CSCC11

Assignment 1

Huan Ma

1000955120

mahuan2

1. Printed version of your Matlab scripts and function files

A)polynomialRegression.m

```
function w = polynomialRegression(K,x,y)
%initialize x matrix first
xMatrix = zeros(length(x), K+1);

% assign values to xMatrix
for i = 1:length(x)
   for j = 0:K
    xMatrix(i, j+1) = x(i)^j;
   end
end

w = xMatrix\y;
end
```

B)evalPolynomial.m

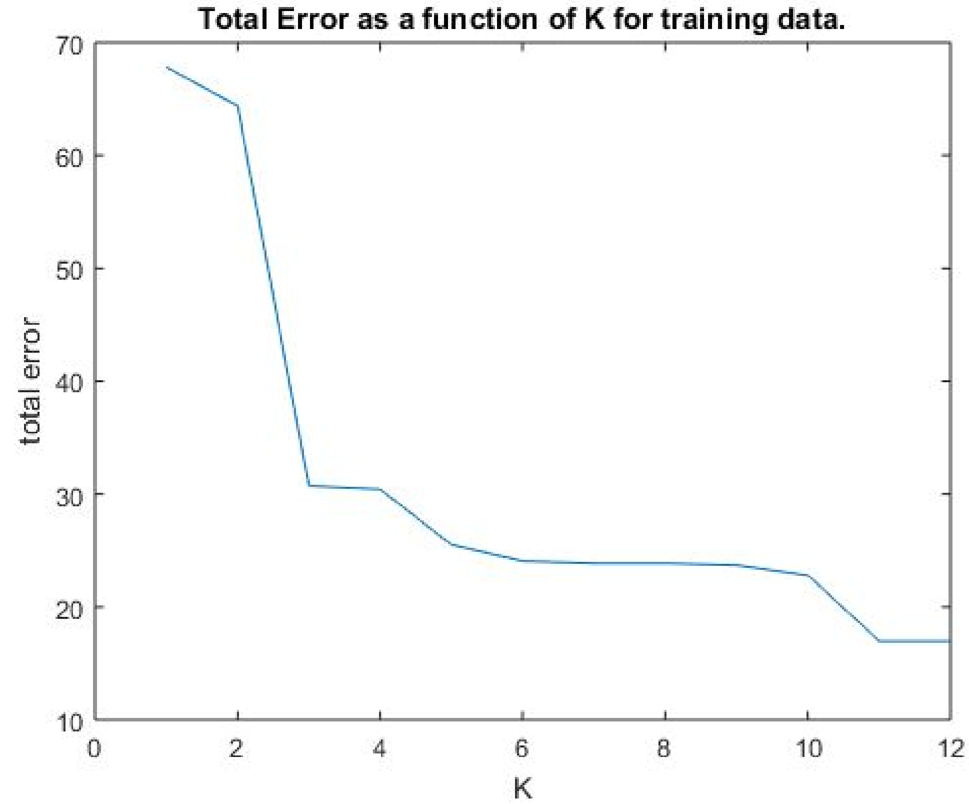
```
function y = evalPolynomial(x,w)
y = zeros(length(x), 1);
for i = 1:length(x)
   for j = 1:length(w)
    y(i, 1) = y(i, 1) + w(j)*x(i)^(j-1);
   end
end
end
```

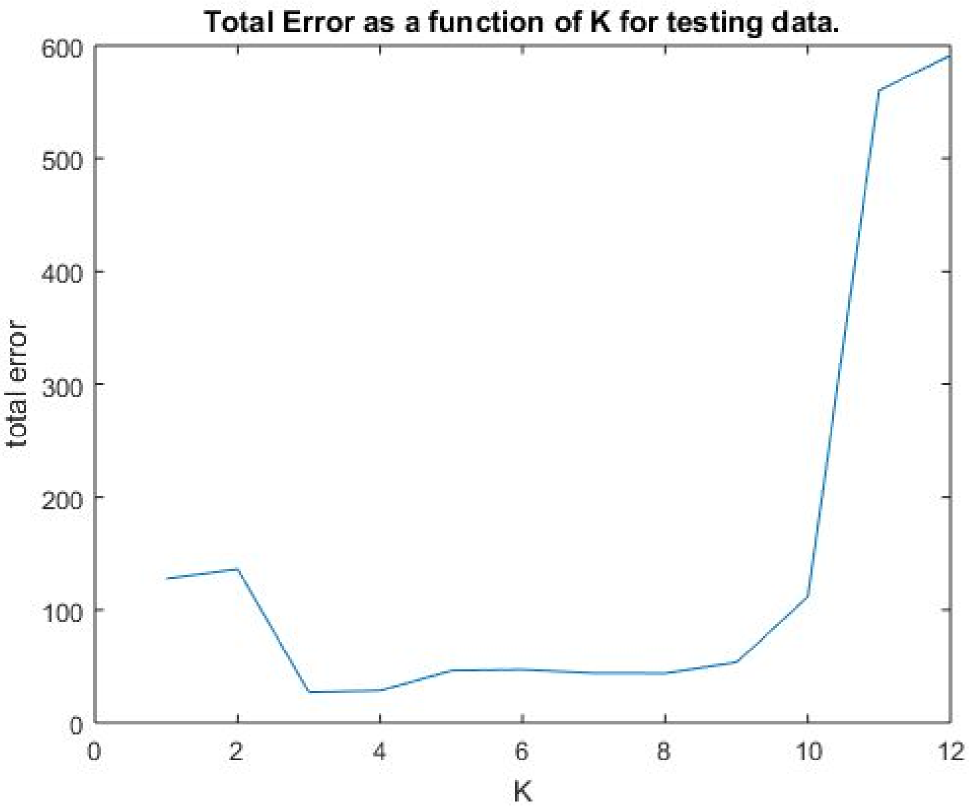
C)a1Script.m

```
%%%%%%%%%%%%%%%%%% FIt
load('a1TrainingData.mat');
%plot(x, y, '*b');

lossTotal = zeros(12, 1);
xPrime = [-2.2:0.1:2.2];
wMatrix = zeros(13, 12);

for K = 1:12
   % find the LS estimate for the polynomial coefficients
   w = polynomialRegression(K,x,y);
   wMatrix( 1: K+1, K) = w;

   % compute the total amount of residual error
   predictedY = evalPolynomial(x,w);
   squaredDifference = (y-predictedY).^2;
   loss = sum(squaredDifference);
   lossTotal(K, 1) = loss;

   % plot the fitted model for a new x
   yPrime = evalPolynomial(xPrime,w);
   figure();
   plot(xPrime, yPrime);
   hold on
end

% plot the total error as a function of K
K=[1:1:12];
```

```
figure();
plot(K, lossTotal);
title('Total Error as a function of K for training data.)'
xlabel('K')
ylabel('total error')
hold on


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Test Models
load('a1TestData.mat');

testLossTotal = zeros(12, 1);

for K = 1:12
  predictedYTest =  evalPolynomial(xTest,wMatrix(:, K));
  % compute the total amount of residual error
  squaredDifference = (yTest-predictedYTest).^2;
  loss = sum(squaredDifference);
  testLossTotal(K, 1) = loss;

end

% plot the total error as a function of K
K=[1:1:12];
figure();
plot(K, testLossTotal);
title('Total Error as a function of K for testing data.)'
xlabel('K')
ylabel('total error')
hold on
```

2. printouts of the two error plots





3. printout of the fit models for each value of K

For each K, the number of Wi is(from W0 to Wk):

K=1: -9.1349   3.7733
2: -9.7244 3.7733 0.3454

3: -9.7244 1.3313 0.3454 0.9255

4: -9.9467 1.3313 0.7259 0.9255 -0.0948

5: -9.9467 2.8549 0.7259 -0.7002 -0.0948 0.3355

6: -9.2930 2.8549 -1.3411 -0.7002 1.1957 0.3355 -0.2088

7: -9.2930 3.2923 -1.3411 -1.6033 1.1957 0.7973 -0.2088 -0.0665

8: -9.2775 3.2923 -1.4156 -1.6033 1.2787 0.7973 -0.2406 -0.0665 0.0039

9: -9.2775 3.8730 -1.4156 -3.5574 1.2787 2.5967 -0.2406 -0.6778 0.0039 0.0687

10: -8.2365 3.8730 -8.2034 -3.5574 12.8890 2.5967 -7.9348 -0.6778 2.1538 0.0687   -0.2126

11: -8.2365 8.8304 -8.2034 -28.0935 12.8890 38.0069 -7.9348 -21.6743 2.1538   5.5018   -0.2126   -0.5083

12: -8.3888 8.8304 -6.9402 -28.0935 9.8536 38.0069 -4.8800 -21.6743 0.6955   5.5018 0.1145   -0.5083   -0.0277

4. written answers to the questions described above

First, consider the plot of the error on the training data as a function of degree K. Explain which models seem to be “good” models and which ones seem to be bad. If you had to pick a value of K just from this plot, what would it be? Explain your reasoning in one or two sentences.

From the total error for taining data plot, it seems like K= 12 is a good model and K=1 is a bad model because K=12 has a lowest total error but K=1 has the largest total error. If I had to choose a value of K just based on this plot, I will choose K=12 because it has the lowest total error, but normally it means overfitting too.

Now, consider the plot of error on the testing data as a function of model degree K. What are some qualitative differences between the behaviour you see in the performance of the models on the training data and on the test data? Can you say which models (i.e., which values of K) appear to overfit the training data? Can you guess the degree of the polynomial that was used to generate the training and test data? Explain the basis for your opinion.

The total error on the training data is decreasing as value of K is increasing while the total error on the testing data is increasing from value 1 to 2 and then decreasing from 2 to 4 and then slowly increasing from 4 to 10, rapidly increasing from 10 to 12. It seems like at K=11 and K=12, the model is overfitting the training data. When K=6, 7 or 8, the total error for training and testing data are both relatively small which gives us confidence to believe that models with these values for K will provide good prediction in the future too. Based on Occam'razor therom, I would like to choose K=6.