Assignment 2: Polynomial Regression

Logan Keeler
Department of Statistics
University of Nebraska, Lincoln
logan.keeler@huskers.unl.edu

Jigyasa Chauhan
School of Computing
University of Nebraska, Lincoln
jchauhan2@huskers.unl

Sanjay Alamaru
School of Computing
University of Nebraska, Lincoln
salamuru2@huskers.unl.edu

***Abstract*—We are using the wine quality data set, performing linear regression using gradient descent. We are try to get best hyper parameters to find predictions as required,**

***Index Terms*—polynomial regression, gradient descent, linear regression, mean squared errors**

I. Introduction

1) **Q1:Describe whether or not you used feature scaling and why or why not.**
2) **Q2:Describe whether or not you dropped any feature and why or why not.**
3) **Q3:In the lecture we have studied two types of Linear Regression algorithm: closed-form solution and iterative optimization. Which algorithm is more suitable for the current data set? Justify your answer.**
4) **Q4:Would the batch gradient descent and the stochastic gradient descent algorithm learn similar values for the model weights? Justify your answer. Let's say that you used a large learning rate. Would that make any difference in terms of learning the weights by both algorithms?**

II. Solutions

III. Describe whether or not you used feature scaling and why or why not.

We do scale the features to get a better interpret -able value for intercepts and we get a value for mean as zero. This also helps us get a unit variance. Since the data can be skewed scaling helps us get more data points with a mean close to zero. Gradient Descent also takes a longer time to converge when feature scales vary meaning that it would require more iterations for our weights to converge to the correct values.

IV. Describe whether or not you dropped any feature and why or why not.

For this assignment we do not drop any feature since we do not correlate with any of the features and provide analysis for our target variable i.e. the wine quality in this case. We usually drop features when the time complexity is usually high and we would improve computational complexity. It could have helped us identify the least resistance for a feature to help predicting, but in this case we do not need.

V. In the lecture we have studied two types of Linear Regression algorithm: closed-form solution and iterative optimization. Which algorithm is more suitable for the current data set? Justify your answer.

For this specific problem we use the iterative optimization algorithm. Since the iterative approach helps us generate a sequence of outputs. The outputs helps us understand the series improved approximation values. Therefore, similar approach is used by the gradient descent which takes one input and provide the series of improved converged path. Whereas the closed form solution would only take some definite number of iterations. The gradient descent increments the weights after every epoch whereas in closed form solution this does not happen. The closed form algorithm can also have very high time-complexity for high dimension. For example, one time consuming calculation can be finding the inverse of $X^T X$. We have 11 features. However, as we grow to check our polynomial regression, the number of parameters that you have to check grows. An iterative optimization like Gradient descent is very useful for a smaller data set with a lot of features like our data set. Note also that we are using a convex cost function meaning it has doesn't have any local minima and only has a global minimum meaning that Gradient Descent is guaranteed to reach the minimum with a reasonable learning rate.

VI. Would the batch gradient descent and the stochastic gradient descent algorithm learn similar values for the model weights? Justify your answer. Let's say that you used a large learning rate. Would that make any difference in terms of learning the weights by both algorithms?

The batch gradient descent is faster than the stochastic gradient descent because it only has to manipulate little data. It bounces around more once it gets close to the optimal value. So stochastic gradient descent will always get good parameter values but not optimal. Using a large learning rate might which would make worse values for optimization. We fix this by doing a learning schedule though and gradually decreasing the learning rate as we go through the iterations, so it does not oscillate so much at the end.

Figure 1 describes the Mean, standard deviation and quartiles required for question 6, in the assignment.

VI. .DETERMINETHE BEST MODEL HYPERPARAMETERVALUESFOR THE TRAINING DATA MATRIX WITH POLYNOMIAL DEGREE 3

lambda	learningrate	regularizer	mse
1	0.1	L1	0.43795
1	0.1	L2	0.43845
1	0.01	L1	0.43926
1	0.01	L2	0.43945
1	0.001	L1	0.44597
1	0.001	L2	0.44618
1	0.0001	L1	6.88483
1	0.0001	L2	6.88903
0.1	0.1	L1	0.43843
0.1	0.1	L2	0.43849
0.1	0.01	L1	0.43945
0.1	0.01	L2	0.43948
0.1	0.001	L1	0.44619
0.1	0.001	L2	0.44622
0.1	0.0001	L1	6.88947
0.1	0.0001	L2	6.88989
0.01	0.1	L1	0.43848
0.01	0.1	L2	0.43849
0.01	0.01	L1	0.43948
0.01	0.01	L2	0.43948
0.01	0.001	L1	0.44621
0.01	0.001	L2	0.44622
0.01	0.0001	L1	6.88994
0.01	0.0001	L2	6.88998
0.001	0.1	L1	0.43849
0.001	0.1	L2	0.43849
0.001	0.01	L1	0.43948
0.001	0.01	L2	0.43948
0.001	0.001	L1	0.44622
0.001	0.001	L2	0.44622
0.001	0.0001	L1	6.88998
0.001	0.0001	L2	6.88999
0.0001	0.1	L1	0.43849
0.0001	0.1	L2	0.43849
0.0001	0.01	L1	0.43948
0.0001	0.01	L2	0.43948
0.0001	0.001	L1	0.44622
0.0001	0.001	L2	0.44622
0.0001	0.0001	L1	6.88999
0.0001	0.0001	L2	6.88999

MSE values for cross validation

Fig. 1. Best Hyperparameters

Finding 1: Table VI: Best hyperparameter is lambda 1 with learningrate 0.1 and using L1 regularization

Finding 2: Figure 3: We plotted the best rmse's of each polynomial. We used 0.001 for lambda and learning rate because it had the best hyperparameters for each of the models. As you can see the rmse is best at polynomial degree of 1 with an rmse of 0.66728. When it starts to get higher, the validation rmse starts to go up as the training error stays relatively the same.

Finding 3: Table VII: Best hyperparameter is lambda=1 with learning rate of 0.001

Finding 4: As per question 9, in the assignment, MSE for test data is 0.44331.

VII. EVALUATE YOUR MODELON THE TEST DATAAND REPORT THE MEAN SQUARED ERROR.

lambda	learningrate	regularizer	mse
1	0.1	L1	nan
1	0.1	L2	nan
1	0.01	L1	0.500965
1	0.01	L2	0.522798
1	0.001	L1	0.782414
1	0.001	L2	0.843326
1	0.001	L1	0.782414
1	0.001	L2	0.843326
0.1	0.1	L1	nan
0.1	0.1	L2	nan
0.1	0.01	L1	0.519434
0.1	0.01	L2	0.524919
0.1	0.001	L1	0.840538
0.1	0.001	L2	0.846861
0.1	0.001	L1	0.840538
0.1	0.001	L2	0.846861
0.01	0.1	L1	nan
0.01	0.1	L2	nan
0.01	0.01	L1	0.524548
0.01	0.01	L2	0.525187
0.01	0.001	L1	0.84657
0.01	0.001	L2	0.84723
0.01	0.001	L1	0.84657
0.01	0.001	L2	0.84723
0.001	0.1	L1	nan
0.001	0.1	L2	nan
0.001	0.01	L1	0.525147
0.001	0.01	L2	0.525213
0.001	0.001	L1	0.847188
0.001	0.001	L2	0.84724
0.001	0.001	L1	0.847188
0.001	0.001	L2	0.84724
0.0001	0.1	L1	nan
0.0001	0.1	L2	nan
0.0001	0.01	L1	0.525209
0.0001	0.01	L2	0.525213
0.0001	0.001	L1	0.84724
0.0001	0.001	L2	0.847245
0.0001	0.001	L1	0.84724
0.0001	0.001	L2	0.847245

Fig. 2. Mean, Standard deviation and Quartiles

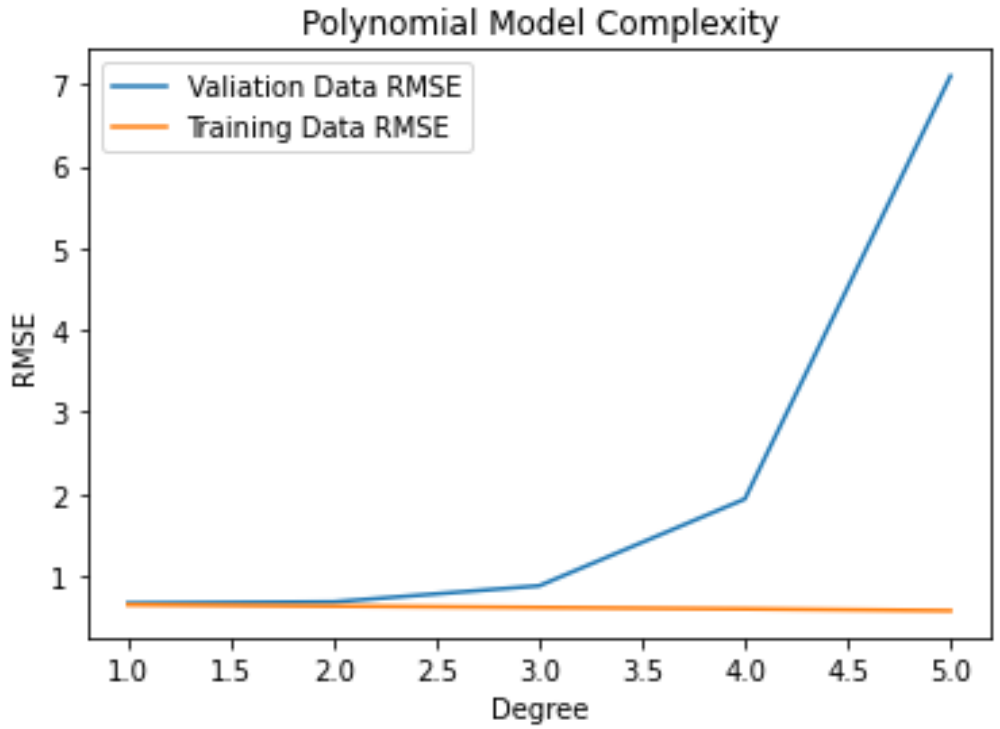

Fig. 3. RMSE Curve

Finding 5: As per question 12, Best hyperparameters are lambda = 1 with learning rate=0.1 with L2 regularization with an MSE of 0.478 Test MSE is 0.4251

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Fig. 4. Mean, Standard deviation and Quartiles