ENGN2520 Homework 1

Ming Xu (Banner ID: B01532164)

Problem1

(a)$P(apple) = P(apple|red) * P(red) + P(apple|blue) * P(blue) + P(apple|green) * P(green)$

$$= \frac{30}{30+4+3} * 0.1 + \frac{1}{1+1} * 0.3 + \frac{3}{3+3+4} * 0.6$$

$$= 0.411$$

(b)$P(red|orange) = \frac{P(orange|red)*P(red)}{P(orange|red)*P(red)+P(orange|blue)*P(blue)+P(orange|green)*P(green)}$

$$= \frac{\frac{4}{37} * 0.1}{\frac{4}{37} * 0.1 + \frac{1}{2} * 0.3 + \frac{3}{10} * 0.6} = 0.031$$

Problem2

Let Y be the random variable indicating the number that the fair six-sided dice comes up.

According to the problem, $x = p * Y + (1-p) * 6$

The expectation and variance of Y are shown as below:

$$E(Y) = \sum_{y=1}^{6} y * P(Y = y) = (1+2+3+4+5+6)/6 = 3.5$$

$$var(Y) = E(Y^2) - E^2(Y) = \sum_{y=1}^{6} y^2 * P(Y = y) - 3.5^2 = \frac{1+4+9+16+25+36}{6} - 3.5^2 = 2.917$$

Therefore:

(a) $E(x) = p * E(Y) + (1-p) * 6 = 6 - 2.5p$

(b) $var(x) = p^2 * var(Y) = 2.917p^2$

Problem3

(a) Use the training set to estimate polynomials of degree 1 through 10. The RMS error on the training and test sets are shown in Table.1.

Degree	RMS error on training set	RMS error on the testing set
1	2.366834	3.05745
2	0.84513	1.211299
3	0.636243	1.022073
4	0.592057	0.78546
5	0.591813	0.783315
6	0.584947	0.902174
7	0.567769	1.323815
8	0.544793	4.802609

9	0.454195	6.683277
10	0.446183	11.04976

Also, a figure showing the RMS errors on the training and test sets against degree is shown in Fig.1.

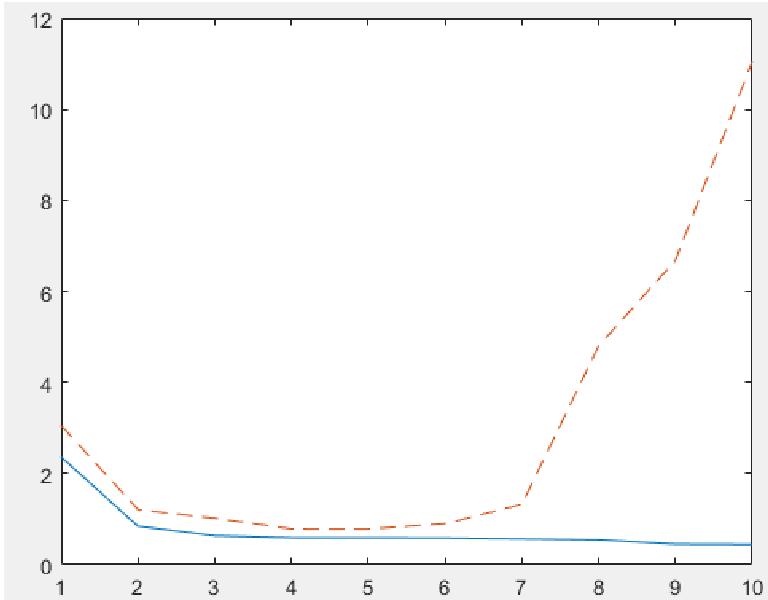

Fig.1. RMS errors on the training and test sets against degree

From the figure, we can easily find the overfitting after degree>7.

(b) The plot showing the training data and the degree 3 polynomial estimated from the data is shown in Fig.2.

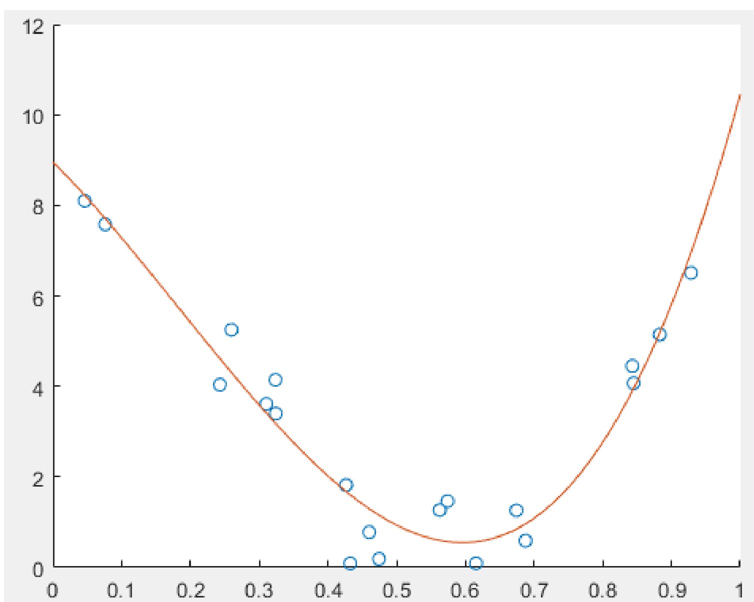

Fig.2. The training data and the degree 3 polynomial estimated from the data

(c) The plot showing the training data and the degree 10 polynomial estimated from the data is shown in Fig.3

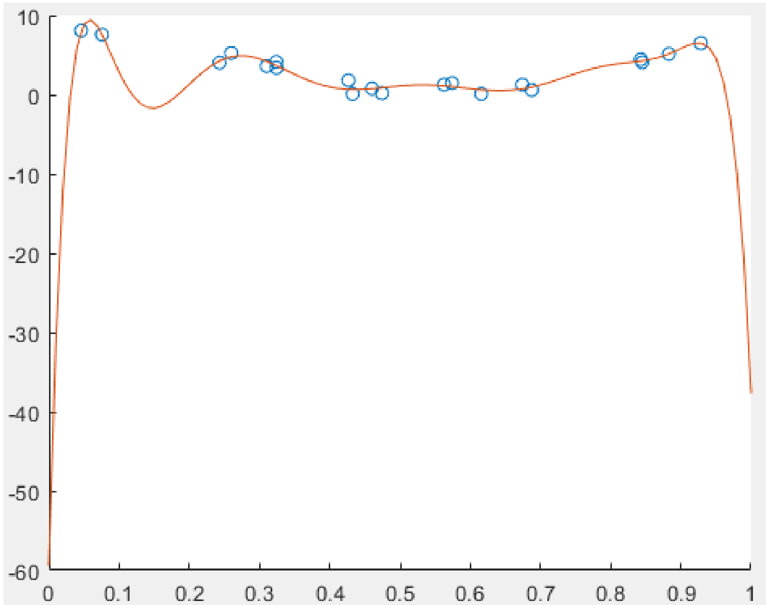

Fig.3. The training data and the degree 10 polynomial estimated from the data

(d) Source code

1. Function to calculate w from training set: “solveW.m”

```
function [w] = solveW(x,y,deg)
    %initialization
    M = zeros(deg+1,deg+1);
    Z = zeros(deg+1,1);
    %contruct matrix M
    for row = 1:deg+1
       for col = 1:deg+1
           M(row,col) = sum((x.^(row-1)).*(x.^(col-1)));
       end
    end
    %construct matrix Z
    for row = 1:deg+1
        Z(row,1) = sum(y.*(x.^(row-1)));
    end
    %calculate w
    w = M\Z;
end
```

2. Function to calculate fx by giving x and w: “calculateFxByGivingW.m”

```
function [fx] = calculateFxByGivingW(x,w)
    %get degree
    [deg, col] = size(w);
```

```
    %get size
    [rowNum, col] = size(x);
    %initialize fx
    fx = zeros(rowNum, col);
    %calculate fx by usuing polynomial function defined by w
    for row = 1:rowNum
        for i = 1:deg
            fx(row,1) = fx(row,1) + w(i)*x(row,1)^(i-1);
        end
    end
end
```

3. Function to calculate RMS error of x and y based on w: “calculateRMS.m”

```
function [rms] = calculateRMS(x,y,w)
    %initialize rms
    rms = 0;
    %get size of input data
    [N,col] = size(x);
    %calcuate fx based on x and w
    fx = calculateFxByGivingW(x,w);
    %calculate rms
    for i = 1:N
        rms = rms+(fx(i)-y(i)).^2;
    end

    rms = sqrt(rms/N);
end
```

4. Main function: “main.m”

(1)For problem 3.1, calculate RMS error on training and testing set using degree through 1 to 10:

```
%loal data
load Xtrain
load Ytrain
load XTest
load Ytest

%loop for 1 to 10 deg to calculate rms for both train data and test
data
degree = 1:10;
degree = degree';
rmsTrain = zeros(10,1);
rmsTest = zeros(10,1);
```

```
for deg = 1:10
    w = solveW(Xtrain,Ytrain,deg);
    rmsTrain(deg) = calculateRMS(Xtrain,Ytrain,w);
    rmsTest(deg) = calculateRMS(Xtest,Ytest,w);
end

plot(degree,rmsTrain,degree,rmsTest,'--')
```

(2) For problem 3.2, plot the figure showing the training data and the degree 3 polynomial estimated from the data:

```
%plot showing the training data and degree 3 polynomial estimated from
the data
w = solveW(Xtrain,Ytrain,3);
scatter(Xtrain,Ytrain); hold on;
x = 0:0.01:1;
x = x';
fx = calculateFxByGivingW(x,w);
plot(x,fx);
```

(2) For problem 3.3, plot the figure showing the training data and the degree 10 polynomial estimated from the data:

```
%plot showing the training data and degree 10 polynomial estimated from
the data
w = solveW(Xtrain,Ytrain,10);
scatter(Xtrain,Ytrain); hold on;
x = 0:0.01:1;
x = x';
fx = calculateFxByGivingW(x,w);
plot(x,fx);
```

Please find the source code on "https://github.com/Xuming8812/ENGN2520"