Programming Assignment 2

Classification and Regression

CSE 574 | Group 24

Mohamed Ismail
Nikhil Pillai
Krithika Krishnan

April 12, 2017

REPORT 1 : Experiment with Gaussian Discriminators

1. **Implementation of Linear Discriminant Analysis (LDA)**

   LDA approaches the problem by assuming that the conditional probability density function, p(x|y=1) and p(x|y=2) is normally distributed with mean and covariance parameters (μ_1, Σ_1). Under this assumption, the Bayes optimal solution is to predict points as being from the second class if the log of the likelihood ratios is below some threshold T, so that:

   $(x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + \ln|\Sigma_1| - (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) - \ln|\Sigma_2| < T$

   Without any further assumptions, the resulting classifier is referred to as QDA (quadratic discriminant analysis). Whereas in LDA,the assumption is made that all variances are equal: $\Sigma_1 = \Sigma_2 = \Sigma$ (homoscedasticity assumption)

   These classifiers can be extended to include more than 2 classes, which was the case in our data. The above concepts are analogous in the case of 5 classes, except with 5 means, 5 covariance matrices, etc. All classes had the same prior likelihood.

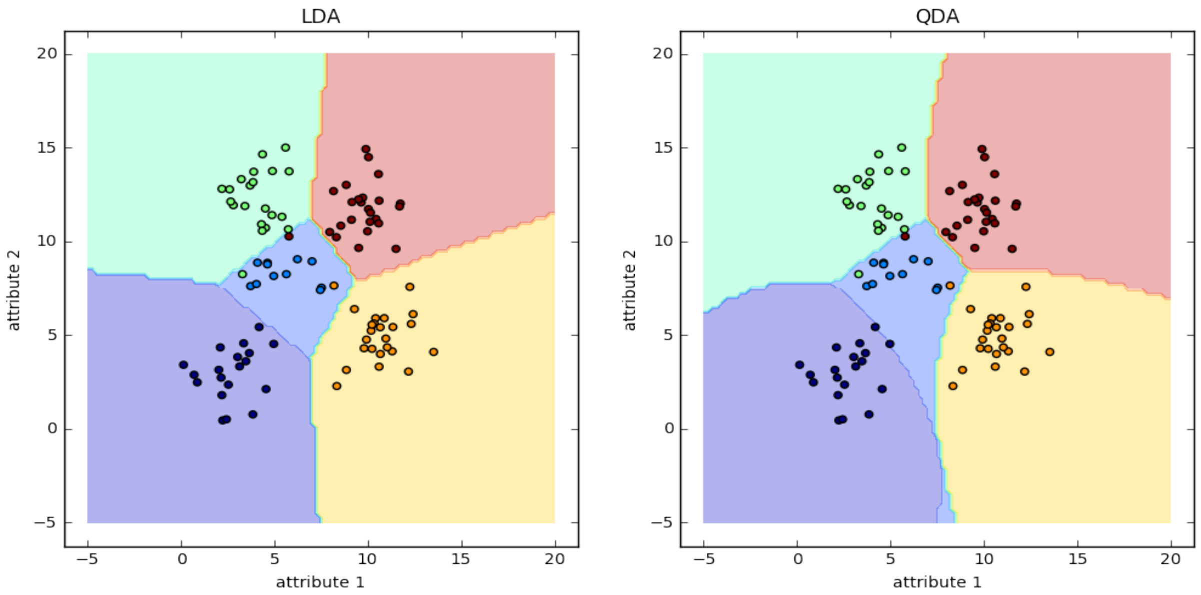
2. **Implementation of Quadratic Discriminant Analysis (QDA)** In QDA, $\Sigma_1 \neq \Sigma_2 ... \neq \Sigma_5$

3. **Accuracy of LDA and QDA**

   The accuracy of LDA and QDA on the provided test data set (sample test) for the respective methods,

   - Linear Discriminant Analysis (LDA) Accuracy = **97%**
   - Quadratic Discriminant Analysis (QDA) Accuracy = **96%**

4. **Plot the discriminating boundary for linear and quadratic discriminators**



5. **Difference in the two boundaries**

   The discriminating boundary in the two plots differs in the fact that the boundary for LDA is linear and in QDA they are curved.

   Conclusion : The linear discriminant function is a linear function of x. This is because the covariance matrix is identical when we vary values of k. However, the quadratic discriminant function is a non-linear function of x as the assumption for the covariance matrix here is not said to be identical for different values of k.

REPORT 2 : Experiment with Linear Regression

1. **Test Data**
   - MSE for **test data** having intercept : 3707.84018154
   - MSE for **test data** without intercept : 106775.361554
2. **Training Data**
   - MSE for **training data** having intercept : 2187.16029493
   - MSE for **training data** without intercept : 19099.4468446

   Mean Squared Error(MSE) using intercept is lesser than without intercept in both the training and test data. Therefore, the MSE for training and test data for the second case, with using an intercept is better

REPORT 3 : Experiment with Ridge Regression

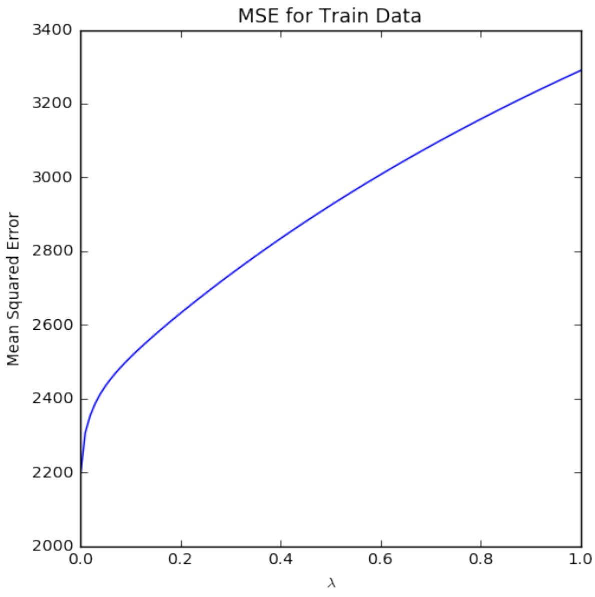
- Used the following equation to implement ridge regression by minimizing the regularized squared loss

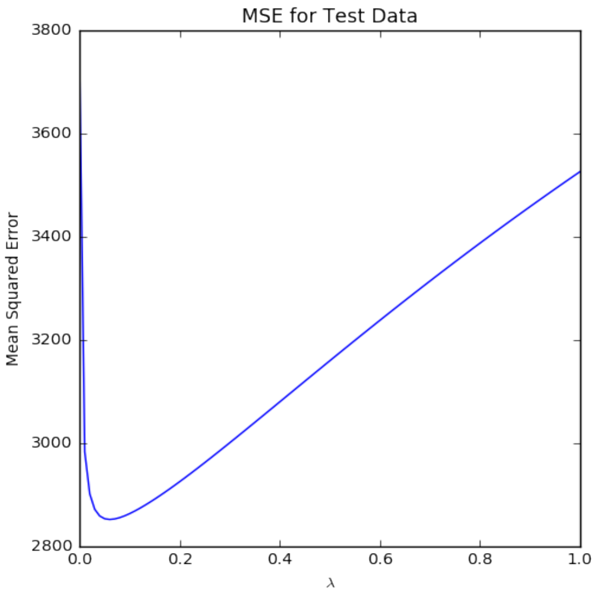
$$J(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - w^T x_i)^2 + \frac{1}{2}\lambda w^\intercal w \tag{1}$$

- The squared loss in matrix-vector notation

$$J(w) = \frac{1}{2}(y - Xw)^T(y - Xw) + \frac{1}{2}\lambda w^\intercal w \tag{2}$$

1. The MSE for training and test data using ridge regression parameters using the the testOLERegression function that was implemented in Problem 2,(using data with intercept)
2. The errors on train and test data for different values of λ was plotted. λ was varied from 0 (no regularization) to 1 in steps of 0.01.





Compared the relative magnitudes of weights learnt using OLE (Problem 2) and ridge regression

By computing l2norm of the individual weights for OLE(with and without intercept) and Ridge Regression, we found that the weights of Ridge regression indicated a lower value.

OLE without intercept = 1977655.3932250608

OLE with intercept = 124531.526521

ridge regression = 430.12869154822266

It can be observed that ridge regression had a significantly lower value. This makes sense since adding the λ value adds penalty to the weights and doesn't allow them to become too large, preventing overfitting of the train data. The magnitudes of the weights are smaller in ridge regression as one might guess, since ridge regression penalizes large weights.

Compared the two approaches in terms of errors on train and test data

Linear Regression :

MSE for test data having intercept : 3707.84018154

MSE for test data without intercept : 106775.361554

MSE for training data having intercept : 2187.16029493

MSE for training data without intercept : 19099.4468446

Ridge Regression :

MSE for test data at optimal λ: 2851.330

MSE for train data at optimal λ: 2451.528

From the MSE values it can be observed that Ridge regression is more robust since its average MSE is lower than that for the OLE regression. This can be due to overfitting issues.

The optimal value for λ is **0.06**. This is the point at which Mean Squared error (MSE) for the test data is minimal.

From the plot we can observe that in terms of the test data, optimum value of λ is 0.06. This is the point at which the best performance of the model on the test data is obtained.

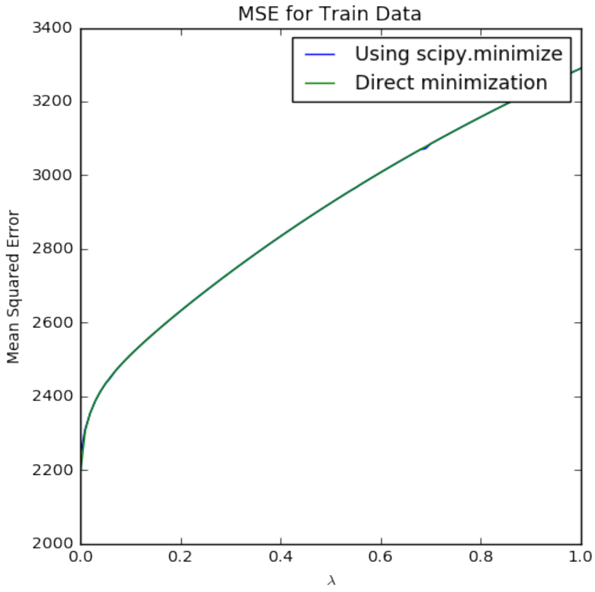
REPORT 4 : Using Gradient Descent for Ridge Regression Learning

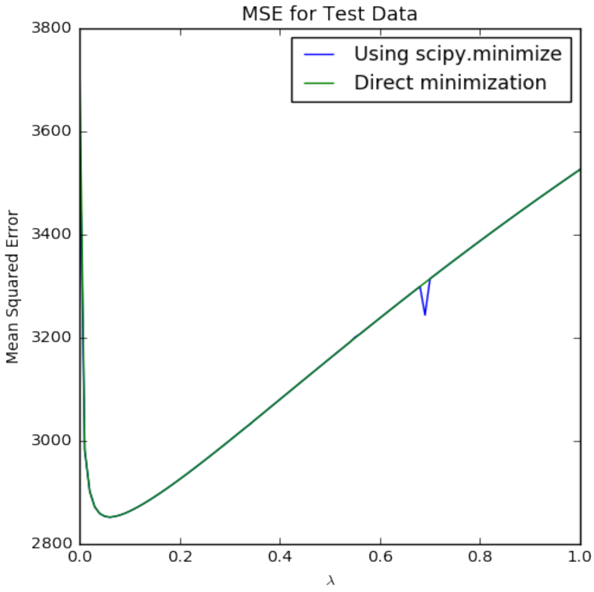
To avoid computation of $(X^\intercal X)^{-1}$, gradient descent was used to minimize the loss function (or to maximize the log-likelihood) function. In this problem, gradient descent procedure was implemented for estimating the weights, w.

The square loss function, J(w) was obtained from problem 3. The derivative of this equation (equation 1) with respect to w, was found to be the gradient of the regularized squared error.

$$\frac{dJ(w)}{dw} = \frac{-y^T x + w^T (x^T x)}{N} + \lambda w^\intercal \tag{3}$$

The errors on train and test data obtained by using the gradient descent based learning by varying the regularization parameter, λ was plotted.





Comparison with the results obtained in Problem 3 :

As can be observed from the plot in problem 3, there isn't a big variation in the minimum value of Mean Squared Error(MSE) for the test data in both the methods conducted in problem 3 and 4 when iterations are sufficiently high.

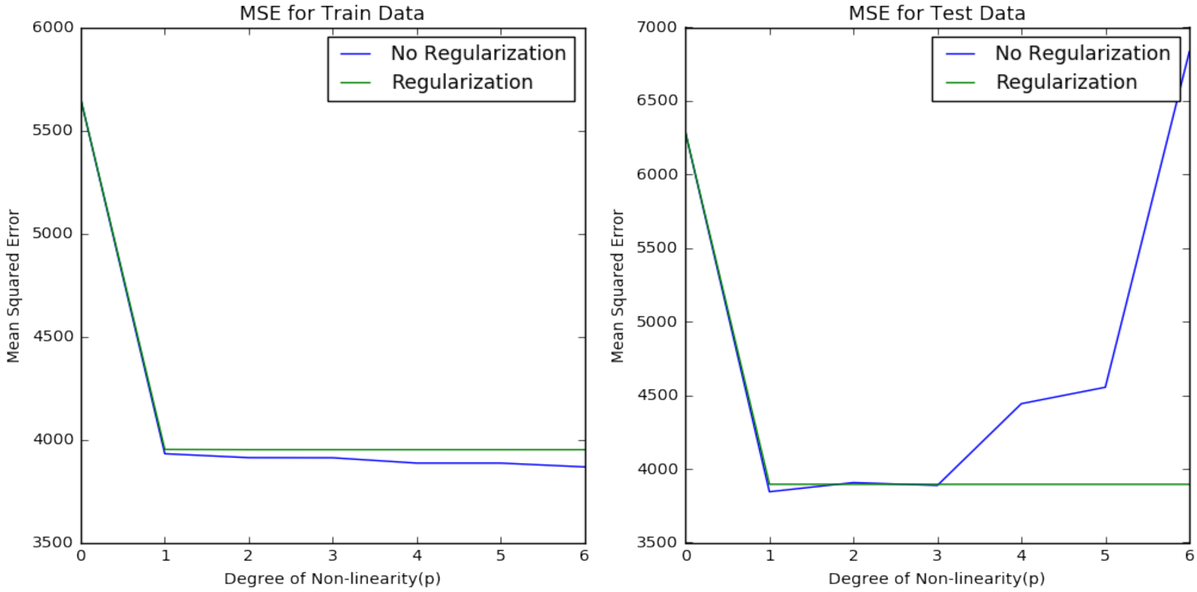
Computing using gradient descent is computationally cheaper than computing ridge regression. Since in ridge regression we have to calculate the inverse matrix, for larger X it becomes too difficult to do so. In order to be able to take inverse of X, X should be invertible. i.e. there should be no dependency between two rows or columns of X. Multiplication between 2 matrices for weight calculation as in ridge regression scales the computational cost as $o(n^3)$. Therefore, gradient descent with regularisation is computationally cheaper than normal regression and ridge.

REPORT 5 : Non-linear Regression

In this problem, we investigated the impact of using higher order polynomials for the input features. For this problem we use the third variable as the only input variable:

Implemented the function mapNonLinear which converts a single attribute x into a vector of p attributes, 1,x,x^2,.....,x^p.

The graph below shows the errors plotted for test and train data for the corresponding values of p.



From the graph, for the case of no regularization it can be seen that the overall mean squared error for training data reduces with increase in p for the train data and for the test data it decreases then remains constant and finally, later with increase in p it increases. This is likely due to overfitting issues. This overfitting issue is resolved by using regularization as observed in the test data plot.

Training data

When there is zero regularization, i.e. λ = 0, the optimal value of p = 6.

When the regularization, λ is set to an optimum value of 0.06, model performs better, having optimal p value = 4.

Test data

When there is no regularization done, λ = 0, the optimal value of p = 1. The model is allowed to over fit, and we can observe this by a sharp decrease in performance with p > 3.

When the λ = 0.06, over fitting is not so significant and the optimal value of p is observed to be 3.

Plot for the curve to find the optimal value of p so as to compare with both values of λ .

REPORT 6 : Interpreting Results

From the results obtained in the previous 4 problems, final recommendations for anyone using regression for predicting diabetes level using the input features. We trained the model on the training dataset in order to obtain the individual efficiencies of the methods when applied on a test dataset like the diabetes prediction.

Linear Regression :

MSE for test data having intercept : 3707.84018154

MSE for test data without intercept : 106775.361554

MSE for training data having intercept : 2187.16029493

MSE for training data without intercept : 19099.4468446

Ridge Regression :

MSE for test data at optimal λ: 2851.330

MSE for train data at optimal λ: 2451.528

Using Gradient Descent for Ridge Regression :

MSE for test data at optimal λ: 2850.854

MSE for train data at optimal λ: 2449.293

Non Linear Regression :

MSE for test data at optimal λ and optimal P: 3895.583

MSE for train data at optimal λ and optimal P: 3950.682

For the diabetes data we would suggest using gradient descent with ridge regression. This method provides the least MSE compared to other methods and also compared to normal ridge regression this method is computationally cheaper since it does not involve matrix inverse and matrix multiplication (as discussed in report 4 section). The metric to be used to choose the best setting is the MSE of the model on the test data.