A Linear Regression to Predict GPA

Yuanjian Zhou
ECON 433

Februay, 7, 2020

1 Introduction

In this report, I will learn a linear regression algorithm to predict a student's university GPA. In the linear regression model, SAT Maths score and Verb score are chosen as features. Following instruction, I use 60 of 105 observations as training set, and others as testing set. As instruction, I will try different values of parameters and compare the model's performance using test error.

2 Algorithm

2.1 Normalization

Since features might in different scales, before learning the model we need to do the normalization. In this report, I will choose three types of normalization(before the section of testing of normalization, I will use 0-1 normalization as default):

2.1.1 0-1 Normalization

$$X = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

2.1.2 Demean Normalization

$$X = \frac{X - \bar{X}}{X_{max} - X_{min}} \tag{2}$$

2.1.3 Z Normalization

$$X = \frac{X - \bar{X}}{\sigma_X} \tag{3}$$

2.2 Gradient Descent

In this report, I will use Gradient Descent to learn the parameters of linear regression:

$$\theta_j := \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j} \tag{4}$$

We will use this algorithm to update θ until they converge. In the following section, I will choose different numbers of max iteration and learning rate α to show how the cost of the linear regression model changes with these two parameters.

3 Results

3.1 Testing of Max Iteration

I denote max iteration as N, and for each case, I will report the convergence curve In this section, I will fix α as 0.1.

- Case 1 N=1000

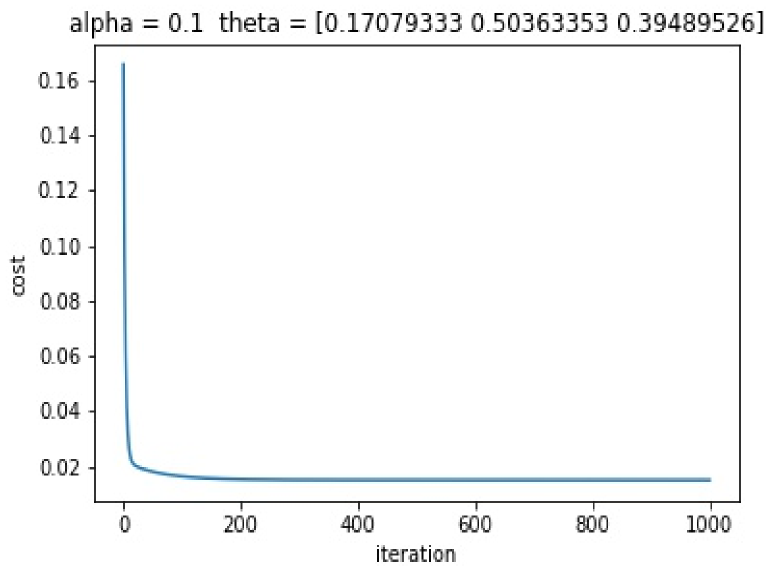

Figure 1: N = 1000

- Case 2 N=500

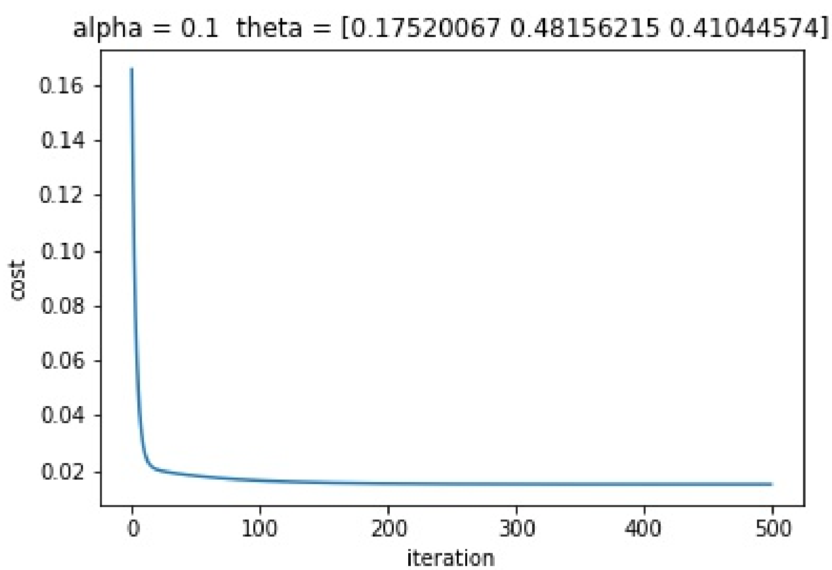

Figure 2: N = 500

- Case 3 N=100

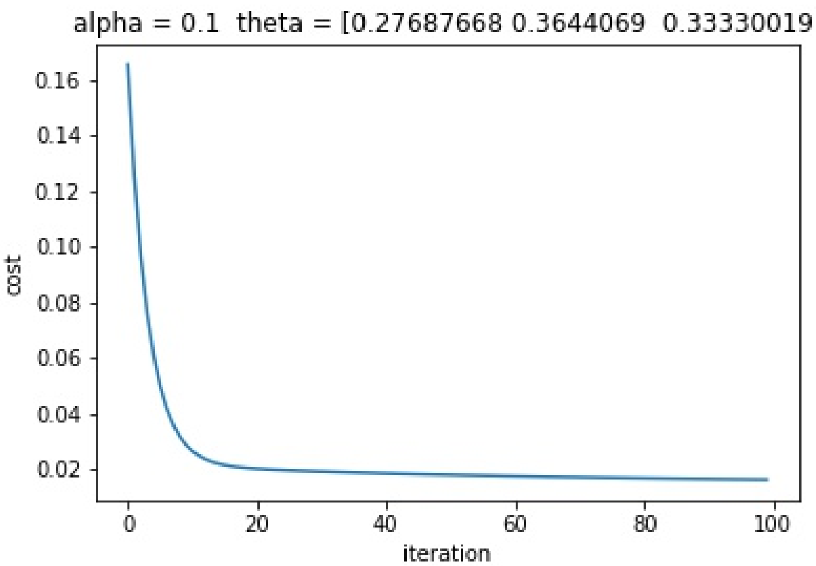

Figure 3: N = 100

- Case 4 N=50

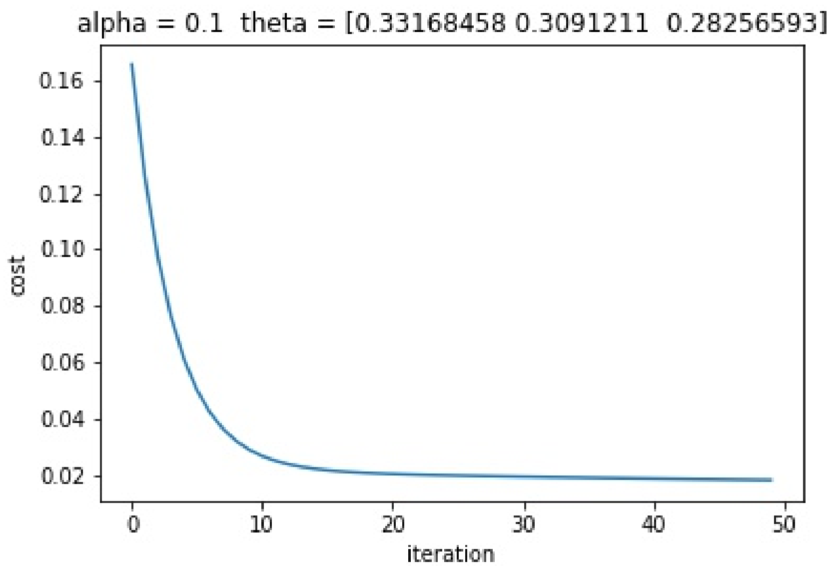

Figure 4: N = 50

From these four cases, we find that the parameter converges after about 200 iterations, since the cost function does not drop down a lot. This observation means that when learning rate is 0.1, adding iterations to over 200 will not do much help. However, in practice, when you have limited features, adding iterations will not cost much more time, so I think we should use a large number such as 5000 to guarantee the convergence.

3.2 Testing of Learning Rate α

In this section, I will try learning rate with 0.01, 0.1 and 1 and fix N as 1000.

- Case 1 α=0.01

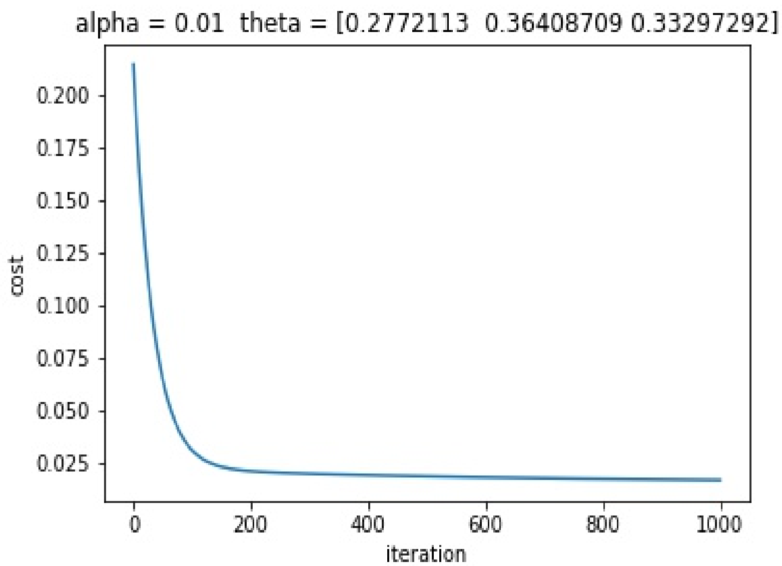

Figure 5: α=0.01

- Case 2 α=1

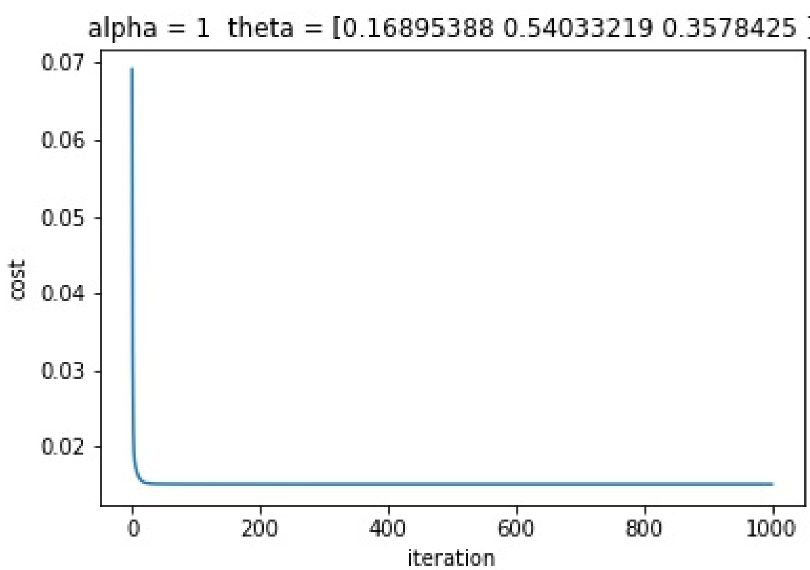

Figure 6: α=1

From these cases, we find that there is a positive correlation between the learning rate and the iteration needed for convergence.

3.3 Evaluation for trials

In this section, I will give a detailed discussion by reporting the test error of the 6 cases above.

Table 1: Statistics of Test Error

	mean	std
N=50,α=0.1	0.1470	0.1172
N=100,α=0.1	0.1541	0.1190
N=500,α=0.1	0.1728	0.1269
N=1000,α=0.1	0.1744	0.1272
N=1000,α=0.01	0.1540	0.1190
N=1000,α=1	0.1748	0.1274

From Table 1, we find that holding α constant, the average test error increases with iteration. I think this is because the cost will decrease a small number after each iteration, and this behavior will make our model fit the training set well while might be bad for the testing set. Besides, when holding N constant, test error increases with α, this might have little explanatory power because the parameter might not converge at N = 1000.

3.4 Testing of Normalization

According to the results in the above table, we will fix N = 1000 and α = 0.1 now. In this section, we will report test error using raw data, data using z-normalization, and data using demean normalization. From the table, we find that performing normalization to features will largely increase the performance of prediction, and for this data set, it seems that 0-1 normalization and demean are better.

Table 2: Statistics of Test Error

	mean	std
Raw	$overflow$	$overflow$
Z	0.6794	0.4946
01	0.1744	0.1272
Demean	0.1739	0.1273

4 conclusion

In this report, I learned a linear regression to predict a student's university GPA. My main work is to try different parameters and normalization method, researching how the convergence curve and test error will change.
I just list a few possible cases in this report, so I can not make conclusion about which one is the best since there are too many combinations of parameters, i.e max iteration and learning rate. What I can conclude is just that normalization will make the model perform better and a smaller learning rate needs more iterations to come to convergence. I think maybe I could try Normal Equation next time, because it could give me a numeric answer about which thetas will minimize the cost function.