UNIVERSITY AT BUFFALO

CSE574 - Introduction to Machine Learning

Programming Assignment 2 – Report (Group 10)

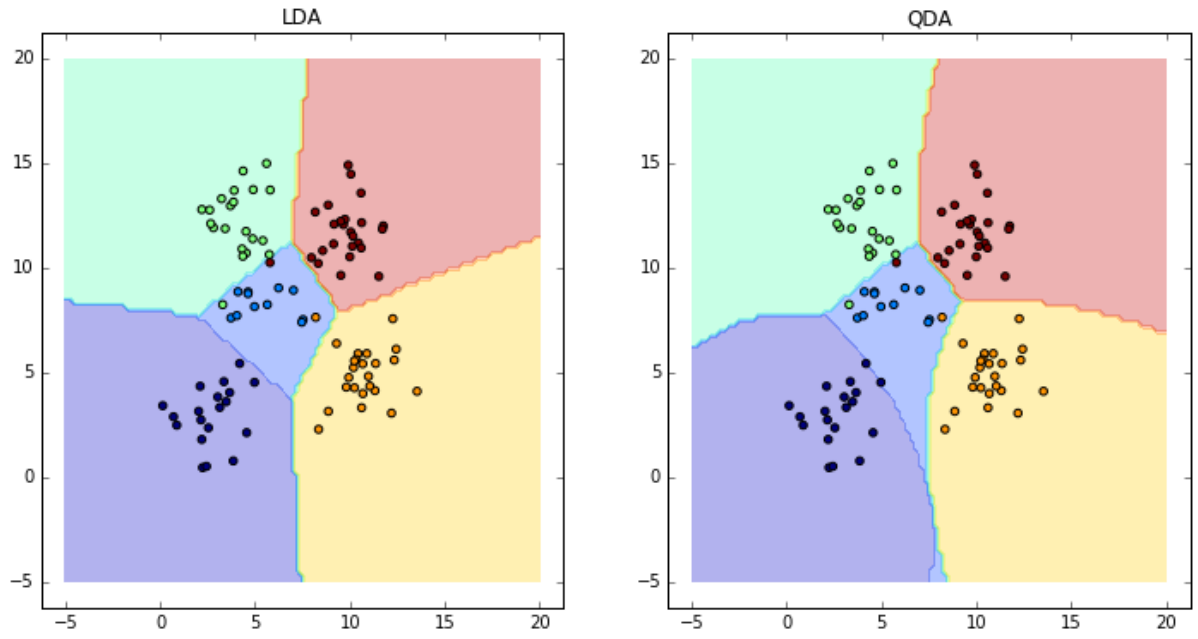
Aayush Shah (50207564)
Siddharth Shah (50205787)
Haril Satra (50208283)
4/12/2017

Experiment with Gaussian Discriminators

Accuracy

LDA: 97 %
QDA: 96 %

Plot



Analysis

In LDA (Linear Discriminant Analysis) we assume that the covariance matrix for all the different output classes is the same. This results in a linear classifier. On the other hand, in QDA (Quadratic Discriminant Analysis) the covariance matrix for each output class is different.

This quadratic discriminant function in QDA is similar to the linear discriminant function except that the covariance matrix is different for different classes. Hence the quadratic terms cannot be ignored. Thus the discriminant function in QDA results in a quadratic function containing second order terms. Hence there is a difference between the boundaries of the two classifiers. LDA has linear boundaries while QDA has non-linear boundaries due to the second order terms in the discriminant function.

Experiment with Linear Regression

MSE (Mean Square Error)

Training Data:

- Without Intercept: 19099.44684457
- With Intercept: 2187.16029493

Testing Data:

- Without Intercept: 106775.36155355
- With Intercept: 3707.84018134

Analysis

If we do not add the intercept, we impose a restriction on the regression line to pass through the origin leading to the fit getting heavily pulled down (Systematically shifted towards larger or smaller values).

Thus usually linear regression with intercept will be a better fit and provide a smaller MSE as compared to linear regression without intercept. This can also be observed in the above results that are obtained.

The only reason to use Linear regression without using a intercept is when it is known that the process has a model with zero intercept.

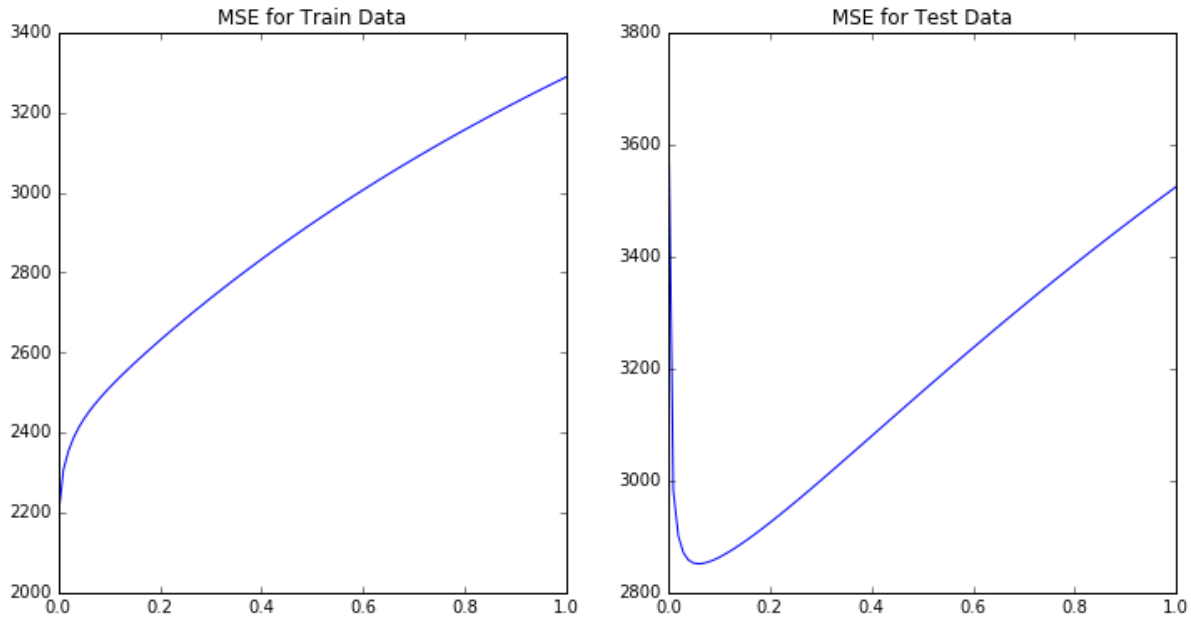
Experiment with Ridge Regression

MSE (Mean Square Error) for Ridge Regression for optimal lambda (0.06).

Training Data (With intercept): 2451.52849064

Testing Data (With Intercept): 2851.33021344

Plot



Comparison of relative magnitudes of weights learnt using Linear Regression and Ridge Regression

In the case of linear regression for weights learnt, we get extreme values (large positive or large negative values). Whereas in case of ridge regression or l2 norm regression the values are regularised and have lesser variance.

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\lambda\mathbf{w}^\top\mathbf{w}$$

Considering the above equation, we can see that if weights are very large then the sum of square error term will minimize but the penalty will increase. However, if weights are less, then the penalty term will minimize but square error will increase.

We can validate this with our results as the extreme weights in linear regression are regularized.

Comparison of Linear Regression and Ridge Regression in terms of errors on train and test data.

	Train Data	Test Data
Linear Regression	2187.16029493	3707.84018134
Ridge Regression	2451.52849064	2851.33021344

Linear Regression often faces the issue of overfitting on the training data. But this fit might not necessarily be good for the test data.

This can be overcome by introducing a penalty term. Ridge regression introduces this penalty term as l2 norm of the weights, leading to generalization of the fit or in other terms reduces the complexity of the fit. This may increase the error on the training data but will usually perform better on the testing data as compared to Linear Regression.

Parameters

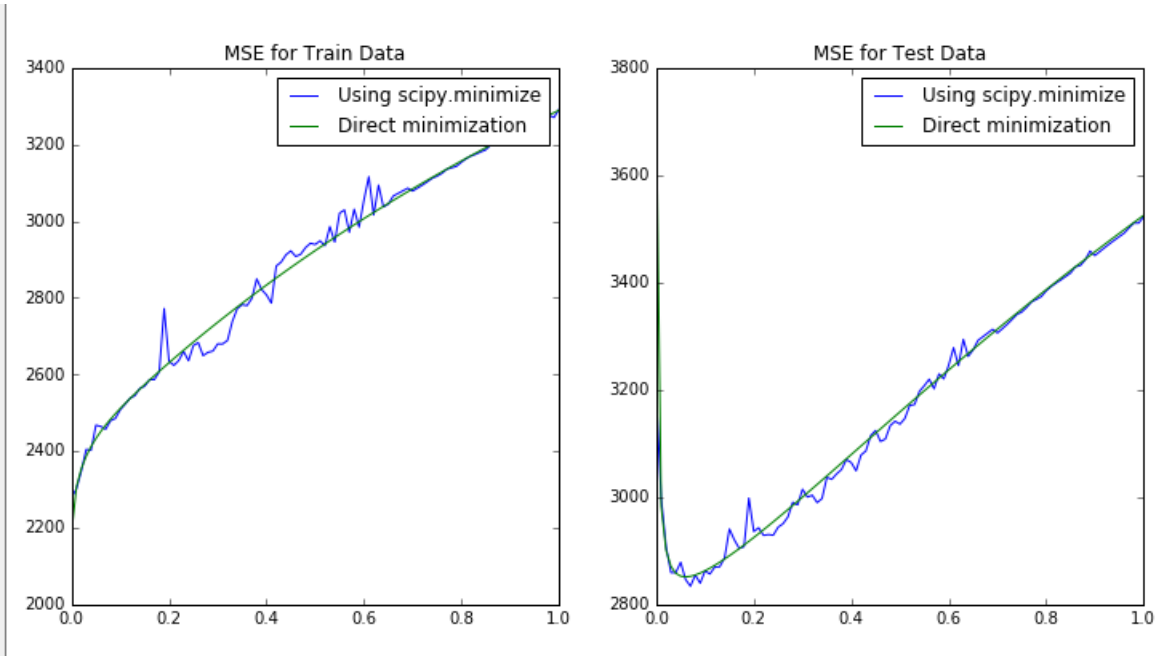
Lambda: The optimal value of lambda which we obtained is 0.06.

With growing lambda, the training error increases as the residual sum of squares become larger. However, in case of test data the training error reaches a minimum for a specific value of lambda after which it again increases. The optimal value of lambda suggests that it is a good generalization fit.

There is no fixed value of lambda which provides the minimum error for any data. It varies from data to data and the best way to obtain is to vary it in small steps starting from 0 (no regularization) up to some higher value.

Using Gradient Descent for Ridge Regression Learning

Plot



What gradient descent basically does is that it starts from some point and moves downhill to find the point with the lowest error. The output of gradient descent will eventually (if enough iterations are provided) be the same as that of methods used in problem 2 and problem 3.

Non-Linear Regression

Comparison of the results

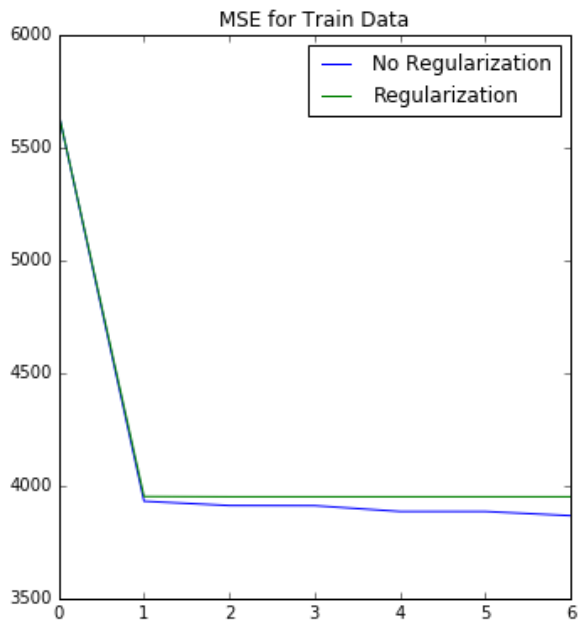
Training Data:

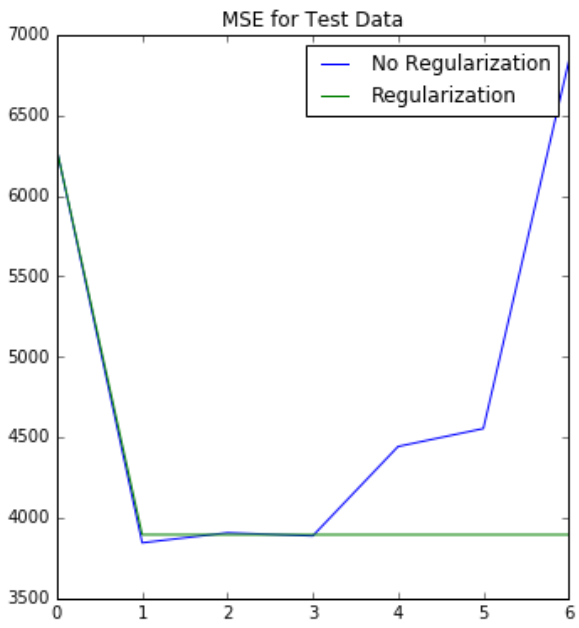
P	NO REGULARIZATION ($\lambda = 0$)	WITH REGULARIZATION ($\lambda = 0.06$)
0	5650.7105389	5650.71190703
1	3930.91540732	3951.83912356
2	3911.8396712	3950.68731238
3	3911.18866493	3950.68253152
4	3885.47306811	3950.6823368
5	3885.4071574	3950.68233518
6	**3866.88344945**	**3950.68233514**

Test Data:

P	NO REGULARIZATION ($\lambda = 0$)	WITH REGULARIZATION ($\lambda = 0.06$)
0	6286.40479168	6286.88196694
1	3845.03473017	3895.85646447
2	3907.12809911	3895.58405594
3	**3887.97553824**	3895.58271592
4	4443.32789181	**3895.58266828**
5	4554.83037743	3895.5826687
6	6833.45914872	3895.58266872

Plot





Analysis

We can clearly see from the results and the plots that non-linear regression without regularization leads to overfitting of the model for the training data but performs poorly on the testing data. Thus the MSE on training data is very less as compared to the testing data.

Regularization helps to solve this issue by generalizing the model or in other words reducing the complexity of the fit. This can also be seen from the results and plots that non-linear regression with regularization reduces the complexity of the fit and performs a bit poorly as compared to non-linear regression without regularization. However non-linear regression with regularization performs much better than its counterpart overall (especially for larger values of p). Thus the MSE on training and testing data is quite similar.

Comparison of above approaches

Sr No.	Approach	MSE for Train Data	MSE for Test Data
1	Linear Regression	2187.16029493	3707.84018134
2	Ridge Regression	2451.52849064	2851.33021344
3	Ridge Regression Using Gradient Descent	2279.25957418	2834.01266301
4	Non Linear Regression (With Regularization)	3950.68233514	3895.58266828

Linear Regression:
It establishes a relationship between dependent variable and one or more independent variables using a best fit straight line.

This model is too simplistic meaning even if the data is non-linear it will try to fit it using a straight line.
Sometimes in trying to represent the entire data which may even include the noise or outliers it results in an overly complex fit. This leads to low MSE for train data but incorrect hypothesis or high MSE for test data.
Another issue is that it is unstable in presence of correlated input attributes.

Ridge Regression:
Ridge regression solves one of the major issue of linear regression which is correlated input attributes. The issue with collinearity is that the variance of the parameter estimate is huge. Ridge regression reduces this variance at the cost of introducing a bias to the estimates.

Thus we can see in the results that ridge regression gives a slightly higher error on the training data but provides a much better performance on the test data as compared to the linear regression.

Thus ridge regression is often used when the independent variables are colinear.

Ridge Regression using Gradient Descent:

$$\widehat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

In ridge regression we use the above equation to find the w which minimises the error. This equation involves the calculation of Inverse of a matrix. However, there is no guarantee for the above system of equation to be computable. For this the matrix for which we take the inverse should be a non-singular matrix. In other words, it's rank should be d, it cannot be less than d. This cannot be guaranteed in real life applications. So instead of using the above equation to find the weight which minimises the error we use the gradient descent approach.

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

This is a much feasible computation as compared to computing the Inverse.

Thus the results will be quite similar if we allow it to minimize through large number of iterations.

Non Linear Regression:
We know the issue of underfitting in linear regression since it cannot model non-linear curves it may result in large error. This can be overcome using non-linear regression.
But there can be infinite number of functions so it can be more difficult to find the function that finds the optimal fit.

Another issue of non-linear regression is that the effect feature has on the result can be less intuitive to understand.

Hence linear regression should always be tried first but if we don't get a very good fit then we should try out non-linear regression.

Metric

The Mean Square Error (MSE) should be used as a metric to choose the best setting.
MSE of an estimator measures the average of the squares of the errors or deviations. That is, the difference between the estimator and what is estimated.
The smaller the value of MSE the better the estimator.

References

[1] http://statistiksoftware.blogspot.com/2013/01/why-we-need-intercept.html

[2] https://www.quora.com/What-are-the-benefits-of-using-ridge-regression-over-ordinary-linear-regression

[3] http://stats.stackexchange.com/questions/108364/demonstration-of-benefits-of-ridge-regression-over-ordinary-regression

[4] http://blog.minitab.com/blog/adventures-in-statistics-2/linear-or-nonlinear-regression-that-is-the-question