Khoi Duong

Prof. Yang

CS483

6/29/2022

MIDTERM

1.

Based on observation, we will choose the highest order as 5.

Thus, we have the hypothesis function:

$$h(\theta) = y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$$

Loss function:

$$L = [h(x^{(i)}) - y^{(i)}]^2$$

Cost function:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}[h(x^{(i)}) - y^{(i)}]^2 = \frac{1}{n}\sum_{i=1}^{n}[(\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}) - y^{(i)}]^2$$

We have the partial derivative function for each coefficient as below:

Partial derivative with respect to θ_0:

$$\frac{dJ}{d\theta_0} = \frac{-2}{n}\sum_{i=1}^{n}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

Partial derivative with respect to θ_1:

$$\frac{dJ}{d\theta_1} = \frac{-2}{n}\sum_{i=1}^{n}x^{(i)}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

Partial derivative with respect to θ_2:

$$\frac{dJ}{d\theta_2} = \frac{-2}{n}\sum_{i=1}^{n} x^{(i)^2}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

Partial derivative with respect to θ_3*:*

$$\frac{dJ}{d\theta_3} = \frac{-2}{n}\sum_{i=1}^{n} x^{(i)^3}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

Partial derivative with respect to θ_4*:*

$$\frac{dJ}{d\theta_4} = \frac{-2}{n}\sum_{i=1}^{n} x^{(i)^4}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

Partial derivative with respect to θ_5*:*

$$\frac{dJ}{d\theta_5} = \frac{-2}{n}\sum_{i=1}^{n} x^{(i)^5}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)} + \theta_2 x^{(i)2} + \theta_3 x^{(i)3} + \theta_4 x^{(i)4} + \theta_5 x^{(i)5}))$$

If the hypothesis function generates a high error for the testset as follows after modeling, it means that the model is overfit. There are a few ways to prevent overfitting:

- Cross-validation
- Early stopping before it becomes overfit the training data
- Train with more data
- Remove hidden features in some built-in algorithms

2.

The dataset presents binary classification with two features

Supposed that x_1 is alcohol, x_2 is malic acid feature.

Thus, we have the hypothesis function as below

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ where $g(z) = \frac{e^z}{1 + e^z}$

We have the cost function as below:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} * \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) * \log(h_\theta(x^{(i)}))]$$

And we also have the loss function:

$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) * x_j^{(i)}$$

Let $\theta_0 = \theta_1 = \theta_2 = 0$, $m = 5$, and learning rate $\alpha = 0.0001$

We have the source code below:

```
import numpy as np
import matplotlib.pyplot as plt
import math

alcohol = [14.23, 13.2, 13.16, 14.37, 13.24]
malic_acid = [1.71, 1.78, 2.36, 1.95, 2.59]
y = [0,1,1,0,0]

theta_0 = theta_1 = theta_2 = 0
alpha = 0.0001
i = 0

while i <= 500000:
  diff_theta_0 = diff_theta_1 = diff_theta_2 = 0
  for m in range (5):
    diff_theta_0  +=  (1/(1  +  math.exp(-(theta_0  +  theta_1*alcohol[m]  + theta_2*malic_acid[m])))) - y[m]
    diff_theta_1  +=  ((1/(1  + math.exp(-(theta_0  +  theta_1*alcohol[m]  + theta_2*malic_acid[m])))) - y[m]) * alcohol[m]
    diff_theta_2  +=  ((1/(1  + math.exp(-(theta_0  +  theta_1*alcohol[m]  + theta_2*malic_acid[m])))) - y[m]) * malic_acid[m]
  diff_theta_0 = diff_theta_0 * (1/5)
  diff_theta_1 = diff_theta_1 * (1/5)
  diff_theta_2 = diff_theta_2 * (1/5)
```

```
    theta_0 = theta_0 - alpha * diff_theta_0
    theta_1 = theta_1 - alpha * diff_theta_1
    theta_2 = theta_2 - alpha * diff_theta_2
    i += 1

print("diff_theta_0 = " + str(diff_theta_0) + ", " + "Theta 0 = ",
str(theta_0))
print("diff_theta_1 = " + str(diff_theta_1) + ", " + "Theta 1 = ",
str(theta_1))
print("diff_theta_2 = " + str(diff_theta_2) + ", " + "Theta 2 = ",
str(theta_2))
```

Run program & result:

```
diff_theta_0 = -0.011688662162064524, Theta 0 =  0.6061870850042725

diff_theta_1 = 0.0013360767057946533, Theta 1 =  -0.15566558771634545

diff_theta_2 = -0.0031292279040260776, Theta 2 =  0.508615476555844
```

Thus, the hypothesis function is:

$h_\theta(x) = g(0.60619 - 0.15567x_1 + 0.50862x_2)$ where $g(z) = \dfrac{e^z}{1 + e^z}$ and x_1 is alcohol feature, and x_2 is malic acid feature.

3.

Hypothesis function: $h(\theta) = \theta_0 + \theta_1 x$

Loss function: $L = [h(x^{(i)}) - y^{(i)}]^2$

Cost function: $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$

Gradient decent algorithm:

$$\theta_0 = \theta_0 - \alpha\frac{\partial J(\theta)}{\partial \theta_0}$$

$$\theta_1 = \theta_1 - \alpha\frac{\partial J(\theta)}{\partial \theta_1}$$

We have:

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}[(\theta_0 + \theta_1 x_1^{(i)}) - y^{(i)}]$$

$$\frac{\partial J(\theta)}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}[(\theta_0 + \theta_1 x_1^{(i)}) - y^{(i)}] * x_1^{(i)}$$

Let $\theta_0 = \theta_1 = 0$, m = 5, and learning rate $\alpha = 0.0001$

We have the source code below:

```
import numpy as np
import matplotlib.pyplot as plt
import math

x = [1,2,3,4,5]
y = [7,9,12,15,16]

theta_0 = theta_1 = 0
alpha = 0.0001
i = 0

while i <= 500000:
  diff_theta_0 = diff_theta_1 = 0
  for m in range (5):
    diff_theta_0 += (theta_0 + theta_1*x[m] - y[m])
    diff_theta_1 += (theta_0 + theta_1*x[m] - y[m])*x[m]

  diff_theta_0 = diff_theta_0 * (1/5)
  diff_theta_1 = diff_theta_1 * (1/5)

  theta_0 = theta_0 - alpha * diff_theta_0
  theta_1 = theta_1 - alpha * diff_theta_1
  i += 1
```

```
print("diff_theta_0 = " + str(diff_theta_0) + ", " + "Theta 0 = ", str(theta_0))
print("diff_theta_1 = " + str(diff_theta_1) + ", " + "Theta 1 = ", str(theta_1))
```

Run program & result:

```
diff_theta_0 = -0.00013183008940238495, Theta 0 =  4.599220175459972

diff_theta_1 = 3.6514820883226665e-05, Theta 1 =  2.4002159988930627
```

Thus, the linear regression is $h(\theta) = y = 2.4 + 4.6x$

4.

In the process of applying gradient descent algorithm to find max value for each coefficient in hypothesis function, appropriate learning rate α is very important because it can impact the training result and the regression function. For example, a large learning rate will decrease the accuracy of the regression function, since the "diff_theta" will be very large as it misses the minimum or maximum point. On the other hand, a very small learning rate will make the process of regression become very slow, thus it impacts on the running time while the accuracy does not increase proportionally. A balance learning rate will balance between the accuracy and the running time of the regression process.

5.

Hypothesis function: $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$

The model predicts y = 1 if

$-1 + x_1^2 + x_2^2 \geq 0 \Leftrightarrow x_1^2 + x_2^2 \geq 1$

Cost function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_0(x^{(i)}))$$

Loss function:

$$L = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

Partial derivative function for gradient descent is the same form of terms with the one used for linear regression. Thus, we have:

$$\begin{bmatrix} \frac{\partial J(\theta)}{\partial \theta_0} \\ \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{bmatrix} = \frac{1}{m}x^T(h(x) - y)$$

Thus, the partial derivative is:

$$\frac{\partial(J(\theta))}{\partial(\theta)} = \frac{1}{m}X^T[h_\theta(x) - y]$$

6.

7.

If k = 2, we randomly choose 2 points A2 and A4.

	Pnts(x)	Pnts(y)	Cluster 1 Dist to A2(8,4)	Cluster 2 Dist to A4(6,4)	Cluster
A1	2	10	12	10	Cluster 2
A2	8	4	0	2	Cluster 1
A3	5	8	7	5	Cluster 2
A4	6	4	2	0	Cluster 2
A5	1	2	9	7	Cluster 2

Center of cluster 1

	Pnts(x)	Pnts(y)
A2	8	4
Mean	8	4

Center of cluster 2

	Pnts(x)	Pnts(y)
A1	2	10
A3	5	8
A4	6	4
A5	1	2
Mean	3.5	6

			Cluster 1	Cluster 2	
	Pnts(x)	Pnts(y)	Dist to (8,4)	Dist to (3.5,6)	Cluster
A1	2	10	12	5.5	Cluster 2
A2	8	4	0	6.5	Cluster 1
A3	5	8	7	3.5	Cluster 2
A4	6	4	2	4.5	Cluster 1
A5	1	2	9	6.5	Cluster 2

Center of cluster 1

	Pnts(x)	Pnts(y)
A2	8	4
A4	6	4
Mean	7	4

Center of cluster 2

	Pnts(x)	Pnts(y)
A1	2	10
A3	5	8
A5	1	2
Mean	2.67	6.67

			Cluster 1	Cluster 2	
	Pnts(x)	Pnts(y)	Dist to (7,4)	Dist to (2.67,6.67)	Cluster
A1	2	10	11	4	Cluster 2
A2	8	4	1	8	Cluster 1
A3	5	8	6	3.66	Cluster 2
A4	6	4	1	6	Cluster 1
A5	1	2	8	6.34	Cluster 2

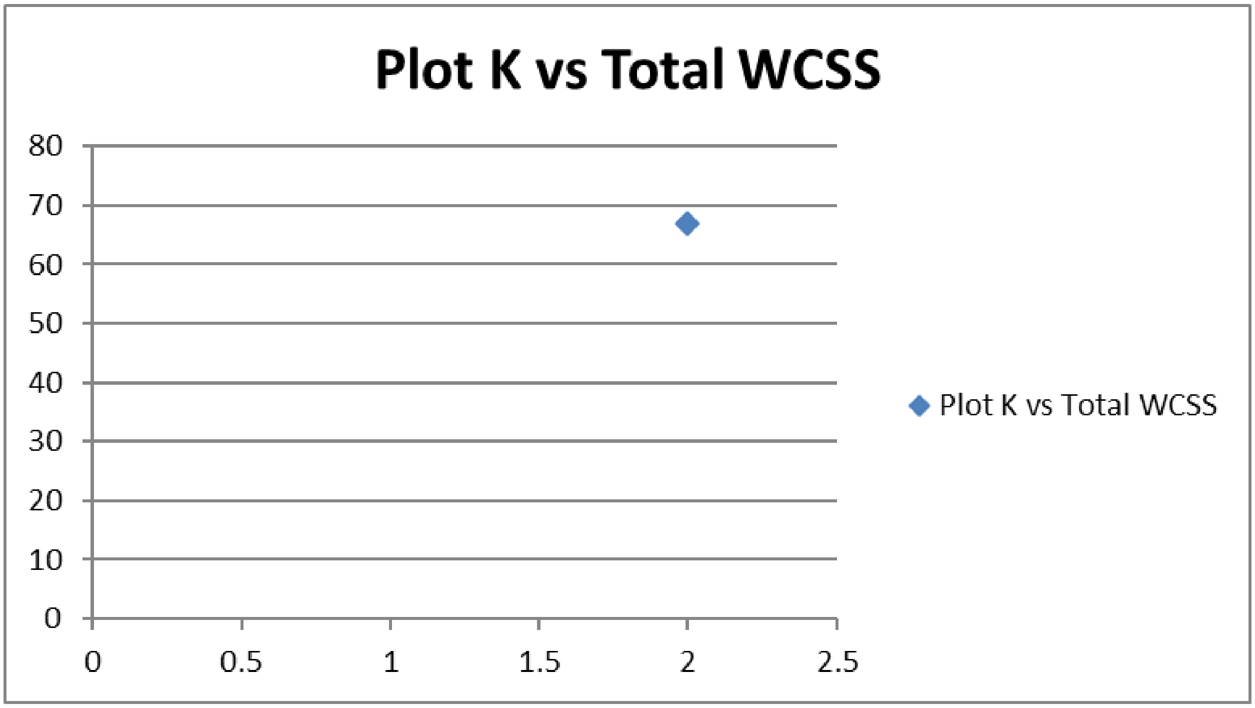
cluster 1

	Pnts(x)	Pnts(y)	Distance A1	A4	
A2	8	4	0		
A4	6	4	4	0	Tot Sum
		Col Sum	4	0	4
	WCSS	**2**			

cluster 2

	Pnts(x)	Pnts(y)	A1	A3	A5	
A1	2	10	0			
A3	5	8	13	0		
A5	1	2	65	52	0	**Tot Sum**
		Col Sum	78	52	0	130
	WCSS	**65**				

K	Total WCSS
2	67


Plot K vs Total WCSS
80
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
2.5
Plot K vs Total WCSS