CSE474/574 - Introduction to Machine Learning Programming Assignment 2 Classification and Regression

TEAM MEMBERS:

SMUNAGAL@BUFFALO.EDU UB ID# 50168800

SHRAVYAT@BUFFALO.EDU UB ID# 50169587

CSE 574 GROUP # 9

INDEX.

Problem 1 Report

Problem Statement: Train both methods using the sample training data (sample train). Problem Report the accuracy of LDA and QDA on the provided test data set (sample test). Also, plot the discriminating boundary for linear and quadratic discriminators. The code to plot the boundaries is already provided in the base code. Explain why there is a difference in the two boundaries.

Observations: In Problem 1, we have implemented the Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). In LDA, we have first learnt the parameters Sigma (Covariance matrix) on the entire data set and Mu (vector) separately for each class data, in the function "*ldaLearn*". Then we implemented the "*qdaLearn*" function in which we again learnt the Sigma (Covariance matrix) and Mu (vector), both separately for each class data.

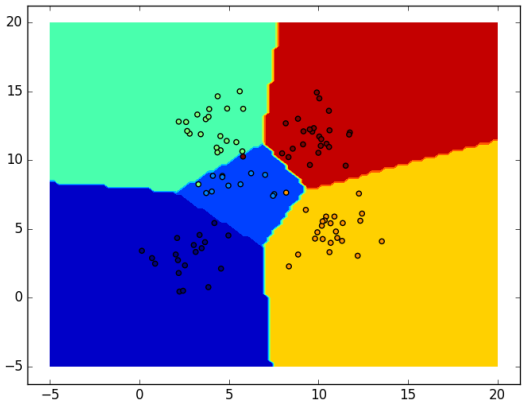
Once the parameters were learnt we have calculated the probabilities with respect to each class for a single example and then labeled it with a class which gave us the highest probability, in the function "*ldaTest*" using the parameters obtained in the "*ldaLearn*". So we have predicted the class labels in the same fashion for the entire test data set.

Similarly we have predicted the class labels for the entire test data set using the learnt parameters obtained from the "*qdaLearn*" function.

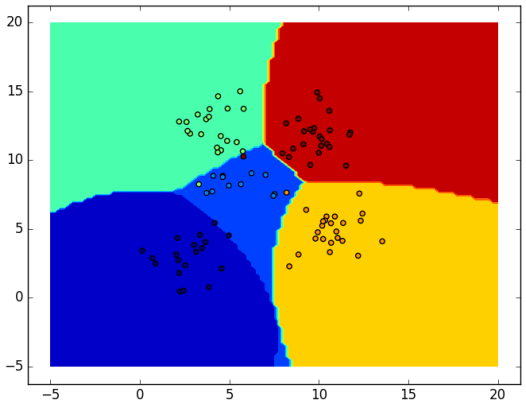
$$\begin{aligned} p(y|\mathbf{x}) &= p(y)\prod_j p(x_j|y) = p(y)\prod_j \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_j-\mu_i)^2}{2\sigma_j^2}} \\ &= p(y)\frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}} \end{aligned}$$

Since we had the true labels for the test data, we went ahead and found out the accuracy of the prediction done by LDA which came up to 95% and for the QDA which came up to 97%.

The Discriminating Boundary for the LDA and QDA are plotted below



LDA boundary plot with predictions of test data



QDA boundary plot with predictions of test data

Conclusion: From the obtained accuracies we can conclude that our prediction was better in case of QDA when compared to that of LDA. In LDA we use same the Covariance Matrix for each class. This could lead to a potential problem as we assumed the distribution radius across the mean for each output label (class) is same and hence can result in some errors.

The above problem does not happen with QDA because we have separate Covariance matrices for each class and the distribution around mean for each class is defined by its own covariance matrix. This difference led to QDA's prediction to be better for the test data in terms of accuracy.

Problem 2 Report

Problem Statement: Calculate and Problem Report the RMSE for training and test data for two cases: first, without using an intercept (or bias) term, and second with using an intercept. Which one is better?

Observations: In Problem 2, we have implemented the "Linear Regression" which is one of the methods used for Probabilistic Analysis of data where the random variable we use is a continuous variable. In this we have first implemented the "learnOLERegression" function to learn the weights vector i.e., the weight coefficients using the training data we had using the below formula

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

Then we do the prediction using the learnt weight vector in the function "testOLERegression" on the test data we have using the below formula.

$$y = \mathbf{w}^\top \mathbf{x}$$

Once we have the predicted data, using the true labels data, we compute the Root mean square error to check how well the function could predict.

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

Root mean square error was calculated for both the training set and validation set and used to evaluate the performance of the learn function which we have implemented. The regularization term was not included in the calculation of the root mean square error for linear regression but we will tell more about in the problem 3.

For both data sets we have recorded the RMSE error value with Intercept and without intercept which are recorded below.

For Training data:

RMSE without intercept = 138.20074835
RMSE with intercept = 46.7670855937

For Test data:

RMSE without intercept = 326.76499438
RMSE with intercept = 60.892037097

Conclusion: We can observe from the above finding's that RMSE error with intercept is lower, hence we can say that the weights are learnt better in the case with the intercept and we can conclude that it is better for both the cases. In General, the regression line is forced to go through the origin. If the fitted line doesn't naturally go through the origin, the regression coefficients i.e.,

the weights learnt may be biased when we don't include the intercept. The intercept and the learnt weights' co-efficient are such that they minimize the root mean square error.

Problem 3 Report

Problem Statement: Calculate and Problem Report the RMSE for training and test data using ridge regression parameters using the "testOLERegression" function that you implemented in Problem 2. Use data with intercept. Plot the errors on train and test data for different values of λ. Vary λ from 0 (no regularization) to 1 in steps of 0.01. Compare the relative magnitudes of weights learnt using OLE (Problem 1) and weights learnt using ridge regression. Compare the two approaches in terms of errors on train and test data. What is the optimal value for λ and why?

Observations: In this problem, we use different type of regression which is called the "Ridge Regression" to over some problems faced using the Linear regression. In this, we use a regularization term in the error function, which solves our problems and the regularization factor is "Lambda (λ)". So in Ridge regression, the weight vector is learnt using the below formula.

$$\widehat{\mathbf{w}}_{MAP} = (\lambda \mathbf{I}_D + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

And the error function with the regularization factor is as below:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{1}{2} \lambda ||\mathbf{w}||_2^2$$

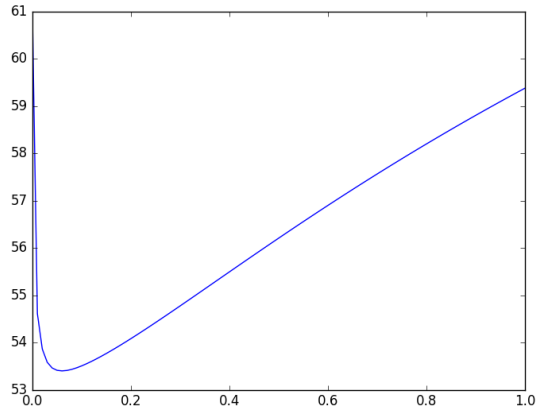
Once the weight vector has been learnt we use the same "testOLERegression" for the prediction. To check the difference between the weights learnt by Linear Regression and Ridge Regression, we have calculated the average of the weight coefficients learnt and we have observed that the weights learnt by the Ridge regression are smaller in magnitude when compared to that of the weighs learnt by linear regression for the same data.

We went ahead and checked the RMSE values obtained by Linear Regression and Ridge Regression and we have observed that the RMSE values are low in case of the Ridge regression and hence we can conclude that the Ridge Regression approach is better in terms of the error. This analysis above is when we consider the Test data, but in case of Training data it's the reverse.

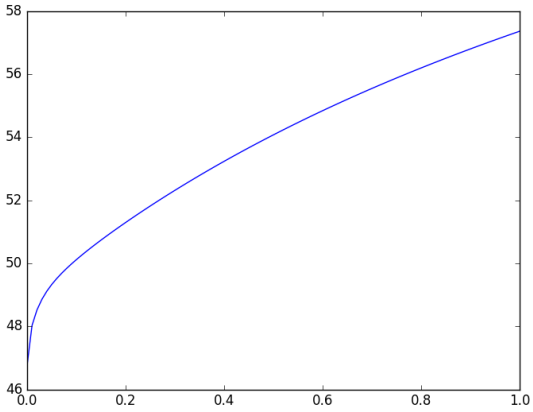
Now, we have used different values of lambda λ i.e., the regularization factor while learning the weights and have calculated the RMSE values for the different weight vectors learnt.
By the values of RMSE, we could say that we had a minimum RMSE values for λ values as below:
Training data: λ = 0.00
Test Data: λ = 0.06

We have plotted graphs for different values of λ against the obtained RMSE values to see how the λ value effects the learning of weight vectors in Ridge Regression and thus affects the RMSE value, since our ultimate aim is to learn the weight vector which can reduce the error vale.

The plots are as below:



Problem 3 Test Data error variation with lambda



Problem 3 Training Data error variation with lambda

Problem 4 Report

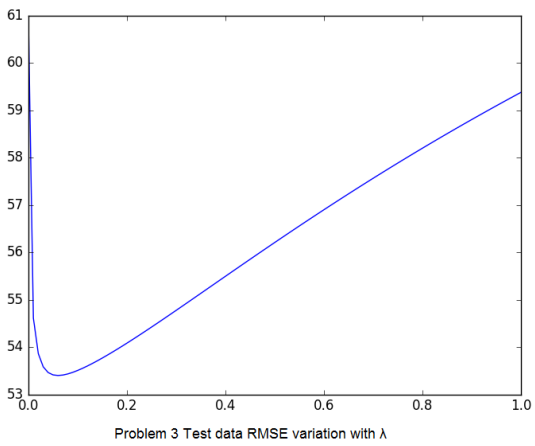
Problem Statement: Plot the errors on train and test data obtained by using the gradient descent based learning by varying the regularization parameter λ, Compare with the results obtained in Problem 3.

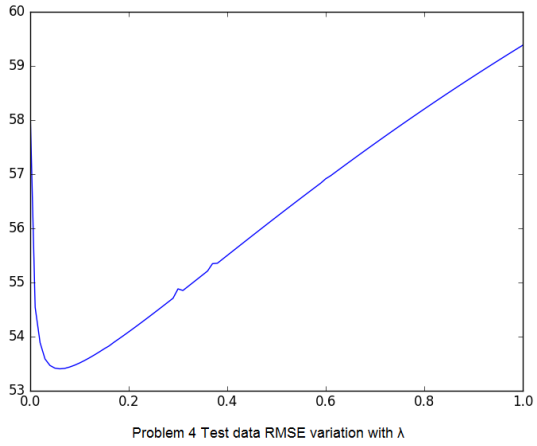
Observations: In Problem 4, we are calculating RMSE using Gradient Descent based learning for different λ values.
In general the Ridge regression while learning the weight vector includes a computation of inverse of a matrix, which isn't feasible always. To overcome this we use the Gradient Descent function over the error function J(w) so that we can learn a weight vector which minimizes this error over the training data set.

For the Gradient descent functionality we use the inbuilt *numpy* function "minimize", to which we pass two values one being the error function J(w) and the other being the derivative of the error function with respective to w, the weight vector.

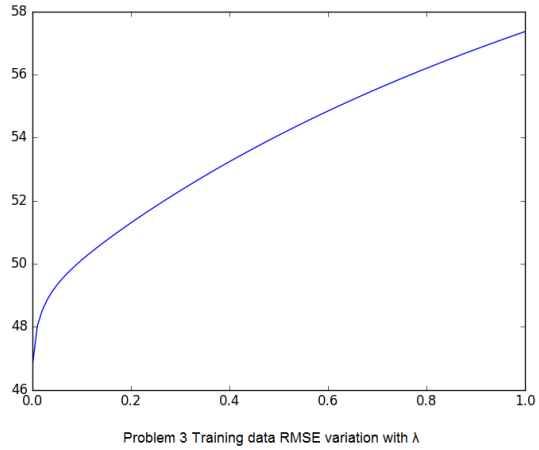
The below graph plotted shows the variation of RMSE Error Computed for different Lambda values

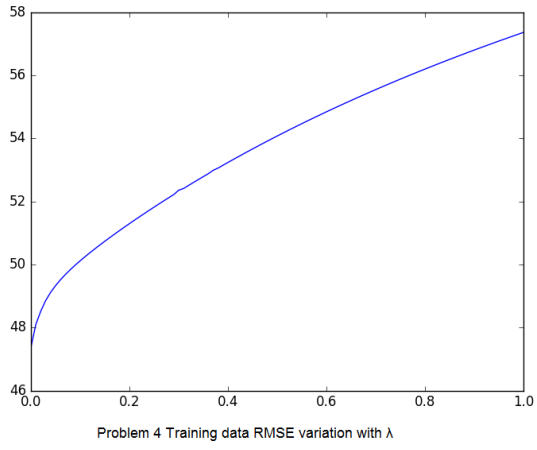
*Problem **3** Test Data RMSE Variation with λ compared to Problem **4** RMSE variation:*





*Problem **3** RMSE Training Data Variation with λ compared to Problem **4** RMSE variation:*





Conclusion: From the figure plotted above for training and test data we can see that, the optimal lambda value calculated from either Ridge Regression method or Gradient Descent algorithm is approximately same.

Problem 5 Report

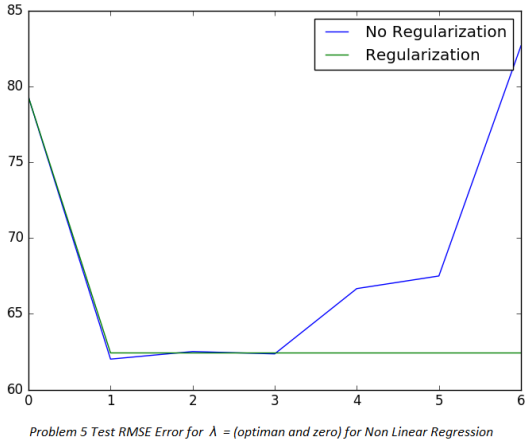
Problem Statement: Using the λ = 0 and the optimal value of λ found in Problem 3, train ridge regression weights using the non-linear mapping of the data. Vary 'p' from 0 to 6. Note that p = 0 means using a horizontal line as the regression line, p = 1 is the same as linear ridge regression. Compute the errors on train and test data. Compare the results for both values of λ. What is the optimal value of d in terms of test error in each setting? Plot the curve for the optimal value of p for both values of λ and compare.

Observations: In Problem 5 we are expanding a single input attribute of the given data to P-Dimensional vector where *range (p) = [0, 7)* and then comparing the RMSE value with Regularization (λ = optimal) vs No- Regularization (λ = 0) over the range of p.

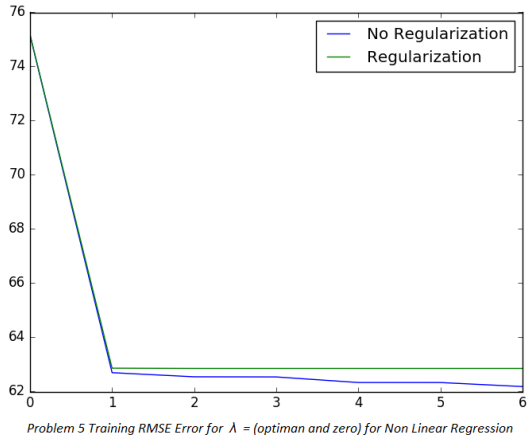
If λ = 0, it implies that regularization term is zero and hence no regularization.
In problem 3, we have calculated the optimal lambda values for both training and test data which are
Training data: λ = 0.00
Test Data: λ = 0.06

The function used for expanding the input data is:
F(x, p) = [x^0, x^1, x^2 ……, x^p] P-Dimensional vector is returned for one input attribute

The comparison of RMSE value with λ = optimal vs λ = 0 for p in range 0 to 7 is shown below:



Problem 5 Test RMSE Error for λ = (optiman and zero) for Non Linear Regression



Problem 5 Training RMSE Error for λ = (optiman and zero) for Non Linear Regression

Conclusion: Observing the results for Test data, we can see that the RMSE error with regularization is lower as the value of 'p' increases and then it becomes stable. And for training data we do not see any improvement with Regularization.

Problem 6 Report

Problem Statement: Compare the various approaches in terms of training and testing error. What metric should be used to choose the best setting?

The first approach in problem 1, which we have used is the Discriminant Analysis – Linear and Quadratic. By looking at the accuracies we can conclude that Quadratic Discriminant Analysis is a better approach when we are doing a Discriminant Analysis of data.

In the rest of the problems we have use two types of Regression:

1. Linear Regression
2. Ridge Regression
3. Regression with Basis Function expansion

By comparing the error values obtained on the training and test data provided to us, we came to a conclusion that Linear regression provided better results on Training data. Whereas Ridge regression provided better results on the Test data.

The ultimate goal is learn the weight vector which reduces the error value which we have been calculating in the above functions.

In real time, there are lot of problems we face when we use the linear regression in terms of

1. Impact of Outliers.
2. Constraint of Linear equation between X and Y.
3. Under fitting problem.
4. Unstable when the data has Correlated Attributes

So we may tend to use the Ridge regression to overcome the under fitting problem and also selecting the λ value accordingly as it also impacts output precision.