Assignment 2: Classification and Regression

CSE 574 Introduction to Machine Learning

Isabela Lago, Nathan Margaglio, Timothy Schuler

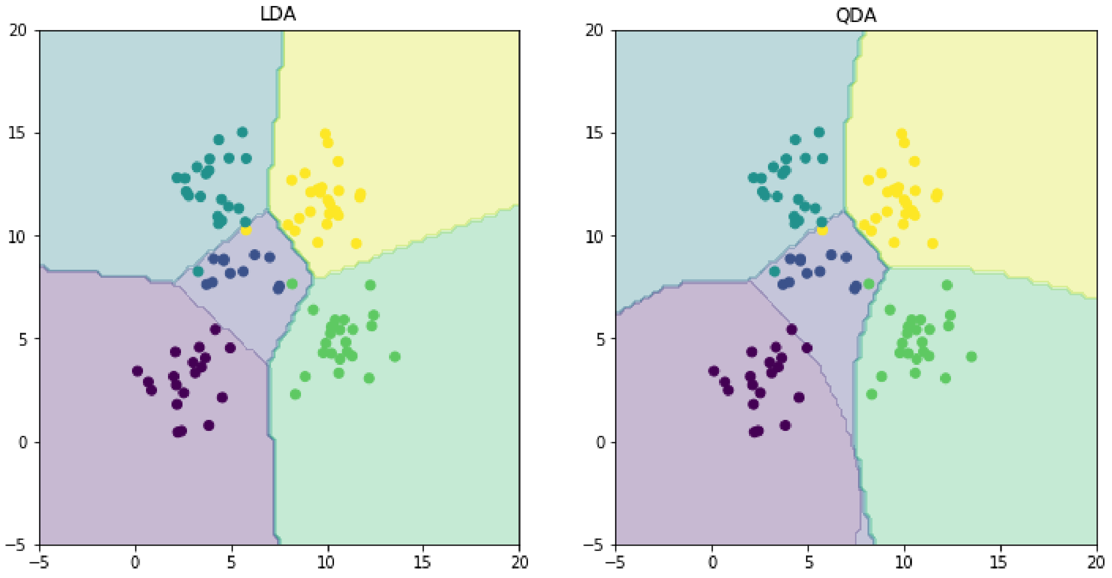
July 24th, 2018

1 Problem 1

1.1 Experiment with Gaussian Descriminators

For this problem, we trained our implementation of **Linear Discriminant Analysis** (LDA) and **Quadratic Discriminant Analysis** (QDA) on the sample training data (**sample_train**).

```
LDA Accuracy = 0.97
QDA Accuracy = 0.96
```



We see that the accuracy of LDA is 97% (97/100 correctly classified) and QDA is 96% (96/100 correctly classified). The plots show the example points (with each point being a different color based on classification and each boundary's area being a different color based on the same scheme).

Notice the difference in the shape of the boundaries. For LDA, we see the shape is more linear, while for QDA the shape is more quadratic. This is because our use of each class' covariance matrix allows for more degrees of freedom through more trained variables, making the QDA's boundaries fit more closely with the data (despite this actually leading to less accuracy in this example).

2 Problem 2

2.1 Experiment with Linear Regression

For problem 2, we implement *ordinary least squares method* to estimate regression parameters by minimizing the squared loss. We note the MSE is **106775.36155730896** when not using an intercept and **3707.8401815996954** when using an intercept (a gain in accuracy of about **103067**).

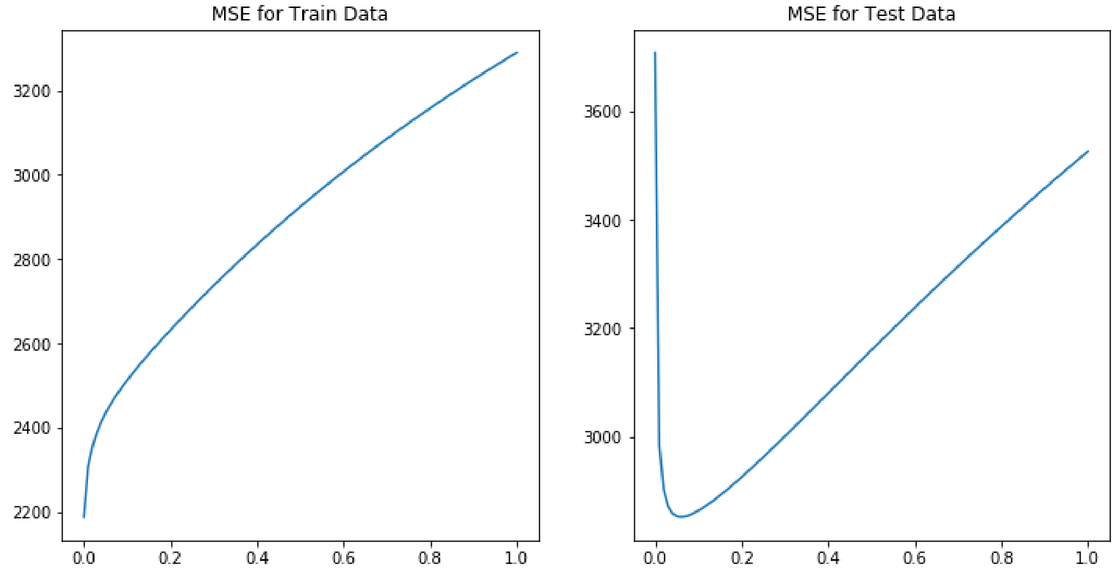
```
MSE without intercept 106775.36155730896
MSE with intercept 3707.8401815996954
Gain in accuracy: 103067.52137570927
```

This increase in accuracy can be attributed to the fact that adding the intercept allows for the regression to fit the data more closely by adding another dimension of freedom for the resulting hyperplane to be translated over.

3 Problem 3

3.1 Experiment with Ridge Regression

For this problem, we implement *Ridge Regression*, similar to linear regression like before, but with *L2 regularization*. This regularization penalizes complexity by adding a term to the error equivalent to the squared sum of the weights multiplied by some constant λ.



```
MSE with intercept 2851.3302134438477
Magnitudes of Ridge Weights: 5240.0949268758395
Magnitudes without Ridge: 286570.97996703925
```

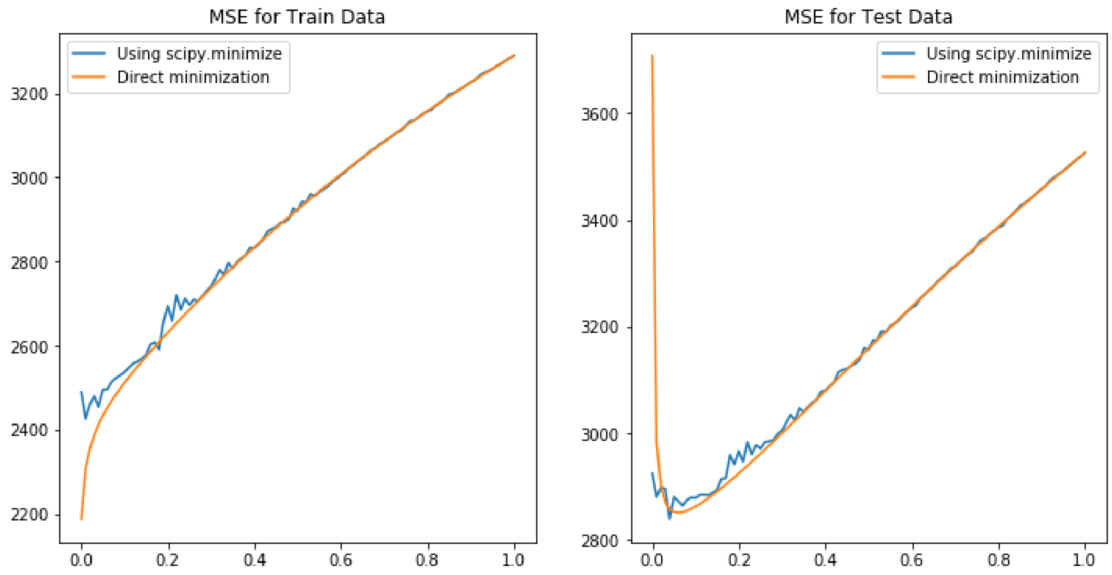
In our implementation, we vary λ through 100 equally spaced values between 0 and 1 and use this value to train and test the regression. In doing so, we found that the λ that produces the highest accuracy in the test data is *0.06*. This value provides optimal accuracy in that it limits the complexity of the regression (i.e., keeping the weights lower than without it), so as to prevent overfitting.

When comparing the magnitudes of the weights for regression using and not using regularization, we see a drastic change. When not using regularization (i.e., the previous problem), we have a magnitude of about **286,570**. When using regularization, the magnitude it **5,240**.

4 Problem 4

4.1 Using Gradient Descent for Ridge Regression Learning

For this problem, we re-implement Ridge Regression, but this time by utilizing *Gradient Descent*. The reason we'd use this method over directly computing the regression is that the closed-form solution of ordinary least squares regression involves taking an inverse of a matrix, which may not be suitable in a numerical setting. Gradient Descent helps us avoid this issue (at the cost of training time and volatility in training).



We see in the plots that Gradient Descent is more erratic in terms of error for every λ we use in learning. This is due to the nature of Gradient Descent and the uncertainty (and sometime inability) of the algorithm to properly find a global minimia.

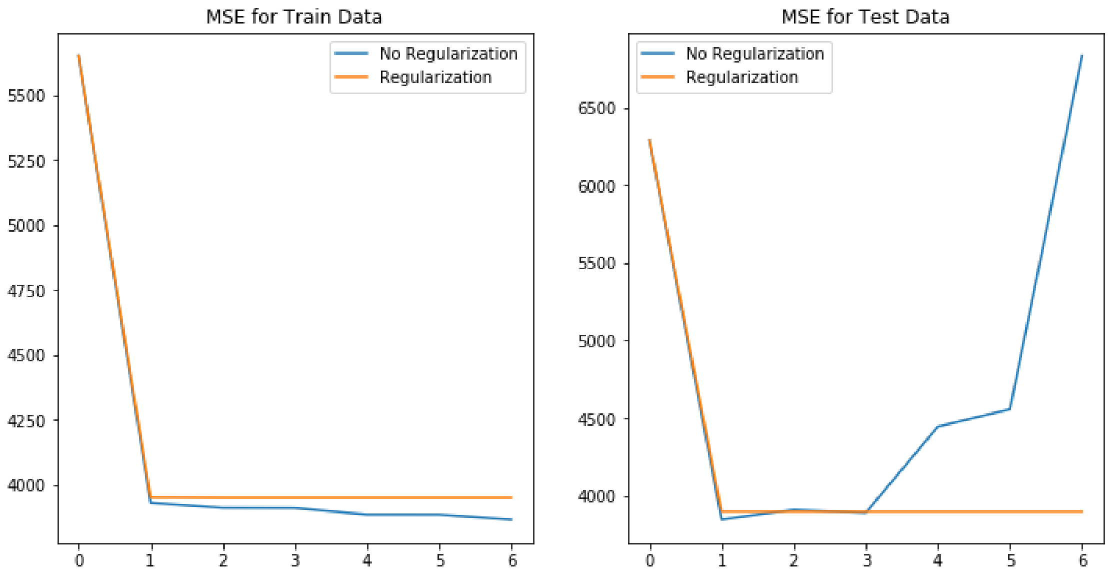
Despite this, Gradient Descent performs well relative to the closed form solution, with higher values of λ producing more closely aligned errors.

5 Problem 5

5.1 Non-linear Regression

For problem 5, we investigate the impact of using higher order polynomials for the input features. This converts a single attribute **x** into a vector of p attributes, *1*, x, x^2, ..., x^p. For this problem, we only use the third variable as the input variable.

We plot the MSE for both no regularization ($\lambda = 0$) and regularization using our optimal value from problem 3 ($\lambda = 0.06$) as we vary p from 0 to 6.



We find that the optimal range of values for p for the test data is from 1 to 3, with *p=1* providing the best accuracy. We note that, as p increases past 3, the training error decreases but the testing error increases dramitically. This can be attributed to overfitting, as the higher degree polynomials can better fit the training data.

6 Problem 6

6.1 Interpreting Results

Based on the previous results, it would seem using Ridge Regression is the best approach in terms of training and testing error for anyone using regression for predicting diabetes levels using the input features. In application, it would probably be best to use Gradient Descent in order to avoid computational issues with the closed-form version. Using non-linear regression might provide higher accuracy (at least in terms of training error), but can lead to overfitting quickly, even with regularization.

Anyone using regression will want to use the MSE of their testing set to choose the best setting. This will prevent them from selecting an approach that is prone to overfitting or isn't accurate.