Statistical Methods in AI

Assignment-7 Report

Analyse how the hyper-parameter λ plays a role in deciding between bias and variance.

- Linear Regression with multicollinear data has very high variance but very low bias in the model which results in overfitting. This means that our estimated values are very spread out from the mean and from one another. The model captures the noise and outliers in the dataset along with the underlying patterns which result in overfitting of the data.
- Similarly, if we have low variance and high bias in our model then it would result in underfitting of the data. This means that model is unable to find the underlying patterns within the dataset. These models are usually simple models.
- For Linear Regression with multicollinear data, a small increase in bias can result in a big decrease in the variance and which would result in a substantial decrease in Predicted error.
- One of the most common methods to avoid overfitting is by reducing the model complexity using **regularization**.
- When the β coefficients are unconstrained they can tend to have high value (in multicollinear cases), which would result in very high variance in the model. In order to control the variance, we add a constraint to the beta coefficients while estimating them. This constraint is nothing but the penalty parameter given by **λ (lambda).** This type of regression where we add penalty parameter λ in order to estimate the β coefficients is called **Ridge and Lasso regression**.

  Bias-Variance Concept depending upon lambda:
  When we have very less data points to train the model overfits the given training data set and has high variance. Using regularization technique we

can reduce this overfit by penalizing the regression coefficients by a factor lambda(λ). This results in reduction in the variance and increase in bias which may not fit the training set perfectly but may result better on testing set.

The main idea behind regularization is to find a new line, that doesn't fit the training data very well and introduce a small amount of Bias so that it fits the test data.

This addition of bias is controlled by the Lambda term by minimizing the term 'sum of squared residuals + λ* slope² ' in case of ridge regression and 'sum of squared residuals + λ*| slope|' in case of Lasso Regression.

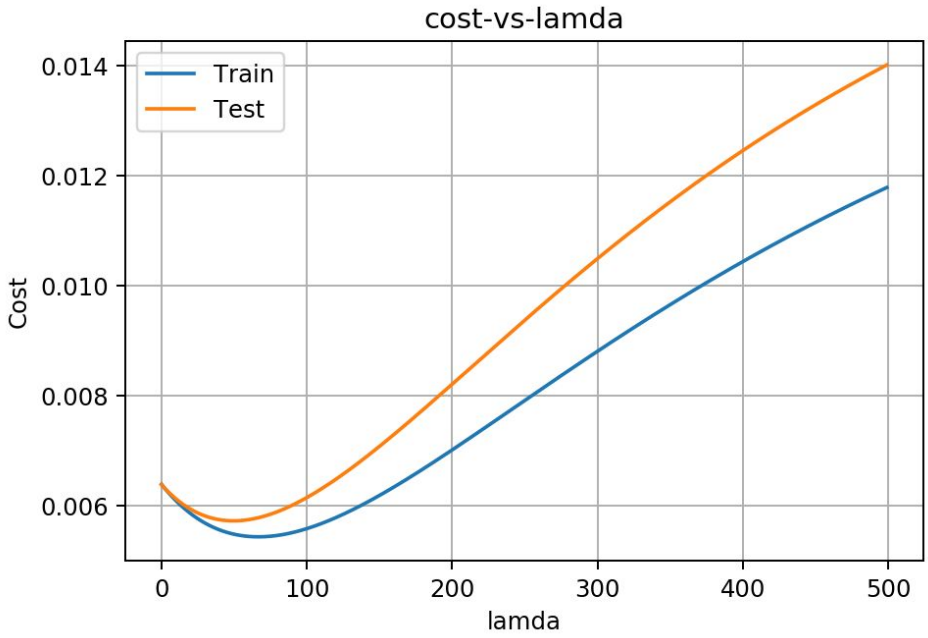
Thus we can conclude, Ridge/Lasso Regression Line, which on increasing small amount of bias due to penalty, has less variance.

1. Ridge Regression

Ridge regression is an extension of Linear regression. It is a regularization method which tries to avoid overfitting of data by penalizing large coefficients. Ridge regression has an additional factor called λ (lambda) which is called the penalty factor which is added while estimating beta coefficients. This penalty factor penalizes high value of beta which in turn shrinks beta coefficients thereby reducing the mean squared error and predicted error.

- λ value will always be greater than 0 (between 0 and infinity) and this parameter controls the amount of shrinkage.
- The penalty parameter is applied β1…βp and not to the intercept β0 since it is simply the mean value of the response variable.
- When λ = 0, the regression will be similar to linear regression and the β coefficient estimates will be similar to linear regression β coefficients.

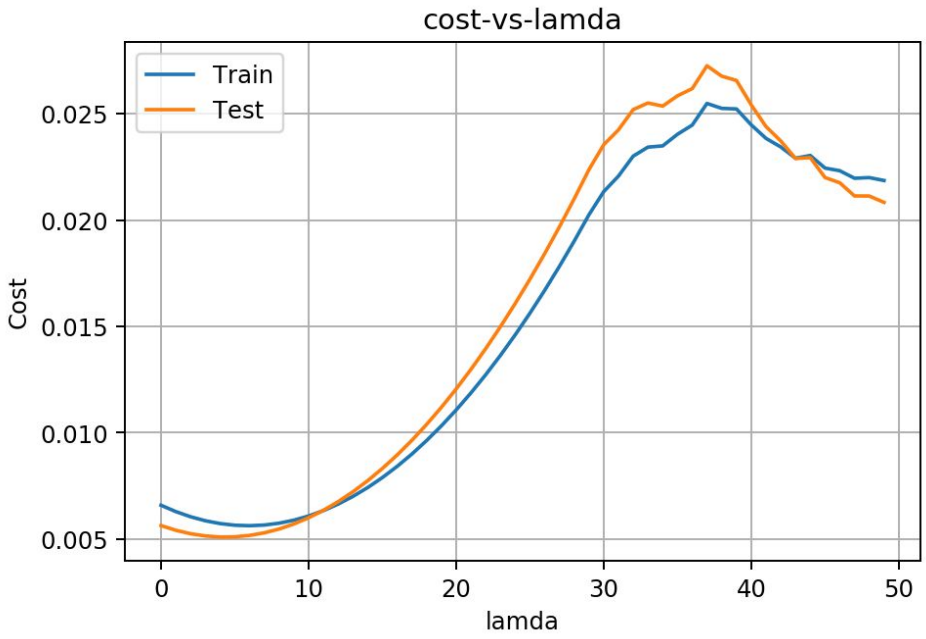
So, when the value of λ is very small, it has less effect on the parameters, and hence there will be overfitting. On increasing the value of λ.



From the graph we can see that the train error is always less than the validation error which is so obvious, because the theta is obtained by the training data itself. Adding Lambda term in the cost and minimizing it means, bias is getting added, hence the variance will be dropped. This will lead to avoidance of overfitting. And hence initially, on increase of Lambda, the error decreases. On further increase in Lambda, underfitting occurs.

2. Lasso Regression

This is similar to Ridge Regression, except the Lambda term.



Analyse how the two different regularisation techniques affect regression weights in terms of their values and what are the differences between the two.

Difference between Ridge and Lasso Regression

When lambda is 0, then ridge and lasso regression, will be same as the Least Square Line. As lambda increases in value, the slope gets smaller until the slope equals to 0.

In case of Ridge Regression we minimize :
The sum of squared residuals + (Lambda * (sum of slope^2))

Now if there are some important parameters, while some unimportant parameters, so on increasing the Lambda value,the parameters that have more effect, shrink less, while that in case of less important parameters, the less important parameters shrink a lot near to 0, but not 0.

In case of Lasso Regression we minimize :

The sum of squared residuals + (Lambda * (sum of |slope|))

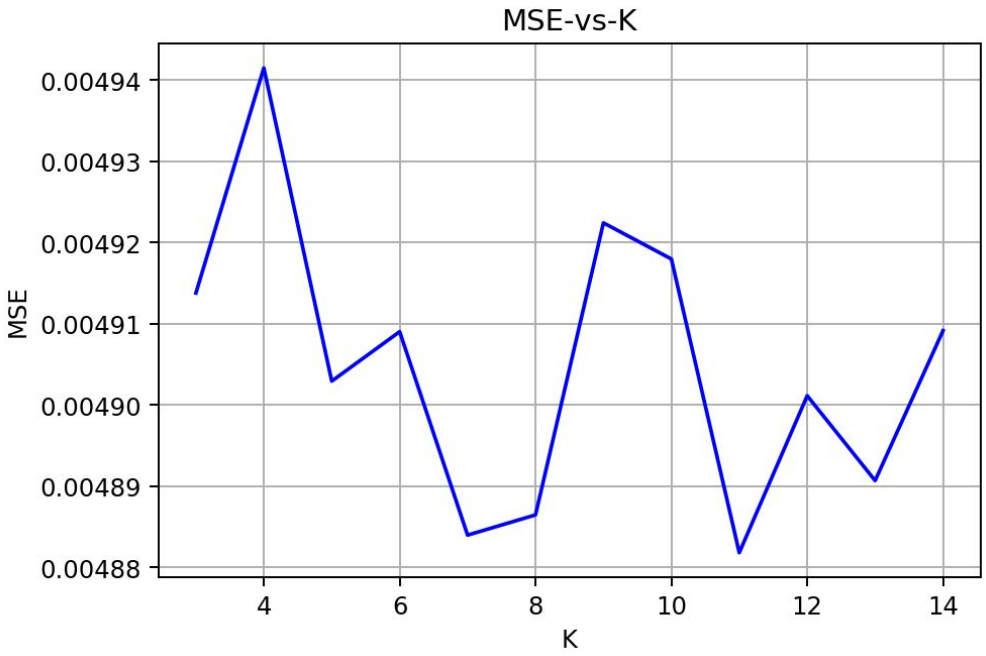
For the same case as above, the less important parameters will shrink to 0, and we are left with only the important parameters. The reason is, in this case we have to minimize the sum of |slope|, while in case of ridge regression we have to minimize the sum of |slope^2|.

From the above equation, it's clear that Lasso Regression can exclude useless variables from equations, so, it's little better than Ridge Regression at reducing the variance in models that, contain a lot of useless variables.

Cross Validation

Cross-validation, it’s a model validation techniques for assessing how the results of a statistical analysis (model) will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

The goal of cross-validation is to define a data set to test the model in the training phase (i.e. validation data set) in order to limit problems like overfitting,underfitting and get an insight on how the model will generalize to an independent data set. It is important the validation and the training set to be drawn from the same distribution otherwise it would make things worse.



Leave-one-out cross validation

Leave-one-out cross validation. LOOCV uses a single observation from the original sample as the the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. This is the same as a K-fold cross-validation with K being equal to the number of observations in the original sample. Leave-one-out cross-validation is usually very expensive from a computational point of view because of the large number of times the training process is repeated.

Mean Error : 0.0048