CSE 574 Introduction to Machine Learning
Programming Assignment 2: Classification and Regression

Team No: 63

Team Members :

Achuth Narayan Rajagopal	5020 6533
Kedar Paranjape	5020 5932
Anjana Guruprasad	5020 5233

1. Experiment with Gaussian Discriminators

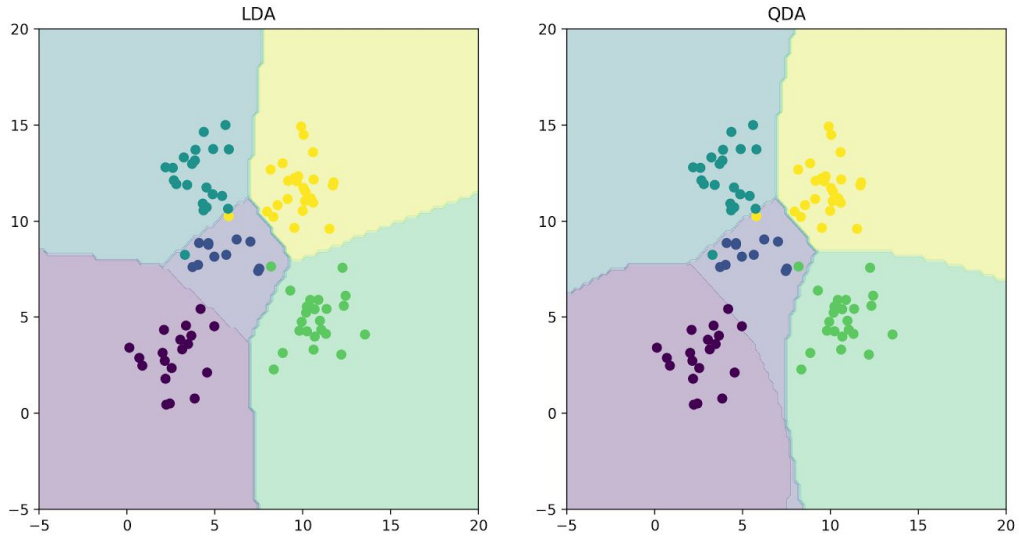

Fig: *Discriminating boundary for linear and quadratic discriminators*

The LDA and QDA accuracy results obtained for the provided test data set are:

LDA - 97%

QDA - 96%

- LDA and QDA have similar assumptions except for the covariance matrix. In LDA, the covariance matrix is common for all the k classes while in QDA each class has a different covariance matrix.
- In case of the discriminating boundaries we see a difference due to the fact that LDA learns linear boundaries while QDA, quadratic. This implies that QDA has more flexible decision boundaries.
- QDA allows for more flexibility for the covariance matrix but the number of parameters increases significantly. QDA can thus be used for classification if there is a large difference in the covariance of each class.

2. Experiment with Linear Regression

The results of running linear regression on both the training and test data with and without intercepts is shown below:

	Train data	**Test data**
MSE without intercept	19099.44684457	106775.36155856
MSE with intercept	2187.16029493	3707.84018153

Using linear regression without regression forces the regression line to pass through the origin. If our fitted line does not naturally go through the origin, our predictions will be biased if don't include the constant. Also, as evident from the table above, using an intercept results in the MSE to reduce by a factor of about 8.73 and for training data . For test data, the reduction is even more visible by a factor of about 28.

3. Experiment with Ridge Regression

- Running ridge regression on the training data results in an almost linear increase in the MSE.
- Furthermore, the minimum MSE is obtained when lambda is zero. This means that regularization does not help reduce the MSE for train data at all.
- However, ridge regression when run on the test data, shows a sudden drop in MSE even with infinitesimally larger lambda values.
- The minimum MSE is obtained when lambda is 0.06, beyond which it linearly increases.
- Also, the minimum MSE for test data starts out higher than that for train data and stays higher until the very end. That would happen when the train data is not completely representative of the actual test data.
- The MSE obtained via Ridge Regression for train data is the same as the one obtained via Linear Regression using intercepts. For test data however, the MLE with ridge regression is higher than the one obtained with intercepts using the linear regression approach. However, ridge regression performs better than linear regression without intercepts for both datasets.

	Train data	Test data
Minimum MSE	2187.16029493	2851.33021344
Corresponding lambda	0	0.06

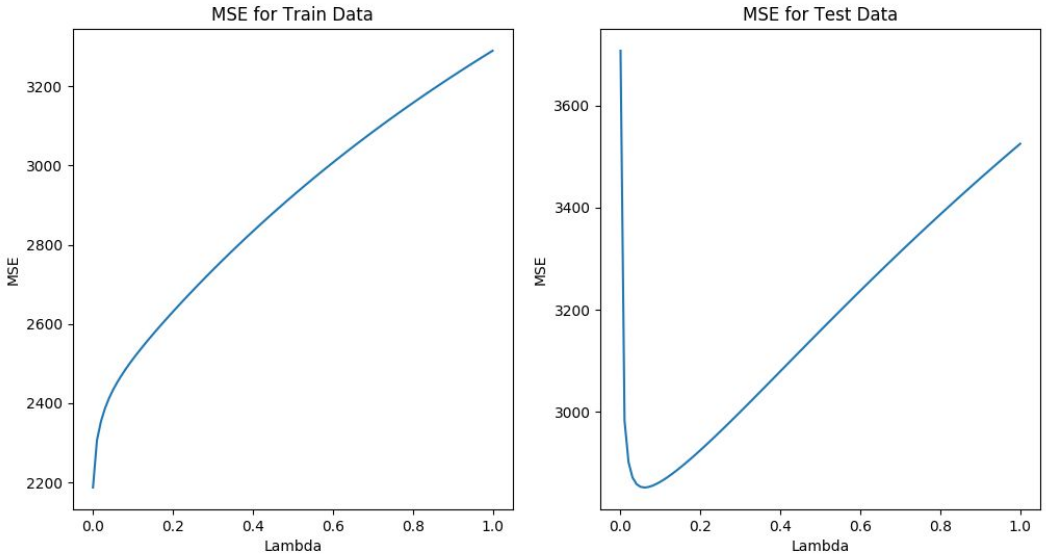

Fig: *Ridge Regression*

Relative magnitudes of weights learnt: Linear regression v/s Ridge Regression.

The L2 norm using linear regression with intercepts on test data is 3.91112086e+12 whereas the same norm using Ridge Regression corresponding to the minimum lambda on test data is 920281.35693682. This implies that using Linear Regression has a higher magnitude of weights compared to Ridge Regression, this could be a factor leading to overfitting and ultimately bad test accuracy. This is validated by the results which show that Linear Regression with intercept has a higher MSE than Ridge Regression. This is another reason why ridge regression performs better than linear regression

4. Using Gradient Descent for Ridge Regression Learning

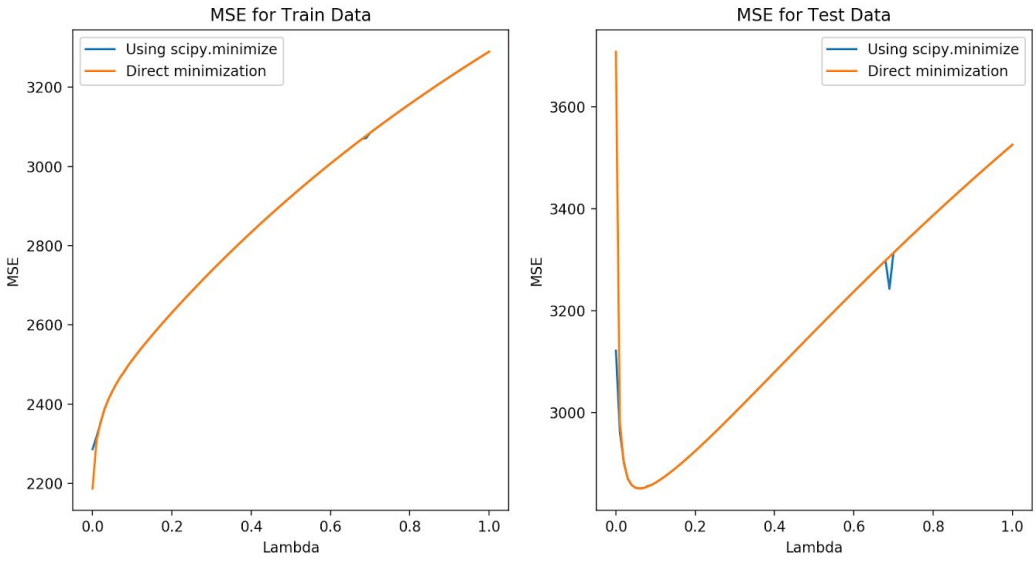

Fig: *Ridge Regression using Gradient Descent*

	Train data	**Test data**
Minimum MSE	2278.6106169	2851.29771954

Observations:

- For the given dataset, the weights obtained from the Gradient Descent for Ridge Regression Learning are nearly identical to those obtained using matrix multiplication in problem 3.
- There are minor variations in the values but those can be ruled out as outliers.
- The execution time was nearly identical for both the methods for this dataset but that might not hold true as the data size goes up as matrix inverse and multiplication can become computationally intensive.
- The Gradient Descent implementation would be ideal for bigger datasets despite the presence of a few outliers.

5. Non-linear Regression

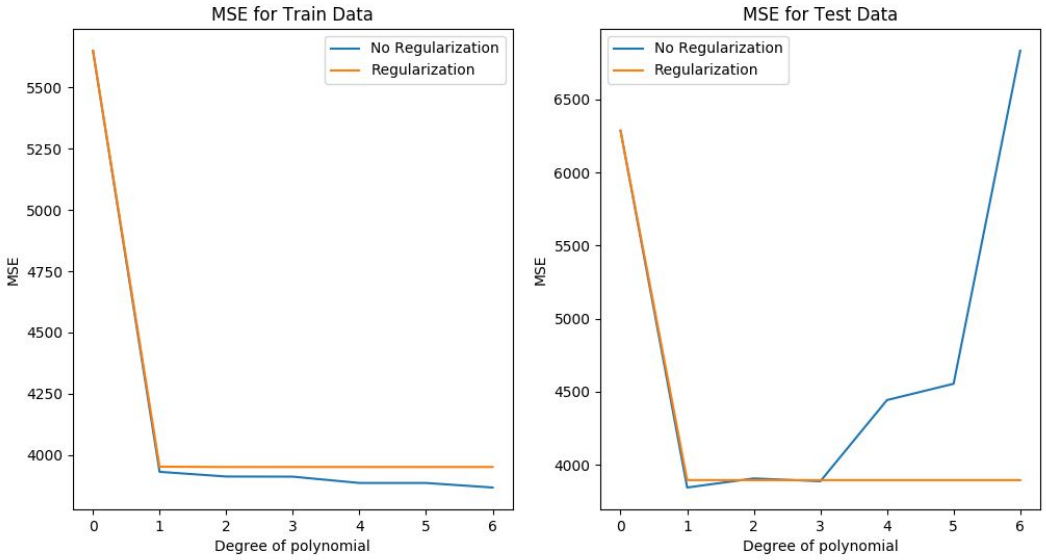

Fig: *Nonlinear Regression*

Observations:

- For the given problem, we have compared the MSE values obtained for training and test data for different higher degree polynomials with and without regression.
- For train data, with no regularization, MSE linearly drops with increase in degree of polynomial. On the same data, with regularization, the MSE stays constant beginning from p = 1
- For test data, with no regularization, MSE increases greatly after dropping to a minimum at p=1. On the same data, with regularization, the MSE keeps decreasing infinitesimally beginning from p = 1
- For training data, the error slowly decreases in the case of with and without regularization. This can be attributed to the fact that higher order polynomials have complex curves that can fit better.
- In the case of test data, the regularized execution has similar output to that of the training data. This is due to the fact that lambda in regularization keeps the weights in check and avoids the problem of overfitting.

- In the case of unregularized data, no such check exists this leads to some weights being much higher than others leading to overfitting. This is one of the main reasons for the high test error.

6. Interpreting results and Conclusion:

Method	Train data	Test data
Ridge Regression (with Regularization)	2187.16029493	2851.33021344
Ridge Regression (Gradient Descent)	2278.6106169	2851.29771954
Linear Regression (using intercepts)	2187.16029493	3707.84018153
Linear Regression (without intercepts)	19099.44684457	106775.36155856

- So far in the project we have implemented different methods of learning and prediction on the given training and test dataset. The next important task is to identify which method is appropriate for a real world implementation.
- The comparison of different methods can be done using the training error, the test error or the running time.
- In the case of training error, non-linear regression with a high degree polynomial will give us the best results but the real metric for comparison should be the comparison with new and unseen data which is given to us by testing error.
- Considering the testing error, the minimum value we obtain is using Ridge regression. The method of implementation slightly varies the result.
- Given that there is a time constraint, we have to consider an approach that is efficient and gives good results. When we have much larger data sets, calculation of $(X^TX)^{-1}$ could get computationally intensive. In this case, gradient descent might be a better alternative. Ridge Regression with Gradient Descent would be the ideal approach for this case compared to direct calculation.
- In the case of real world implementation, Ridge Regression with Gradient Descent and Regularization would be the ideal candidate.