Regression, Cross-Validation, and Regularization: Comparing Regression Models

Cooper Golemme

February 11, 2025

1 Polynomial Regression

For this section of the report, I will report on the findings of fitting polynomial regressions to a fixed validation set. We will compare this model to other models using Ridge Regression and cross-validation.

1.1 Weights for Degree 1 Polynomial Fit

Following the pipeline that we established in the introduction section, I fit a 1-degree, linear regression, model to the training set and got the following weights:

$Features$	$horsepower$	$weight$	$cylinders$	$displacement$
$Coefficients$	-10.43	-18.23	-1.15	0.58

These weights indicate the expected change in efficiency, in miles per gallon, of changing the corresponding feature value. In the next section, we analytically show the relationships between the feautures and the predicted value.

1.2 Feature-Predict Relationship

1. Engine weight

   The coefficient for the engine weight is **-18.23**, which indicates that there is a negative relationship between weight and fuel efficiency. As the weight of the vehicle increases the mpg decreases. This makes sense with common sense; heavier vehicles take more energy to move and thus are usually less efficient in their fuel consumption. It also makes sense with the provided data. From a quick glance, some of the heaviest weight values have some of the least MPG.

2. Displacement

   The coefficient for the displacement is **+0.58**, which indicates a positive relationship between displacement and miles per gallon. This seems to be in conflict with common sense. Common sense would suggest that bigger

engines are more inefficient, weigh more, and have worse mileage. The model seems to suggest otherwise. Perhaps this makes sense, that bigger engines are more efficient as a result of being larger. There might be something about this dataset that could lead to this coefficient. Maybe it is testing highway MPG, in which case, it may be plausible that larger engines may be more efficient. It doesn't agree with common sense but could be valid depending on the dataset.

1.3 Higher Degree Polynomials

The weight values associated with the features in degree 4 regression are much larger, 2 orders of magnitude larger, than the weights associated with degree 1 regression. The degree of the regression directly effects the magnitude of the weights of the model.

1.4 RMSE vs Degree

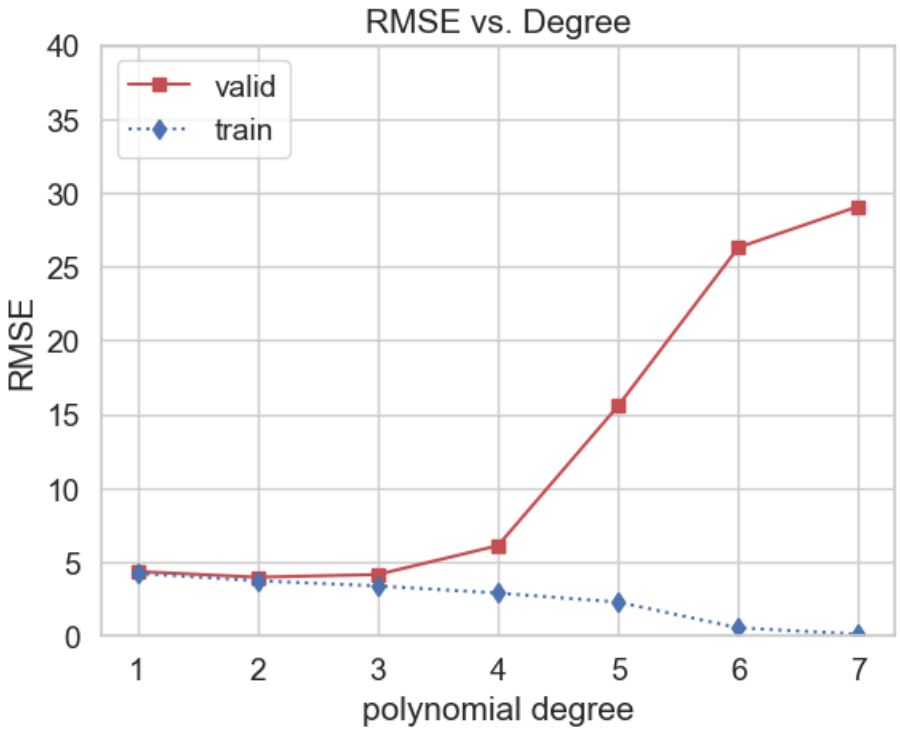

Figure 1: Root mean squared error on both the trainings set and the validation set based on the degree of polynomial fit in training. Degree 2 here seems optimal; it minimizes both training error and validation error. We notice overfitting for degree 4,5,6,7 polynomials. These degrees of polynomials suggest overfitting because the training error drops off almost to 0, while the validation error increases. This suggests that the model does not generalize well to new data, while it may preform very well on the test set.

We notice in the graph, for degree 6 and 7, the train RMSE drops off to almost 0. This means that the difference between the predicted MPG based on the feature vectors and the actual MPG of the training set is almost 0. In theory, this would mean our model is very good, almost perfect; we almost exactly predict the MPG based on the feature data.

The problem with higher degree polynomial fits is that after degree 4, the validation error increases substantially. This can be ascribed to the classic error of overfitting training data. When we overfit, our polynomial function may have be very close to the true values in the training data but may not necessarily be emblematic of the trends in the data, but rather fit nicely with the individual variation in a given training set. In effect, overfitting leads to

poor generalization; overfitting training data leads to increased error when we apply our model to new data points not seen in the training set.

We notice this idea of overfitting with higher degree polynomials (in the graph, degree 6 and 7). They perform very well on the training set – near 0 RMSE – but on the validation set, they have much higher error than degree 1 or 2 polynomial fits.

1.5 Without Pre-Processing

We will now examine what happens when we omit the preprocessing step where we scale the feature values to be within the interval 0-1. Below is a figure similar to Figure 1, but where the models are generated without the preprocessing step.

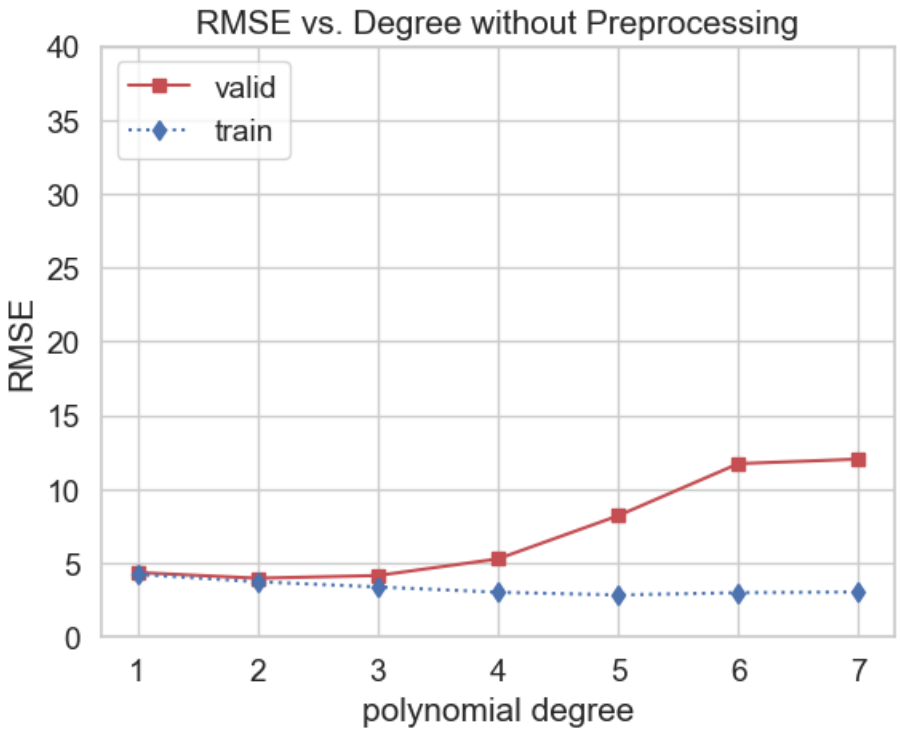

Figure 2: RMSE of different degree linear regression models without the pre-processing step. The red line is the error on the validation set and the blue line is the error on the training set.

At first glance, Figure 1 and Figure 2 look largely similar. To further illustrate the differences and demonstrate the importance of the pre-processing step, below is a graph of the difference in RMSE on the training set for each degree model with and without the pre-processing step.

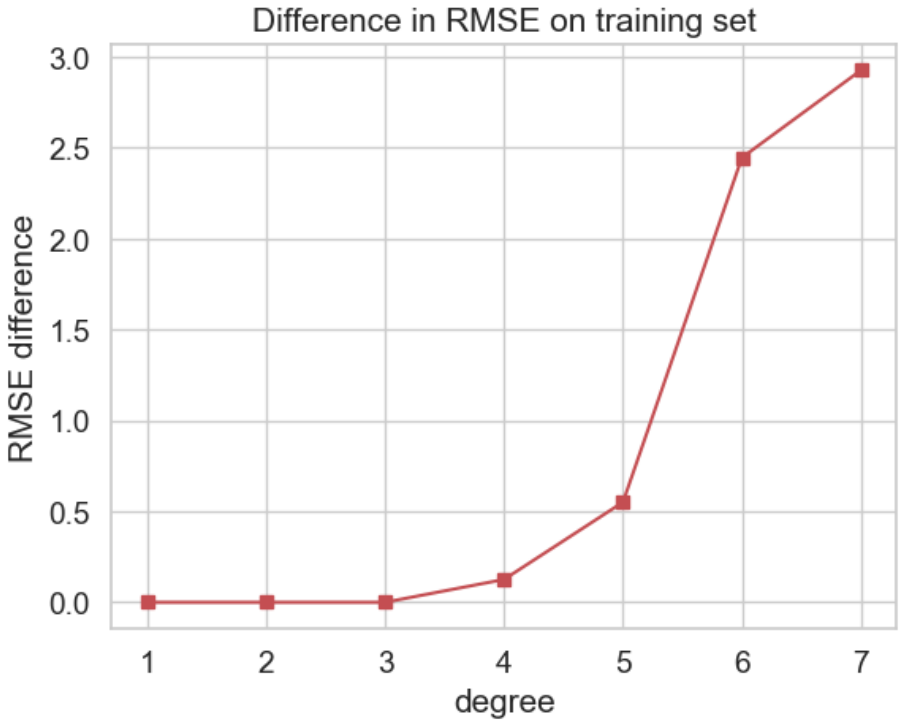

Figure 3: Difference in RMSE between the models without the preprocessing step and with the pre-processing step. Notice how sharply, the error increases for higher degree polynomials when the preprocessing step is omitted.

Degree	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Difference	0	-1.33e-15	6.96e-11	1.25e-01	5.51e-01	2.45	2.93

Table 1: Difference in training error with pre-processing and without. Positive values reflect when the not pre-processed RMSE is higher than the preprocessed RMSE

Without the preprocessing step, the training error does not decrease as much as with the step, when the degree of the polynomial gets higher. In Figure 1, we can see that the RMSE on the training drops precipitously around degree 6 and 7. In Figure 2, however, this is not the case; it mostly stays constant from degree 4 and onward.

Table 1 shows the difference in the training error rates of the models with the pre-processing step and without. With a degree 1 fit, the two error rates are basically indistinguishable (the same to roughly 17 decimal places).We can see the plotted results of this data in Figure 3. For degree 1, 2, and 3. The difference in training error rates is basically 0, but for higher degree polynomials

the difference sharply increases. Instead of near 0 training error in degree 7 polynomial fit with preprocessing, we have almost 3 error without.

Overfitting doesn't really give as a good explanation for the pattern that we see in Figure 3 and Table 1. Overfitting usually refers to the pattern that after some optimal degree fit, the difference between the validation error rate tends to increase and the training error rate drops as a result of fine tuning our model to fit the training data while sacrificing performance on unseen data. This concept does not really apply to this trend. The difference in the training error rates with and without the MinMaxScaler preprocessing step for higher degree polynomials (greater than 2 or 3) can more aptly be attributed to the phenomena that unscaled features create disparate scales, an effect which is particularly pronounced with higher degree fits. For each step of higher order polynomials, we run the risk of increasing the scale of features exponentially. For example, with degree 1 fits a value ranging from 1-10 will correspond to a predicted value ranging from 1-10, but when we move to degree 4, for example, a feature ranging from 0-10 becomes 0 - 10,000 as the upper end becomes 10^4.

If we rescale our features to be within 0-1, the problem no longer presents an issue. A value between 0-1 will still be within 0-1 regardless of the degree we raise it to. The results in a similar range for all of the features which stabilizes the optimization and results in lower training error as we increase the polynomial degree fit. This phenomena of numerical issues when not rescaling the data better characterizes the trend we observe in Table 1 rather than "overfitting".

2 Penalized Polynomial Regression

2.1 Error vs. alpha

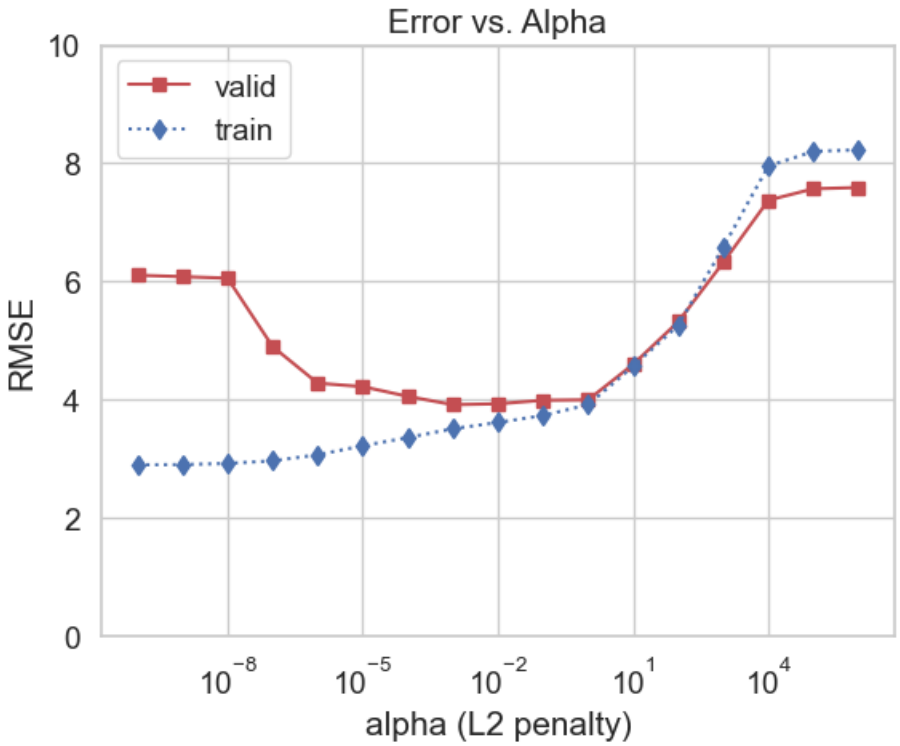

Figure 4: RMSE of the model given different α scale L2 penalties. Notice how initially, at the left most data points, the training error rate is smallest, with a large gap between the training error and the validation error. This can be ascribed to high weight parameters not being penalized with such a small α, resulting in over fitting which performs well in training but poorly in validation. On the other end, when we penalize too much, both the training and validation error rates are high as a result of penalizing potentially good models unnecessarily for their feature weight values. In the middle, we find an optimal α around 10^{-4} where we achieve the best validation error. This is the value of α that I would recommend to deploy new instances to get the lowest possible error rate as it is the α which preforms the best on the new unseen data in the validation set.

2.2 Weight Parameters for Degree 4

$617.45 : x_0$	$-36.60 : x_0$
$-191.08 : x_1$	$18.53 : x_1$
$1774.24 : x_2$	$10.40 : x_2$
$-1343.56 : x_3$	$-24.68 : x_3$
$159.25 : x_0^2$	$-38.19 : x_0^2$
$-323.66 : x_0x_1$	$-56.65 : x_0x_1$
$-5179.64 : x_0x_2$	$26.08 : x_0x_2$
$198.45 : x_0x_3$	$70.53 : x_0x_3$
$61.37 : x_1^2$	$-52.05 : x_1^2$
$2573.25 : x_1x_2$	$10.30 : x_1x_2$
$-1596.95 : x_1x_3$	$-8.06 : x_1x_3$
$-5943.40 : x_2^2$	$-4.93 : x_2^2$
$7359.52 : x_2x_3$	$-98.70 : x_2x_3$
$2883.15 : x_3^2$	$62.87 : x_3^2$
$-630.58 : x_0^3$	$36.80 : x_0^3$
$1377.51 : x_0^2x_1$	$24.02 : x_0^2x_1$
$-1600.47 : x_0^2x_2$	$9.97 : x_0^2x_2$
$1529.25 : x_0^2x_3$	$53.86 : x_0^2x_3$
$-105.12 : x_0x_1^2$	$52.10 : x_0x_1^2$
$-7880.76 : x_0x_1x_2$	$12.10 : x_0x_1x_2$
$12085.95 : x_0x_1x_3$	$62.16 : x0x1x3$
$13051.50 : x_0x_2^2$	$29.58 : x_0x_2^2$
$5626.91 : x_0x_2x_3$	$-28.89 : x_0x_2x_3$

Above are the weight parameters for a degree 4 fit polynomial. The left are the weights for the degree-4 model from problem 1 without the L2 penalty, and the right are the weights for the model with the penalty. From the data, we notice that the model on the right has significantly smaller weight values than the model on the left. This is because the L2 penalty penalizes larger weight values in the optimization problem, leading to a model that has smaller weights.

2.3 A new way to calculate alpha

Your colleague suggests that you can determine the regularization strength alpha by solving the following problem on the training set:

$$\hat{w}, \hat{b}, \hat{a} \leftarrow \text{argmin}_{w \in \mathbb{R}^F, b \in \mathbb{R}, \alpha \geq 0} \sum_{n=1}^{N} (y_n - \hat{y}(x_n, w, b))^2 + \alpha \sum_{f=1}^{F} w_f^2$$

What numerical value of $\hat{\alpha}$ would this strategy favor? (Hint: you aren't limited to the grid in 2A). Why is this problematic if your goal is to generalize to new data well?

If we select alpha based on this formula, the model will always favor $\alpha = 0$ because the weight term on the right is always increasing as it is the sum of positive squared terms. Selecting alpha like this, treats the value as another feature in the optimization problem rather than a hyper parameter that we determine by examining training and validation error data. It has the effect of not including the term at all, and results in un-regularized linear regression like problem 1 of this paper. This is problem because small changes in the training data can drastically change the learned weights, making the model preform poorly on new data.

To properly choose alpha, we must choose it not alongside w and b as a part of the training process, but as a free hyper parameter after training has completed. If we do not, then we preferentially bias lower alpha values and do not achieve any meaningful results.

2.4 Comparing All Models

Model Name	Hyperparameters	Test RMSE
Mean		7.104
LR (1D)	degree=2	3.992
Ridge LR (2B)	alpha=0.01	3.878
Ridge LR (3B)	degree=7, alpha=0.1	3.817

Table 2: Comparison of test set RMSE for four regression models. The Problem 3B model likely outperforms Problem 2B due to the fact that it optimizes both the degree and alpha at the same time, avoiding suboptimal fixed assumptions (like how we assume degree 4 in 2B). It also likely outperforms due to 10-fold CV's reduced variance in error estimation compared to a single validation split. Using cross validation decreases error rates as it effectively represents a larger, less varient training set. Directly minimizing test set error during hyperparameter selection would lead to overfitting and unreliable generalization on new data sets, like how we see overfitting of the training data leads to generalization problems with high degree polynomials in Figure 1. The test set must be separate from parameter selection to avoid issues of overfitting and fitting specicially to the testing data.