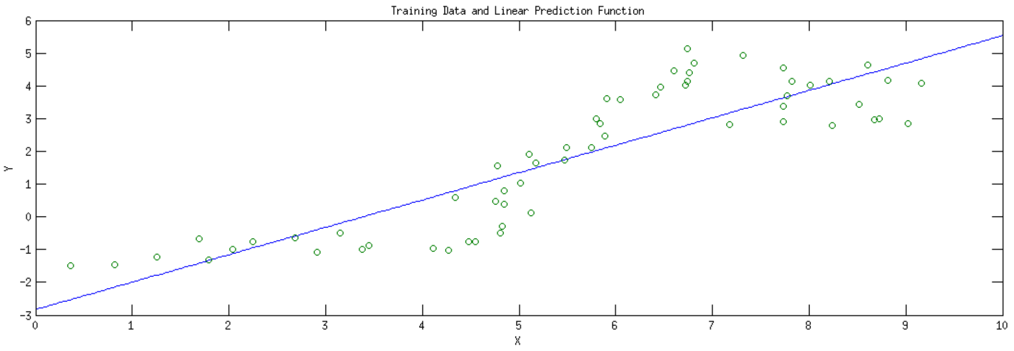
Univ of California Irvine | **Debatri Mitra**
CS 273A-Machine Learning | **Homework 2**
Prof. Alex Ihler | **Jan 20,2015**

Problem - I
1-a

```
 clear all;
DATA=load('data/curve80.txt');
% read data from text
y=DATA(:,end);
x=DATA(:,1);
[Xtr,Xte ,Ytr, Yte] = splitData(x,y, .75);
```

1-b

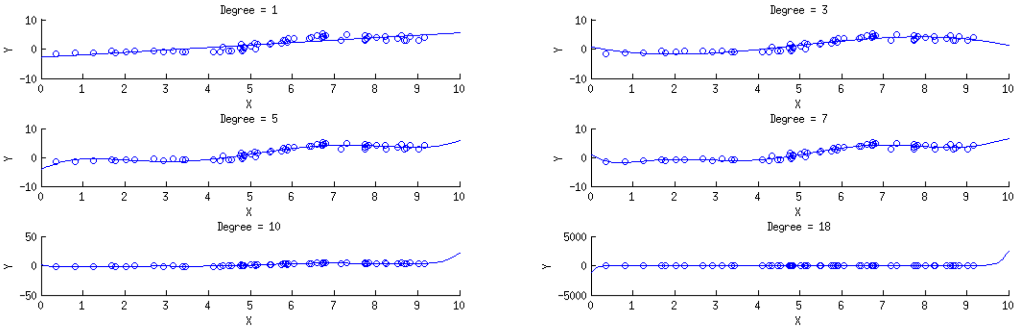
```
%% 1(b)

 lr = linearRegress( Xtr, Ytr );
 lr1 = linearRegress( Xte, Yte );
 xs = [0:.05:10]';
 ys = predict( lr, xs );
 figure,
 plot(xs,ys);
 hold on;
 scatter(Xtr,Ytr);

 ytr_Hat=predict(lr,Xtr);
 yte_Hat=predict(lr1,Xte);
Mse=[(transpose(yte_Hat-Yte)*(yte_Hat-Yte))/(0.25*length(x))];
 Mse_train=[(transpose(ytr_Hat-Ytr)*(ytr_Hat-Ytr))/(0.75*length(x))];
```



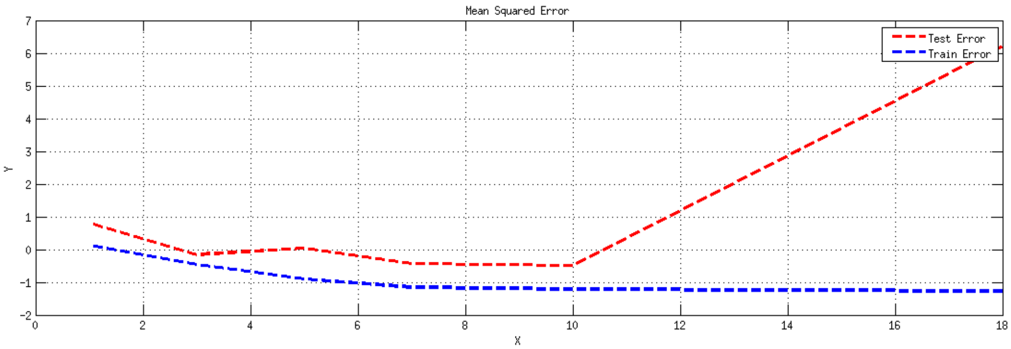
Mean Sq Error (train)=1.127711955609391;
Mean Sq Error (test)=1.768983103736757;

1-c

```
%% 1(c)
j=0;
for d=[1, 3, 5, 7, 10, 18]
XtrP = fpoly(Xtr,d, false);
% create polynomial features up to given degree,

[XtrP, M,S] = rescale(XtrP);
% often a good idea to scale the features
lr = linearRegress( XtrP, Ytr );
% create and train model
xsP= fpoly(xs,d,false);
[xsP] = rescale(xsP,M,S);
ysP=predict(lr,xsP);
ytrP=predict(lr,XtrP);
XteP = fpoly(Xte,d, false);
% create polynomial features up to given degree,

[XteP] = rescale(XteP,M,S);
% often a good idea to scale the features
yteP=predict(lr,XteP);
figure, scatter(Xtr,Ytr);
hold on;
plot(xs,ysP);
j=j+1;
M_se(j)=[(transpose(yteP-Yte)*(yteP-Yte))/(0.25*length(x))];
M_se1(j)=[(transpose(ytrP-Ytr)*(ytrP-Ytr))/(0.75*length(x))];
end
figure; plot([1 3 5 7 10 18],log(M_se),'r-')
hold on;
plot([1 3 5 7 10 18],log(M_se1),'b-');
```

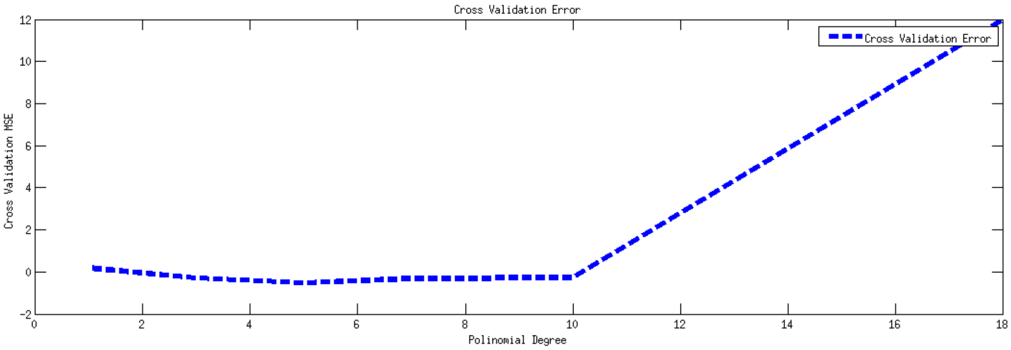


Problem II



```
clear all;

%% 2
DATA=load('data/curve80.txt');                           % read data from text
y=DATA(:,end);
x=DATA(:,1);
[Xtr,Xte ,Ytr, Yte] = splitData(x,y, .75);

% lr = linearRegress( Xtr, Ytr );
% lr1 = linearRegress( Xte, Yte );
% xs = [0:.05:10]';
% ys = predict( lr, xs );
% figure,
% plot(xs,ys);
% hold on;
% scatter(Xtr,Ytr);
%
%
% ytr_Hat=predict(lr,Xtr);
% yte_Hat=predict(lr1,Xte);
% Mse=[(transpose(yte_Hat-Yte)*(yte_Hat-Yte))/(0.25*length(x))];
% Mse_train=[(transpose(ytr_Hat-Ytr)*(ytr_Hat-Ytr))/(0.75*length(x))];

j=1;
for d=[1, 3, 5, 7, 10, 18]

    nFolds = 5;
    for iFold = 1:nFolds,

        [Xti,Xvi,Yti,Yvi] = crossValidate(Xtr,Ytr,nFolds,iFold); % take
        XtrP = fpoly(Xti,d, false);
        [XtrP, M,S] = rescale(XtrP);
        learner = linearRegress(XtrP,Yti);
```

```
            xsP= fpoly(Xvi,d,false);
            [xsP] = rescale(xsP,M,S);
            ysP=predict(learner,xsP);
        % TODO: train on Xti, Yti , the data for this fold
        J(iFold) = [(transpose(ysP-Yvi)*(ysP-Yvi))/(length(Xvi))];
        % TODO: now compute the MSE on Xvi, Yvi and save it
        end;
        MSE(j)=mean(J);
        j=j+1;
end
figure; plot([1 3 5 7 10 18],log(MSE),'r-')
% hold on;
% plot([1 3 5 7 10 18],log(M_se1),'b-');
```



The minimum Cross Validation MSE occurs at Degree= 5.
So, we should choose Degree= 5 for the regression model.
The MSE predicted from Cross Validation is 0.59.
For Degree=5 the test error was 1.034419020563386