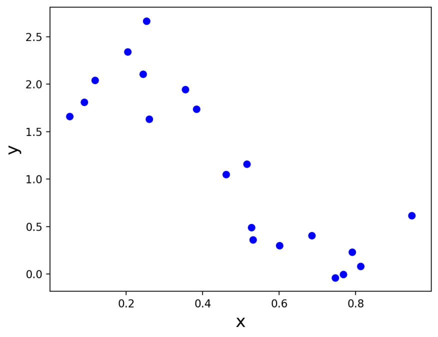
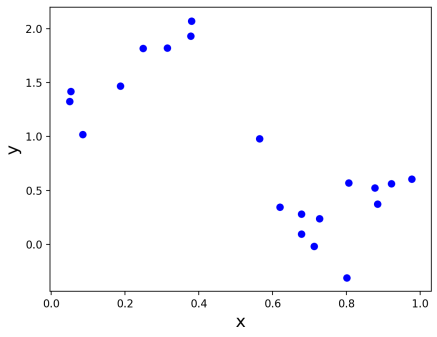
CM146, Fall 2018
Problem Set 2: Xiwei Ma, 704755732

1 Problem 4

1.1 Visualization

(a)





. Figure 1: Training Data Figure 2: Test Data

I observe that there seem to be a negative linear correlation in figure 1, the training data; but it is hard to see the test data negative related. So if training model use linear regression, it may not perform well on test.

1.2 Linear Regression

(b) In regression.py.
(c) In regression.py.
(d)Steps completed in regression.py.

learning rate	coefficient	# of iterations	final value	time
0.0001	[2.27044798 -2.46064834]	10000	4.0864	1.06
0.001	[2.4464068 -2.816353]	7020	3.9126	0.728
0.01	[2.44640703 -2.81635346]	764	3.9126	0.082
0.0407	[-9.40470931e+18 -4.65229095e+18]	10000	2.7109	1.041

For learning rate 10^{-4},10^{-3},10^{-2}, when the rate increases, the number of iterations to converge, time, and error decrease while the coefficients are similar. In this range, the greater the rate, the faster it converges and there would be less error. However, when the rate increases to 0.0407, the step is too big for it to converge in 10000 steps. Thus, the coefficient does not make sense.

(e) The closed-form solution is implemented in regression.py.
The result is:

coefficient	final value	time
[2.44640709 -2.81635359]	3.9126	0.0084

For closed form, the resulting coefficient and final value are similar to the approximations made by gradient descent with appropriate learning rate. The time it takes is similar to the time of gradient descent using the best learning rate, and much faster than the average time running gradient descent.

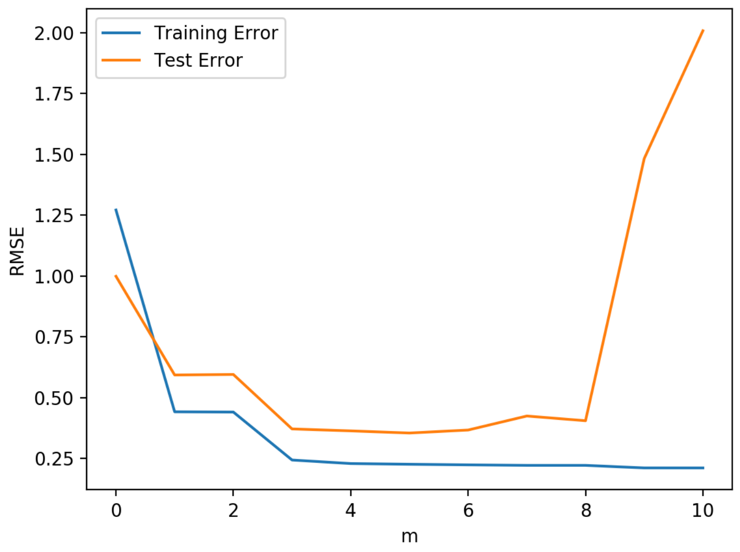
(f)The result is:

coefficient	# of iterations	final value	time
[2.44640676 -2.81635292]	1356	3.9126	0.00019

The coefficient and final value are similar to the results using closed form, gradient descent of 0.001 and 0.01. The number of iteration it takes is in between that of 0.001 and 0.01. However, the time it takes is the shortest.

(g) In regression.py.

(h) Since the result is divided by n in RMSE, the size of data set would no longer affect the evaluations of error of different models that much. Thus, RMSE is more preferable than calculating $J(w)$ since it can represent performances of models better.
(i)



From the graph we can see that when degree m = 5, it fits the data best. This is when both training error and test error being relatively low. When degree m is lower than 3, the training error is apparently high, as well as the test error, which indicates underfitting. And when degree m is higher than 8, the training keeps low(blue line) while the test error(orange line) is high. This indicates the model is overfitting.