CSE-574 Introduction to Machine Learning

Assignment #2

Classification and Regression

Group #41
Pruthvi Mulagala – 50208595
Madhav Jakkampudi - -50206563
Vinod Kumar Veparala – 50208038

PROBLEM 1

EXPERIMENT WITH GAUSSIAN DISCRIMINATORS

Observed LDA Accuracy – **97%**
Observed QDA Accuracy – **96%**

The discriminating boundary for linear discriminators and quadratic discriminators

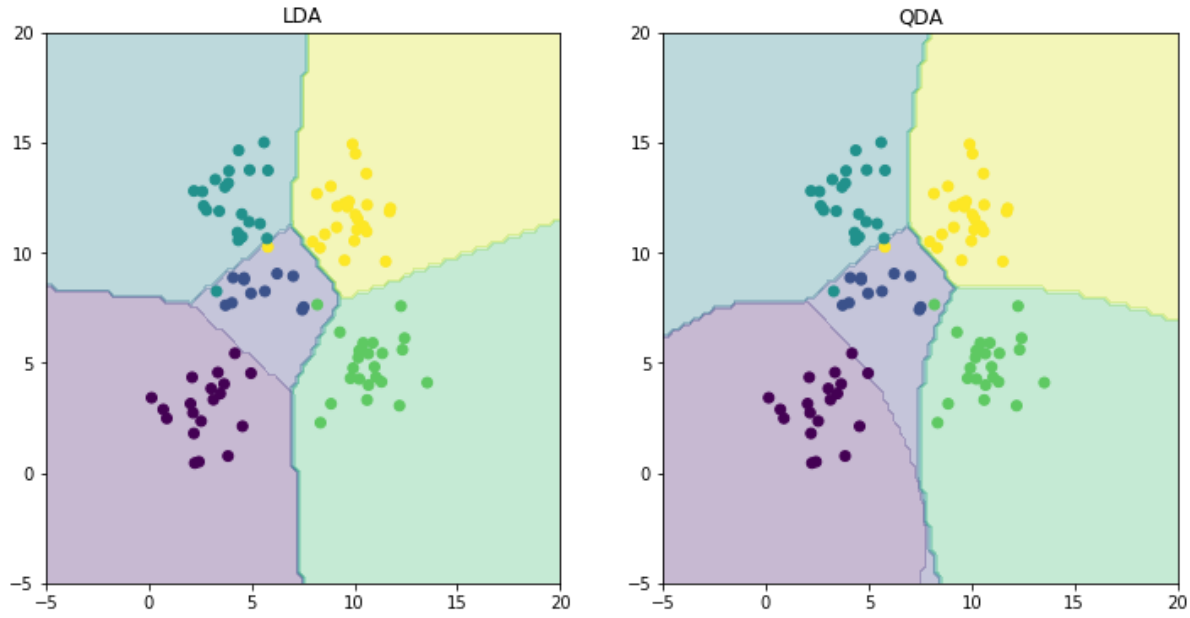


Fig 1.1

Conclusion:

The two plots, LDA and QDA differ in the covariance computed for the input data resulting in different discriminating boundaries. In LDA, the covariance is identical for all the classes and so the classifier becomes linear whereas in QDA covariance differs for each class and hence the decision boundary is determined by a quadratic function due to which the boundaries appears to be curved. Even though LDA here slightly outperform QDA in terms of accuracy, it is worth noting here that the data we used is quite simple and in the cases of classifying data for complex data sets, QDA would perform better over LDA.

PROBLEM 2

EXPERIMENT WITH LINEAR REGRESSION

MSE values for OLE regression

MSE Computed	Without Intercept	With Intercept
Training data	19099.44684457	2187.16029493
Test data	106775.36155789	3707.84018132

Table 2.1

The MSE values obtained highlight two positive effects of using an intercept for prediction:

- A significant error decrease when considering a single data set in isolation (either training or test)
- An even more impressive reduction of the error committed on the test set compared to the training set.

When using an intercept MSE on training data is reduced by close to 7.7 times its value, similarly the error on the test data reduces by close to 27.8 times its value. Moreover, when comparing the MSE for training and test data, we can see that not using an intercept causes the test error to be 5.59 times the error on the training set. However, when using an intercept, the test error is 1.69 times the MSE on the training set.

PROBLEM 3

EXPERIMENT WITH RIDGE REGRESSION

Given below are the list of lambdas and their corresponding MSE for training data and the test data by increasing the value in the increments of 0.01 of lambda. The highlighted lambda – 0.05989899 is the value for which we attained the minimum value of MSE.

lambda	mses3_train	Mses3
0	2187.160295	3707.840181
0.01	2306.832218	2982.44612
0.02	2354.071344	2900.973587
0.03	2386.780163	2870.941589
0.04	2412.119043	2858.00041
0.05	2433.174437	2852.665735
0.05020202	2433.568118	2852.606772
0.05040404	2433.960728	2852.549366
0.050606061	2434.352275	2852.493498
0.050808081	2434.742769	2852.439152
0.051010101	2435.132218	2852.38631
0.051212121	2435.520631	2852.334955
0.051414141	2435.908016	2852.285071
0.051616162	2436.294383	2852.236642
0.051818182	2436.679741	2852.189651
0.052020202	2437.064096	2852.144082
0.052222222	2437.447458	2852.09992
0.052424242	2437.829835	2852.057149
0.052626263	2438.211236	2852.015755
0.052828283	2438.591667	2851.975722
0.053030303	2438.971138	2851.937035
0.053232323	2439.349655	2851.89968
0.053434343	2439.727228	2851.863643
0.053636364	2440.103863	2851.828909
0.053838384	2440.479569	2851.795464
0.054040404	2440.854352	2851.763296
0.054242424	2441.228221	2851.732389
0.054444444	2441.601182	2851.702732
0.054646465	2441.973244	2851.67431

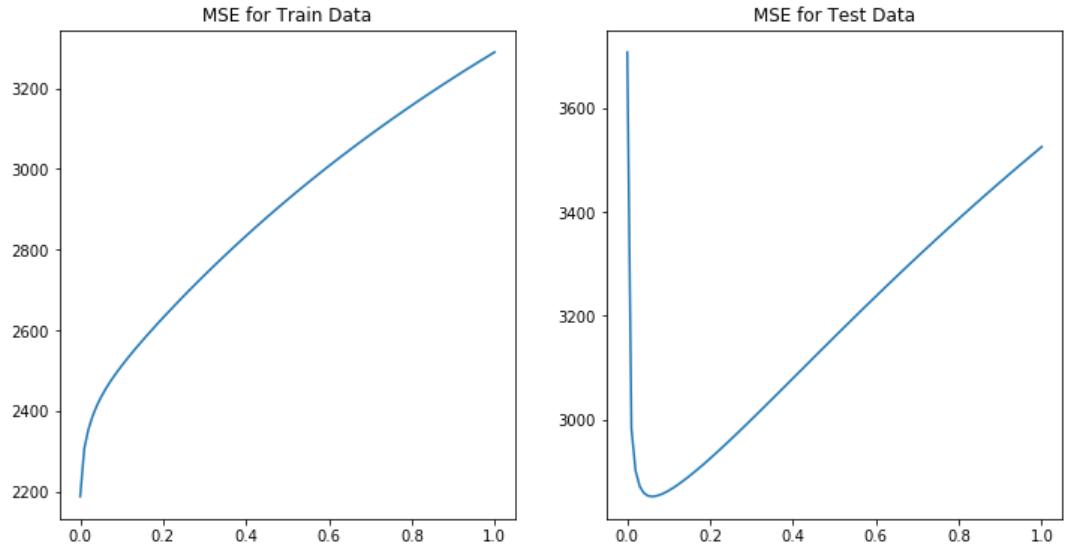
lambda	mses3_train	Mses3
0.055050505	2442.714696	2851.621122
0.055252525	2443.0841	2851.596331
0.055454545	2443.452634	2851.572724
0.055656566	2443.820303	2851.55029
0.055858586	2444.187115	2851.529017
0.056060606	2444.553075	2851.508892
0.056262626	2444.918192	2851.489905
0.056464646	2445.282472	2851.472042
0.056666667	2445.645921	2851.455294
0.054848485	2442.344412	2851.647111
0.056868687	2446.008545	2851.439648
0.057070707	2446.370352	2851.425093
0.057272727	2446.731348	2851.411619
0.057474747	2447.091539	2851.399214
0.057676768	2447.450931	2851.387869
0.057878788	2447.80953	2851.377572
0.058080808	2448.167344	2851.368313
0.058282828	2448.524377	2851.360081
0.058484848	2448.880635	2851.352868
0.058686869	2449.236126	2851.346661
0.058888889	2449.590854	2851.341453
0.059090909	2449.944826	2851.337233
0.059292929	2450.298047	2851.333991
0.059494949	2450.650523	2851.331718
0.05969697	2451.00226	2851.330404
0.05989899	2451.353263	2851.330041
0.06010101	2451.703537	2851.33062
0.06030303	2452.05309	2851.33213
0.07	2468.077553	2852.349994

MSE for Ridge regression

MSE Computed	Without Intercept	With Intercept
Training data	21703.66777322	2451.35326271
Test data	38929.59269319	2851.330041

From the values obtained we can infer that for Ridge regression gives lesser Mean squared errors using intercepts for both training and test data. Also, it gave lesser MSE values when compared to OLE regression and hence is a better proposition.

MSE vs lambda for training and test data

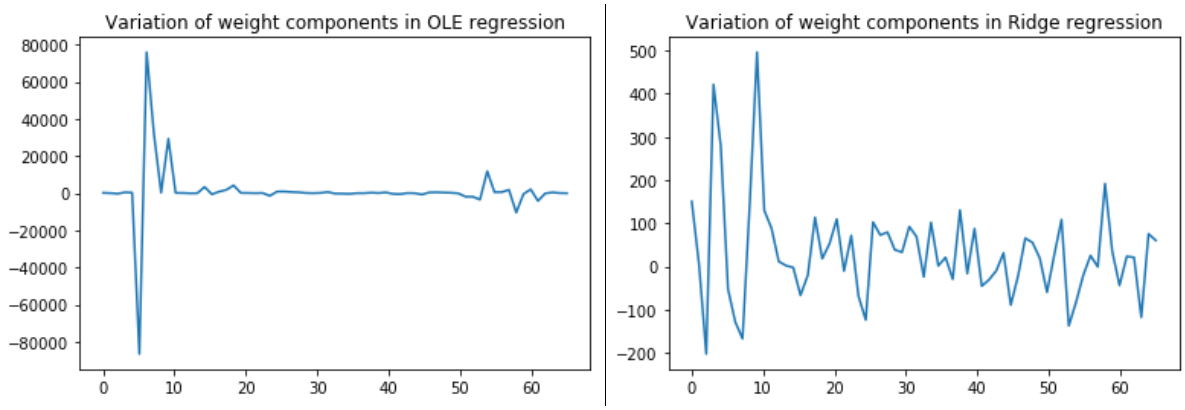


Relative magnitude of weights for OLE and Ridge regressions:

OLE Regression:
Magnitude of the weight vector computed without intercept: **1977655.3933971876**
Magnitude of the weight vector computed with intercept: **124531.52651877706**

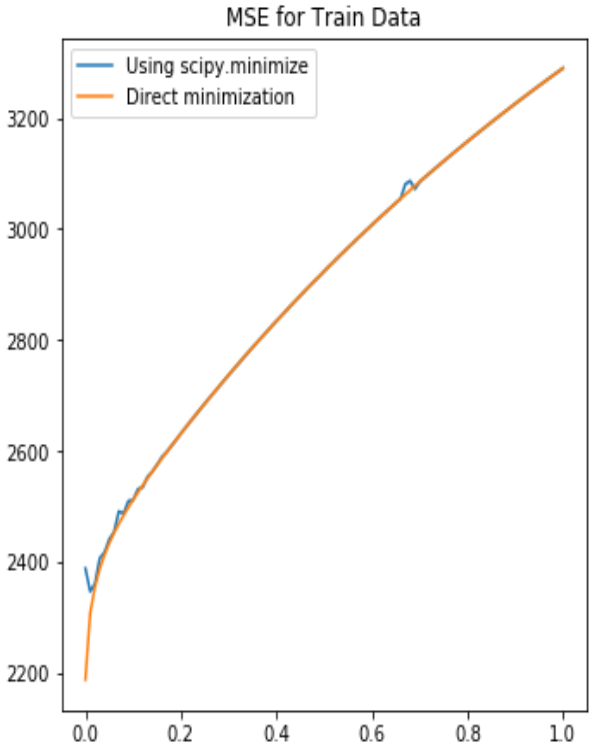
Ridge Regression:
Magnitude of the weight vector computed without intercept: **516.32934217488764**
Magnitude of the weight vector computed with intercept: **430.1286915482226**

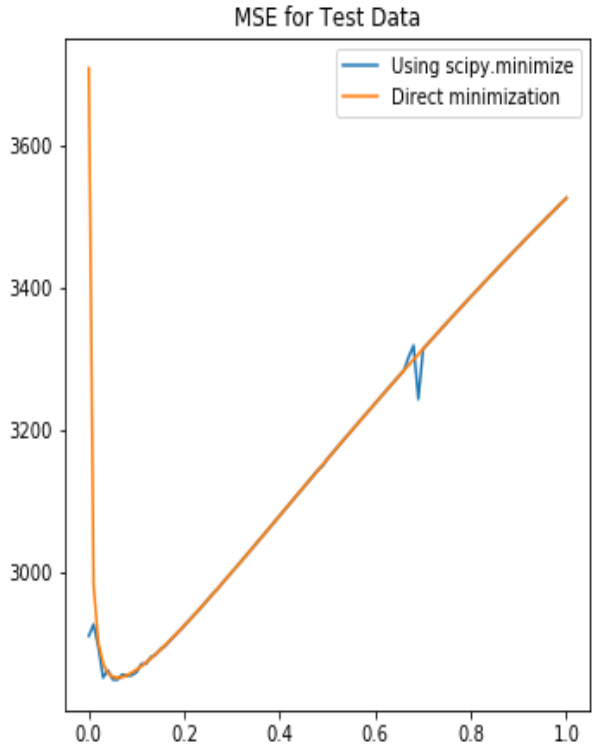
Going by the magnitudes of the weights attained from both the regressions above, we can clearly say that the weights are much higher after OLE regression as opposed to Ridge regression with a lambda value of 1. We can also see the variation of each weight component between OLE and ridge regressions from the graphs below.



PROBLEM 4

RIDGE REGRESSION LEARNING USING GRADIENT DESCENT





The results obtained in the above problem (problem 3) are similar to those obtained in the problem 4. It can be observed that the graph produced using Ridge regression is smoother than the one by gradient descent, though there are some outliers in the data. The curve could be made even smoother by increasing the number of iterations for training the data.

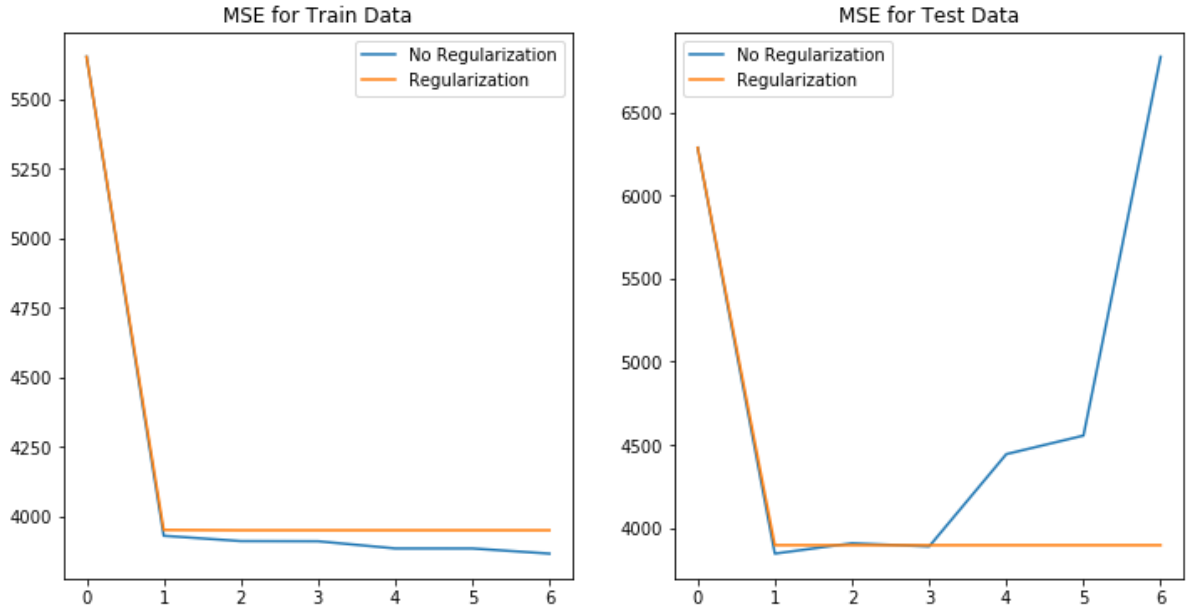
Conclusion:

In our scenarios here, a straight forward ridge regression is performing better when compared to the ridge regression using gradient descent as it takes less time to converge than by using gradient descent However while dealing with large matrices the matrix inversion in ridge regression would make the computations more expensive. Also encountering a singular matrix would jeopardize a direct ridge regression computation. In such cases gradient descent is preferred as each step is easy to compute.

PROBLEM 5

NON-LINEAR REGRESSION

The below figure shows MSE for train and test data with and without regularization. When MSE for train data is concerned, MSE vs Polynomial of degree 0 to 6 is shown in the figure. The prediction error with regularization is shown with optimal value of lambda= 0.05989899, determined above in problem 3.



	Variation of MLE values with order of the polynomial p			
p	**Training data**	**Test data**	**Training data with optimal lambda**	**Test data with optimal lambda**
0	5650.710539	6286.404792	5650.711902	6286.881162
1	3930.915407	3845.03473	3951.776476	3895.739692
2	3911.839671	3907.128099	3950.623263	3895.468341
3	3911.188665	3887.975538	3950.618484	3895.467023
4	3885.473068	4443.327892	3950.618289	3895.466975
5	3885.407157	4554.830377	3950.618287	3895.466976
6	3866.883449	6833.459149	3950.618287	3895.466976

- The optimal value of *p* in terms of test error without regularization is p=1 and the value of MSE is **3845.03473**
- The optimal value of p in terms of test error with regularization is p>=1 (stays same for p in[2,6]) and the value of MSE is **3895.466975**

Conclusion:

From the train data graph, we can see that as the degree of polynomial increases, prediction error decreases in both the cases. This can be assumed as the order of polynomial increases, the curves fit better with testing data which makes the MSE to reduce.

From the test data graph, it can be seen that as the order increases till the order of 3, the value of MSE reduces till 1 and almost stays constant till 3 with and without regularization. But as the degree increases over 3, there was no change in MSE with regularization, but in no-regularization case, MSE increases as there is increase in the order of the polynomial. This could be attributed to the over fitting which happened in the training part, leading to a spike in MSE for the testing data.

PROBLEM 6

INTERPRETING RESULTS

OLE Regression:
MSE for Training data without intercept: **19099.44684**
MSE for Test data without intercept: **106775.3616**
MSE for Training data with intercept: **2187.160295**
MSE for Test data with intercept: **3707.840181**

Ridge Regression:
Optimal Lambda: **0.05989899**
MSE for Training data without intercept: **21703.66777322**
MSE for Test data without intercept: **38929.59269319**
MSE for Training data with intercept: **2451.35326271**
MSE for Test data with intercept: **2851.330041**

Non Linear Regression:
MSE for Training data without regularization: **3866.883449**
MSE for Training data with regularization: **3950.618287**
MSE for Test data without regularization: **3845.03473**
MSE for Test data with regularization: **3866.883449**

Metrics for best setting:

- **Mean Squared Error (MSE)** can be used as a metric to decide the best classification technique as it shows how accurate the algorithm works for a particular classification
- In some exceptional cases like in non-linear regression, increasing the **order of polynomial** may give good training error, but this cannot be taken as a metric as it gives very bad error rate in test due to **over fitting** in the training phase.
- The **complexity and run time** of the algorithm can also be taken into account as a metric for classification. As in the case of Ridge regression, the matrix size causes Ridge regression to fall apart when compared to linear regression. Also encountering a **singular matrix** in ridge regression would hinder the progression of ridge regression. Hence the size and complexity of the data set is also considered in determining the algorithm. With small data set there is no much change in the performance but with larger matrices gradient descent gives better performance with loss in accuracy.

Going by the results above, using our metrics for classification, we can give the following recommendations –

- Using intercepts gives better MSE values for both test and training data in case of OLE as well as Ridge regression
- Ridge regression gives better performance as well as lower MSE values over OLE with an optimal regularization parameter
- For simple data sets, Ridge regression outperforms Regression using gradient descent whereas for complex data sets Regression with gradient descent is preferable over Ridge regression

- Increasing the order of the polynomial for non-linear regression gives better MSE values for training data. However the MSE value of the test data gets hugely impacted when the order of the polynomial goes above 3 without regularization.

Summary:

It can be seen that non gradient descent and gradient descent ridge regression gives same results in terms of accuracy is concerned. But in terms of performance, ridge regression is not chosen with larger data sets as complexity is concerned as it runs faster for smaller datasets and also due to the chance of encountering a singular matrix.