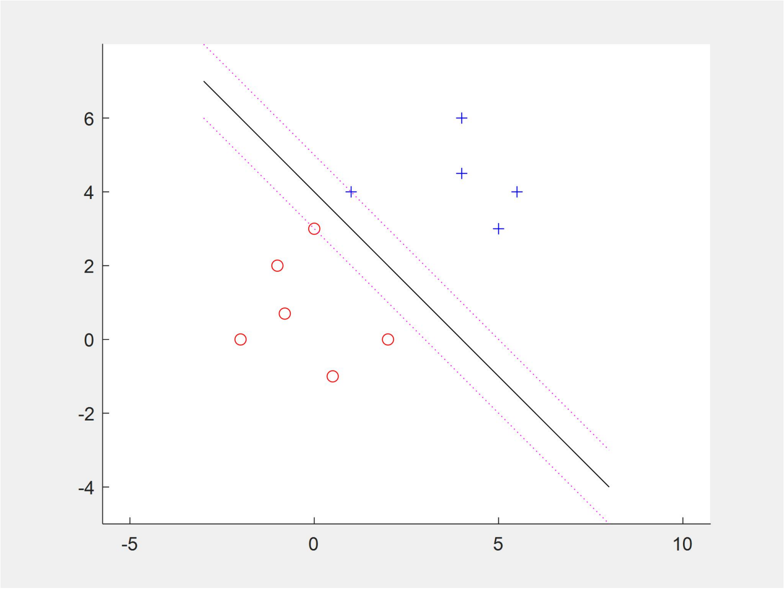
MATH368 Project2 Group27

T1

The code for T1 svm function :

```
%T1 svm function
function [w,beta] = svm(trainingpoints, traininglabels)
%trainingpoints: d*m
%traininglabels: m-dimensional row-vector labels
    numOfAttributes=size(trainingpoints,1);
    numOfExamples=size(trainingpoints,2);
    X=trainingpoints';
    Y=traininglabels';
    H=eye(numOfAttributes+1);
    H(numOfAttributes+1,numOfAttributes+1)=0;
    f=zeros(numOfAttributes+1,1);

    Z = [X ones(numOfExamples,1)];
    A=-diag(Y)*Z;
    c=-1*ones(numOfExamples,1);

    W=quadprog(H,f,A,c);

    w=W(1:numOfAttributes);
    beta=W(numOfAttributes+1);

end
```

T2



The code for T2 :

```
% T2 apply svm to the data
clc;
clear;
X=[-1,-0.8,0.5,-2,2,0,1,4,5,4,5.5;2,0.7,-1,0,0,3,4,6,
3,4.5,4.0];

Y=[-1,-1,-1,-1,-1,-1,1,1,1,1,1];
[w,beta]=svm(X,Y);
disp(w);
disp(beta);

% T2 visualization of the results
XX=X';
YY=Y';
ClassA = XX(find(YY==-1),:);
ClassB = XX(find(YY==1),:) ;
hold on
% Plot the data with label -1
plot(ClassA(:,1),ClassA(:,2),'or');
% Plot the data with label -2
plot(ClassB(:,1),ClassB(:,2),'+b') ;

% Plot the hyperplane
hold on
X1=[-3:8];
w1=w(1);
w2=w(2);
b=beta;
Y1=-(w1*X1+b)/w2;
plot(X1,Y1,'k-');
% Plot the SVM margins
Yup=(1-w1*X1-b)/w2; %Margin
plot(X1,Yup,'m:');
Ylow=(-1-w1*X1-b)/w2; %Margin
plot(X1,Ylow,'m:');

axis equal;
```

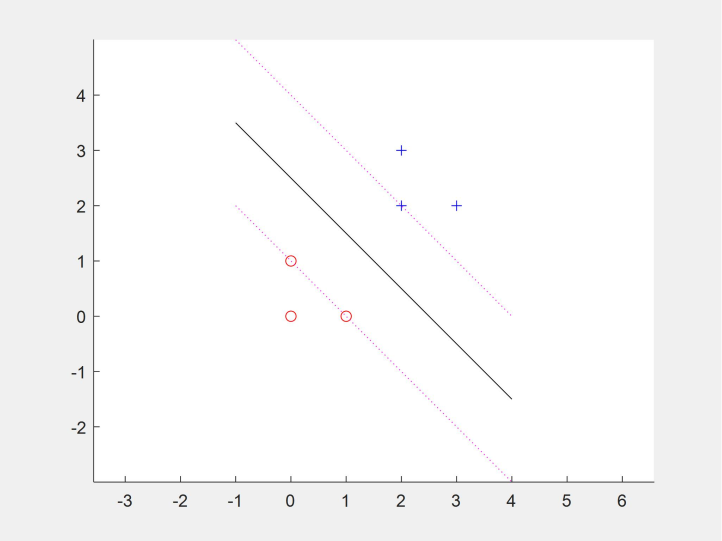
T3

It is **false** that hyperplane would be always perpendicular to the line connecting the two closet points from both label classes.

Counterexample:

A(0,0) B(0,1) C(1,0) D(2,2) E(2,3) F(3,2)

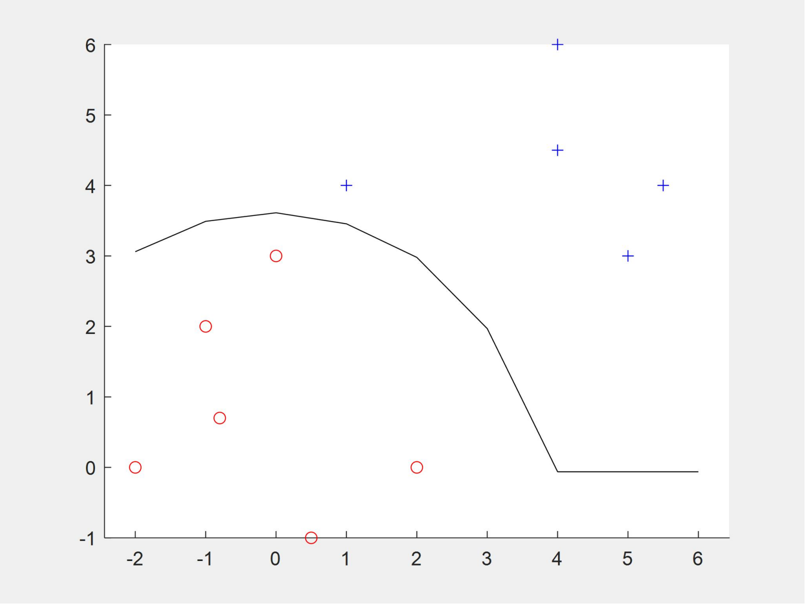
A,B,C are labeled as '-1', D,E,F are labeled as '1'



Clearly the closest two points from two classes are B(0,1) to D(2,2). However if we use svm in linear cases we would obtain a hyperplane with slope -1. The slope of line BD is not 1 obviously. Hence hyperplane and BD are not perpendicular.

T4

A two-dimensional plot of the 11 points colored according to the classification in three dimensions:



```
>> w1

w1 =

    0.0303

>> w2

w2 =

    0.0303

>> w3

w3 =

    0.2424
```

Above is the values of parameters obtained. As we can see from the graph, the decision boundary is **not linear**.

The code for T4 :

```
% T2 apply svm to the data
clc;
clear;
X=[-1,-0.8,0.5,-2,2,0,1,4,5,4,5.5;2,0.7,-1,0,0,3,4,6,
3,4.5,4.0];
Y=[-1,-1,-1,-1,-1,-1,1,1,1,1,1];

% T2 visualization of the results
XX=X;
XX=[XX;XX(1,:).^2+XX(2,:).^2]
YY=Y;




[w,beta]=svm(XX,YY);
disp(w)
disp(beta)
%%

% Now we generate 250 points and classify them using the
results above
x1=-2+(6-(-2))*rand(1,250);
x2=-2+(6-(-2))*rand(1,250);
```

```
Xsquare=x1(1,:).^2+x2(1,:).^2;

Xcombined=[x1;x2;Xsquare];
result=w'*Xcombined+beta;
predict=[];
for i=1:250
    if result(1,i)>=0
        predict(i)=1;
    else predict(i)=-1;
    end
end
disp(predict)
%%

% A two-dimension plot of the 11 points prediction
Yresult=w'*XX+beta;
Ypredict=[]
for i=1:size(Y,2)
    if Yresult(1,i)>=0
        Ypredict(i)=1;
    else Ypredict(i)=-1;
    end
end


XXnew=X'
YYnew=Ypredict'
ClassA = XXnew(find(YYnew==-1),:)
ClassB = XXnew(find(YYnew==1),:)
hold on
% Plot the data with label -1
plot(ClassA(:,1),ClassA(:,2),'or')
% Plot the data with label -2
plot(ClassB(:,1),ClassB(:,2),'+b')
% Plot the hyperplane
hold on
Xs=[-2:6]
w1=w(1);
w2=w(2);
w3=w(3);
b=beta;
Ys=(-w2+sqrt(w2^2-4*w3*(w3*Xs.^2+w1*Xs+b)))/(2*w3);
plot(Xs,Ys,'k-');
axis equal;
```

T5

Result:
Tonny_Blair_0081.pgm to Tonny_Blair_0089.pgm are correctly classified.
Only Tonny_Blair_0090.pgm is misclassified.

The code for T5:

```
clear;
clc;
%set train path
train_path='.\trainset\';
train_list = dir(strcat(train_path,'*.pgm'));
train_num = length(train_list);
train_mat=[];
%loop through trainset and obtain train_mat
for j=1:train_num
    train_name=train_list(j).name;
    train_img=imread(strcat(train_path,train_name));
    train_col=reshape(im2double(train_img),[],1);
    train_mat=[train_mat train_col];
end
train_label=[]
for k=1:30
    train_label=[train_label,-1];
end
for k=1:30
    train_label=[train_label,1];
end
disp(train_label);

%set test path
test_path='.\testset\';
test_list = dir(strcat(test_path,'*.pgm'));
test_num = length(test_list);
test_mat=[];
%loop through trainset and obtain train_mat
for j=1:test_num
    test_name=test_list(j).name;
    test_img=imread(strcat(test_path,test_name));
    test_col=reshape(im2double(test_img),[],1);
    test_mat=[test_mat test_col];
end
%apply the svm function
[w,beta]=svm(train_mat,train_label);
```

```
disp(size(w));disp(beta);
%predict the labels of ten pics in testset
result=w'*test_mat+beta;
predict=[];
for i=1:test_num
    if result(1,i)>=0
        predict(i)=1;
    else predict(i)=-1;
        fprintf('The %f th picture is
misclassified',round(i));
    end
end
disp(predict);
```