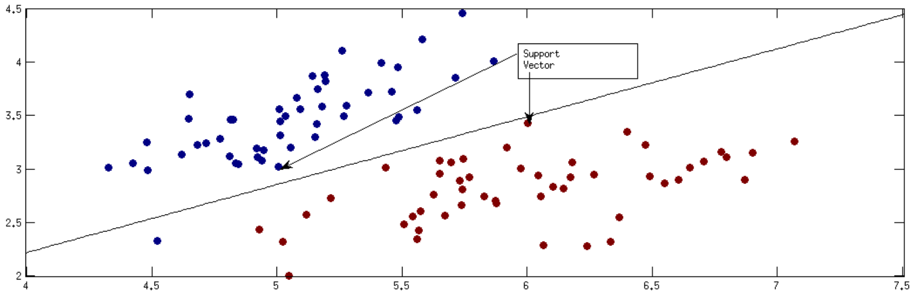
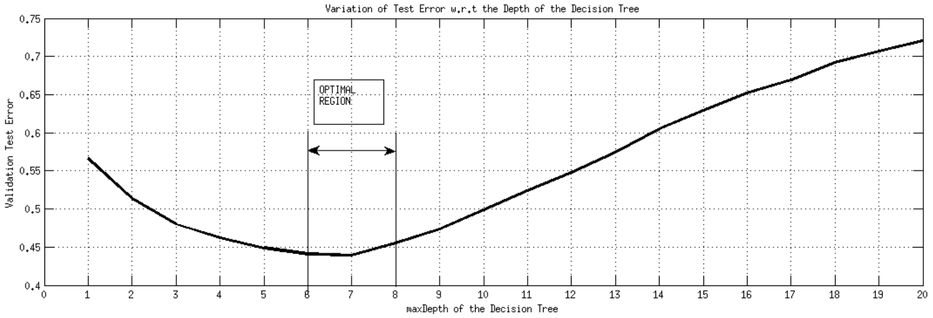
Univ of California Irvine **Debatri Mitra**
CS 273A-Machine Learning **Homework 4**
Prof. Alex Ihler **Jan 27,2015**

Problem-I

```
clear all;
iris=load('data/iris.txt');
X = iris(:,1:2);
Y=iris(:,end);
XA = X(Y<2,:);
YA=Y(Y<2);
YA(find(YA==0)) = -1;
gscatter(XA(:,1),XA(:,2),YA);

m = size(XA,1) ;
n = size(XA,2) ;

H=eye(n+1);
H(n+1,n+1)=0 ;

f=zeros(n+1,1);
Z = [XA ones(m,1)];
A=-diag(YA)*Z ;
c=-1*ones(m,1) ;
w=quadprog(H,f,A,c)

learner=logisticClassify(XA,YA);
learner=setWeights(learner, w');
learner=setClasses(learner, unique(YA));
%pc=setWeights(pc,w);
plotClassify2D(learner,XA,YA);
```

The parameters/ weights found from the SVM training are: [6.35721478443559,-5.36933236163779,-17.2696795465474] The learned Classifier is shown in the next fig.
Problem - III
3-a
3-b
Here I did a 5-fold cross validation over the training data. The max-Depth of the Decision Tree has been varied over 1 to 20 and mean squared error has been calculated on the validation test data as given in plot (1)
It looks like the optimal parameter for max-Depth of Decision Tree is somewhere between 6-8.
As the max-Depth increases the complexity of the learner increases.
For max-Depth=20 Error-MSE = 0.6948

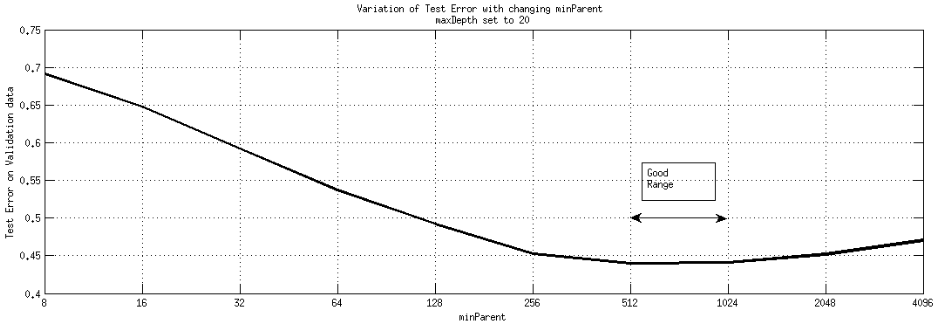




```
clear all;
X=load('data/kaggle.X1.train.txt');
Y=load('data/kaggle.Y.train.txt');

X_TEST=load('data/kaggle.X1.test.txt');
%[Xtr,Xte,Ytr,Yte] = crossValidate(X,Y,5,3);


%dt = treeRegress(Xtr,Ytr, 'maxDepth',20);

%Yte = predict(dt,Xte);
%e=mse(dt,Xte,Yte)

error=zeros(1,20);
for j=1:5
    [Xtr,Xte,Ytr,Yte] = splitData(X, Y, 0.6);
for i=1:20
  dt = treeRegress(Xtr,Ytr, 'maxDepth',i);
  %Yte = predict(dt,Xte);
  error(i)=error(i)+mse(dt,Xte,Yte);
end
end
error=error./5;
plot(error)
dt = treeRegress(X,Y, 'maxDepth',10,'minParent',1000);
```

```
Y_TEST=predict(dt,X_TEST);
```

3-c In this part keeping the max-Depth as max =20 we just vary the min-Parent parameter from 2^3 to 2^{12} and compare the mean squared error as shown in Plot (2). The Complexity decreases as the min-Parent increases. The optimal range would be around 512 as the test (validation) error is minimized.

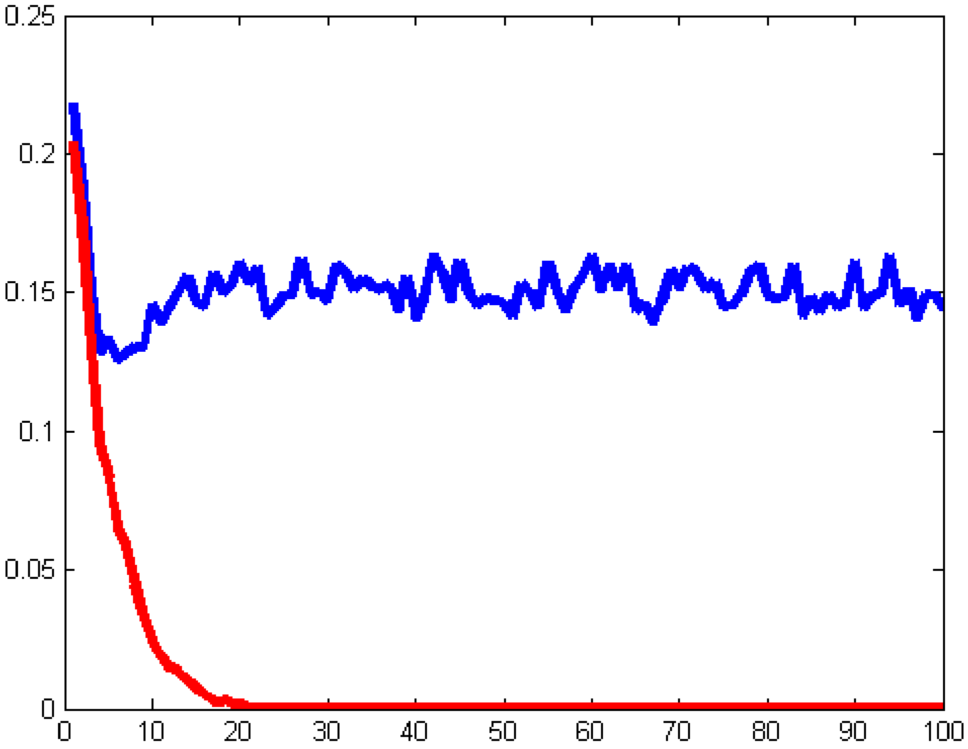


```
Error=zeros(1,10);
for j=1:5
    [X,Y]=shuffleData(X,Y);
    [Xtr,Xte,Ytr,Yte] = splitData(X, Y, 0.6);
for i=3:12
  dt = treeRegress(Xtr,Ytr, 'maxDepth',20,'minParent',2^i);
  %Yte = predict(dt,Xte);
  Error(i-2)=mse(dt,Xte,Yte);
end
end
Error=Error./5;
plot(Error)
```

3-d
Using these setting I got a 0.65807 accuracy in Kaggle. However, some more permutation and combination could easily come up with a more optimal solution.

1 Problem IV

```
 clear all;
X=load('data/kaggle.X1.train.txt');
Y=load('data/kaggle.Y.train.txt');
[Xtr,Xte,Ytr,Yte] = crossValidate(X,Y,5,3);
mu = mean(Ytr);

dY = Ytr - mu.*ones(size(Ytr,1),1);
Nboost=25;

 for k=1:25,
    Learner{k} = treeRegress(Xtr,dY,'maxDepth',3,'minParent',32);
    %alpha(k) = 1;
    dY = dY-  predict(Learner{k}, Xtr);
end

[Ntest,D0] = size(Xte);
[Ntrain,D]=size(Xtr);

prediction_tr = zeros(Ntrain,1);
prediction_te = zeros(Ntest,1);

%error=mse(Learner{5},Xte,Yte)
%j=[1,5,10,25];
error_train=zeros(4,1);
error_test=zeros(4,1);
i=1;
for j=[1,5,10,25]
    prediction_tr = zeros(Ntrain,1);
    prediction_te = zeros(Ntest,1);

    for k=1:j,
        prediction_tr = prediction_tr + predict(Learner{k}, Xtr);
    end;

    error_train(i) = mean( sum( (Ytr - prediction_tr).^2,2) );
    i=i+1;
end
i=1;
for  j=[1,5,10,25]
    prediction_tr = zeros(Ntrain,1);
    prediction_te = zeros(Ntest,1);
    for k=1:j,
        prediction_te = prediction_te + predict(Learner{k}, Xte);
    end;

    error_test(i) = mean( sum( (Yte - prediction_te).^2,2) );
    i=i+1;
end
plot(error_test)
hold on
plot(error_train)
```


0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
70
80
90
100