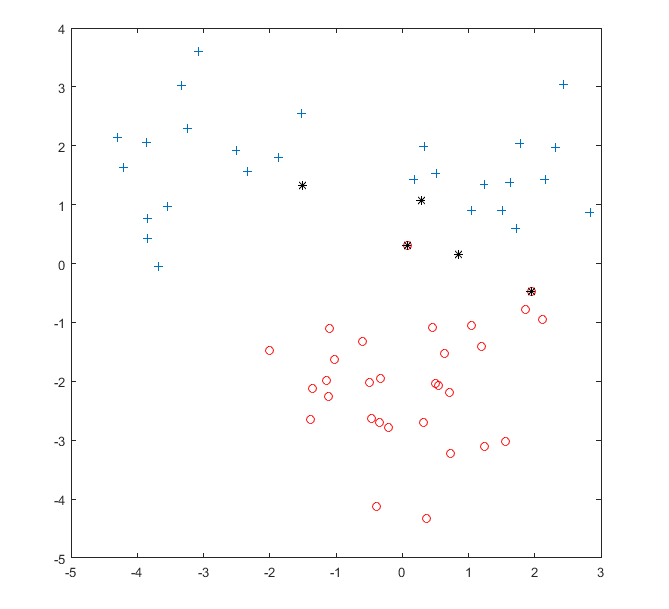
boundary(length,1)=random\_x1;

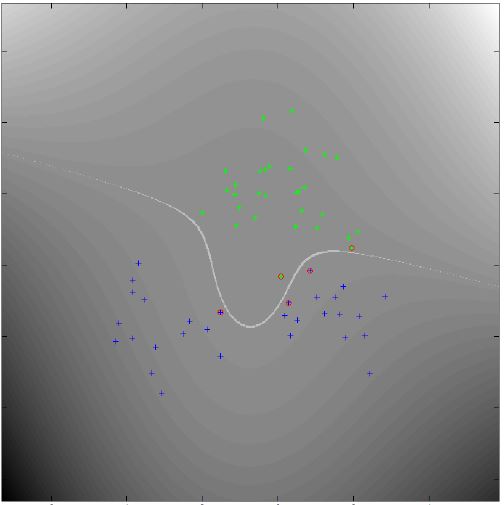
boundary(length,2)=random\_x2;

length=length+1;

end

end





# **SVM Kernel Trick (Polynomial Degree 3)**

**Kernel: K=** Φ(X)T\* Φ(X)

Φ(X)= 1, x1, x2, x2, x1x2, x22, x13, x12x2, x1x22, x23

clear all ;

clc;

clear;

load('SVM\_data\_nonlinear.mat','x','y') ;

[m, n] = size(x);

figure()

plot(x(1:m/2,1),x(1:m/2,2),'+');

hold on

plot(x((m/2)+1:m,1),x((m/2)+1:m,2),'rO');

pbaspect([1 1 1])

K\_X=zeros(m,10);

for i=1:m

K\_X(i,1)= 1;

K\_X(i,2)= x(i,1);

K\_X(i,3)= x(i,2);

K\_X(i,4)= x(i,1)^2;

K\_X(i,5)= x(i,1)\*x(i,2);

K\_X(i,6)= x(i,2)^2;

K\_X(i,7)= x(i,1)^3;

K\_X(i,8)= x(i,1)^2\*x(i,2);

K\_X(i,9)= x(i,1)\*x(i,2)^2;

K\_X(i,10)= x(i,2)^3;

end

X\_T=K\_X;

X=K\_X';

Q=((y\*y').\*(X\_T\*X));

one=-1.\*(ones(60,1));

AD=[y';(-1).\*y';eye(60,60)];

zero=zeros(62,1);

[alpha,fval] = quadprog(Q,one,-AD,zero,[],[]);

for i=1:60

if alpha(i,1)<0.00001

alpha(i,1)=0.0;

end

end

for i=1:60

if alpha(i,1)~=0

plot(x(i,1),x(i,2), '-.k\*');

end

end

W=double(zeros(1,10));

for i= 1:60

W=W+(y(i,1)\*alpha(i,1))\*K\_X(i,:);

end

bias=y(6,1)-W\*K\_X(6,:)';

w1=W(1);

w2=W(2);

w3=W(3);

w4=W(4);

w5=W(5);

w6=W(6);

w7=W(7);

w8=W(8);

w9=W(9);

w10=W(10);

min=-5;

max=3;

random=(max-min).\*rand(1000,1)+min;

random\_y=zeros(1000,1);

syms y

for i=1:1000

eqn=w1+w2\*random(i)+w3\*y+w4\*random(i)^2+w5\*random(i)\*y+w6\*y^2+w7\*random(i)^3+w8\*random(i)^2\*y+w9\*random(i)\*y^2+w10\*y^3+bias==0;

soly=solve(eqn,y);

soly=double(soly);

[m,n]=size(soly);

for j=1:m

p=real(soly(j,n));

%q=soly(j,n).real;

if p>=-5 && p<=4

random\_y(i,1)=p;

end

break;

end

end

for i=1:1000

plot(random(i,1),random\_y(i,1), '+');

end

