CS-596 Machine Learning Homework Assignment 2

Hsuan Yu, Liu Red ID:823327369

**Algorithm and Decision,**

Normalization:

I tried two ways to normalize the data.

1. Scale values to be 0 to 1: In NumPy, I am able to get the max value and the min value by list built-in functions. After that, writing an equation likes it’s numeric. The NumPy will calculate all elements by the same equation.

Code:

value\_max, value\_min = sat.max(), sat.min()

sat = (sat - value\_min)/(value\_max - value\_min)

2. Taking mean off: At first, using numpy built-in function mean(), we can get mean values in data. Changed the sat – value\_min to sat – mean. Then, the result will be what I need.

gradientDescent: Three parts

1. residualError: We know that residual is , so using np.dot can get the result.

Code: residualError = np.dot(X,theta) - y

2. gradient: We know that gradient is . Using transpose X to time residual error can get the sum for all thetas.

Code: gradient = (1/m)\*(transposedX.dot(residualError))

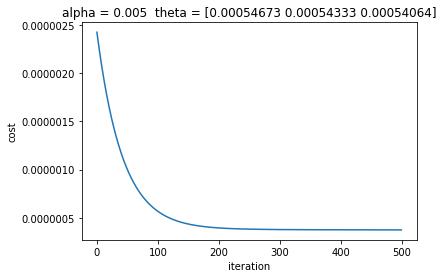
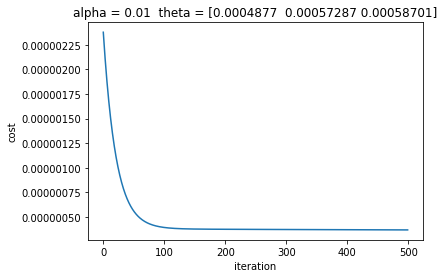
3. According to the equation of cost function , we found is similar to residualError. Also, np.sum() and “\*\*” can help us get the result.

Code: (1 / m) \* np.sum((np.dot(X,theta) - y) \*\* 2)

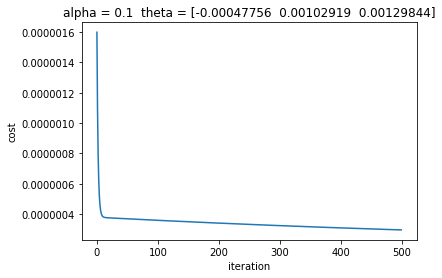
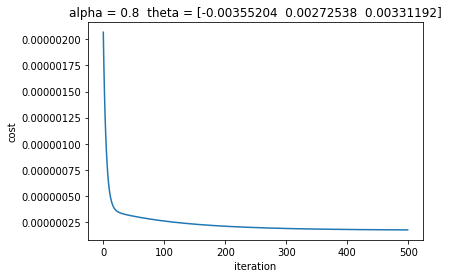
**Convergence curve with different ALPHA and MAX\_ITER,**

Using different ALPHA can control the reducing rate of cost function. Control the MAX\_ITER for observing the reducing rate.

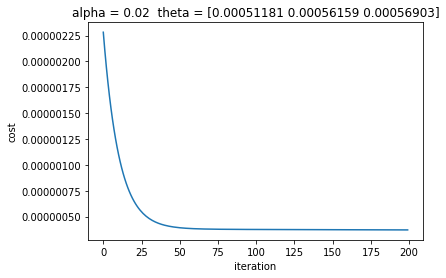
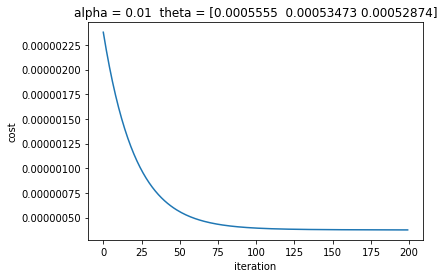
ALPHA = 0.005 ALPHA = 0.01

ALPHA = 0.1 ALPHA = 0.8

Double the ALPHA uses half times to approach to the limit value. Below images show that alpha=0.01 used around 100 times to approach the bottom, and alpha= 0.02 only used around 50 times.



However, ALPHA is not the larger, the better. If ALPHA is too large, the cost will start to increase. The below images show that when the alpha= 0.85, the cost will be higher and higher.

