Department of Computer Science and Engineering

University at Buffalo

CSE4/574 Fall 2021 Introduction to Machine Learning

Programming Assignment 1

Classification and Regression

Group 23

Kudari, Sai Vasanth Vinay (vkudari@buffalo.edu)

Grisafi, Steven (sgrisafi@buffalo.edu)

Dahn, Christina (cdahn2@buffalo.edu)

REPORT 1

We used the linear discriminant analysis (LDA) and quadratic discriminant analysis techniques on the provided training data in order to both approximate a set of hyperplanes (technically hypersurfaces for QDA) dividing the solution space (via code provided by the instructor) and to find MAP estimates for the provided test set in relation to their given true values.

We used the equation below for QDA, neglecting the denominator because the denominators will all be identical for each class. The prior was also omitted because the function template for this assignment did not accept the required data needed to calculate these values.

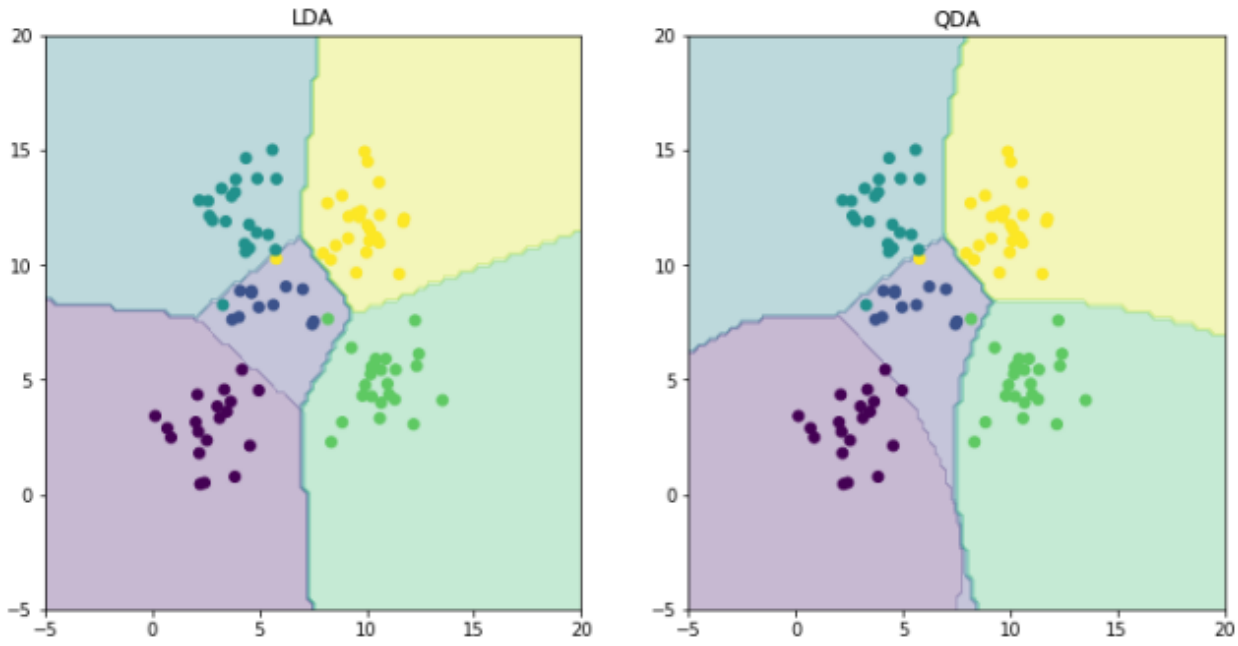
$$\Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}.$$

This is a posterior probability. Here, f_k is given below. For LDA, we substituted the class-specific covariance matrix for a shared covariance matrix defined below. The prior likelihood term is also shown.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1}(x-\mu_k)}.$$

- $\hat{\pi}_k = N_k/N$, where N_k is the number of class-k observations;
- $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$;
- $\hat{\mathbf{\Sigma}} = \sum_{k=1}^{K} \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T/(N - K)$.

We observed differences in the boundary shapes due to the inherent natures of linear versus quadratic discriminant analysis. That is, specifically, that by using a shared covariance matrix, the boundary equations are linear in x when Pr(G = k|X = x) = Pr(G = k|X = x), yet class-specific covariance matrices would make this equation nonlinear (specifically, quadratic) in x. Plots generated are provided below. Some slight curvature can be observed in the QDA plot.



Both techniques performed comparably well, with accuracy rates of 97% for LDA and 96% for QDA. The lower performance of QDA on test data may be attributed to overfitting to training data via quadratic, rather than linear, boundaries.

REPORT 2

We used linear regression with an ordinary least squares estimate loss function in order to generate the weight vector of the regression line for the provided data. In one case, we added a bias (intercept) to the computation, and in another we did not. That is, for data with p features, we used both an N x p and N x (p + 1) X matrix, and utilized the equation below to compute the linear equation.

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

The mean squared errors obtained are summarized below. As expected, adding an intercept produced lower MSE and was therefore preferable. This can be attributed to removing the necessity of forcing the regression line to pass through the origin.

MSE VALUES	Test data	Training data
without intercept added	106775	19099
with intercept added	3707	2187

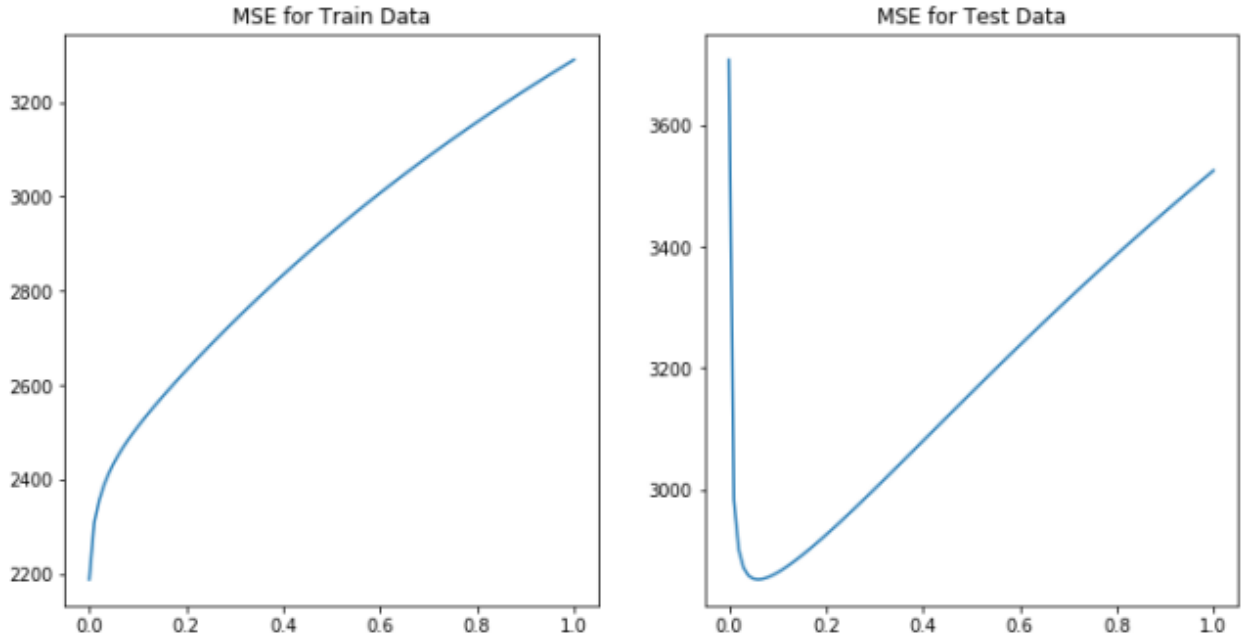
REPORT 3

Similar to Problem 2 we estimated regression. However, we used the ridge regression technique in this problem, utilizing an L2 norm term added to the ordinary least squares estimate for the loss function in order to inhibit overfitting. The loss function is provided below.

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

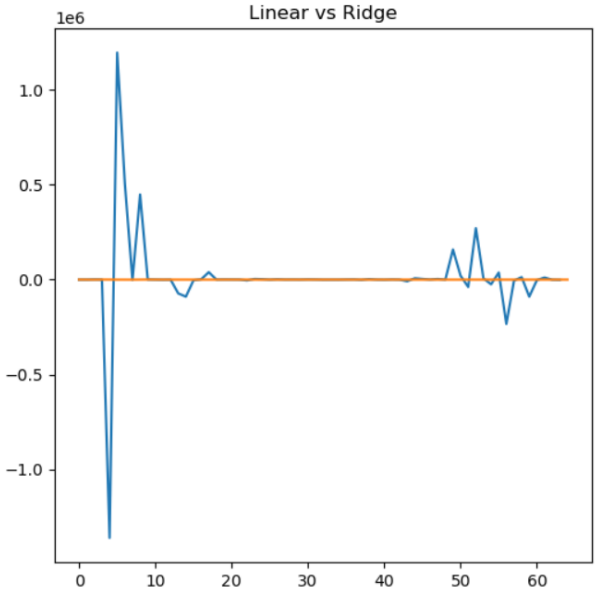
Here, λ was varied from 0 to 1 in steps of 0.01. The plots of λ vs mean squared error are given below. The equation used to compute the slope with intercept is given below as well.

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$



An optimal λ value of 0.06 was observed. The MSE are reported below. Using optimal λ, ridge regression underperformed linear regression on training data, but outperformed linear regression on test data, with respect to MSE. Weights observed in ridge regression remained relatively low and consistent when compared to weights obtained by linear regression. This is depicted in the graph below. Also, performance was observed to decrease on test data as λ increased, consistent with expectations of overfitting reduction.

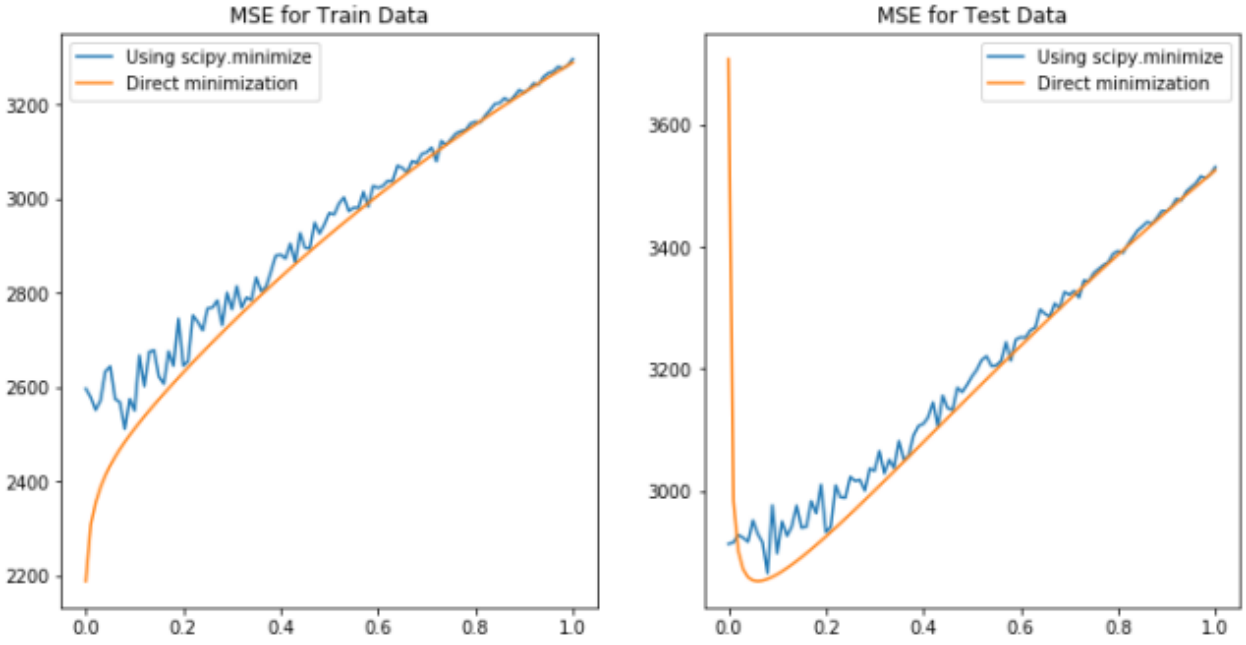
MSE VALUES	Test data	Training data
without intercept added	38923	21704
with intercept added	2851	2451



The above graph plots the weights of the 64 individual features with blue representing linear regression and orange representing ridge regression.

REPORT 4

We used gradient descent to solve the same ridge regression problem presented in Problem 3. Plots for mean squared errors versus λ were obtained for both training and test data. They are provided below. Again, λ was varied from 0 to 1 in steps of 0.01.



The above graphs show direct minimization (Problem 3) versus gradient descent via Scipy (Problem 4), with MSE plotted against λ. Gradient descent was observed to be slightly less accurate, with MSE values fluctuating up and down, as opposed to generating a smooth curve like direct minimization did. Optimal λ values differed as well, with direct minimization generating an optimal λ of 0.06, and gradient descent via Scipy generating an optimal λ of 0.75.

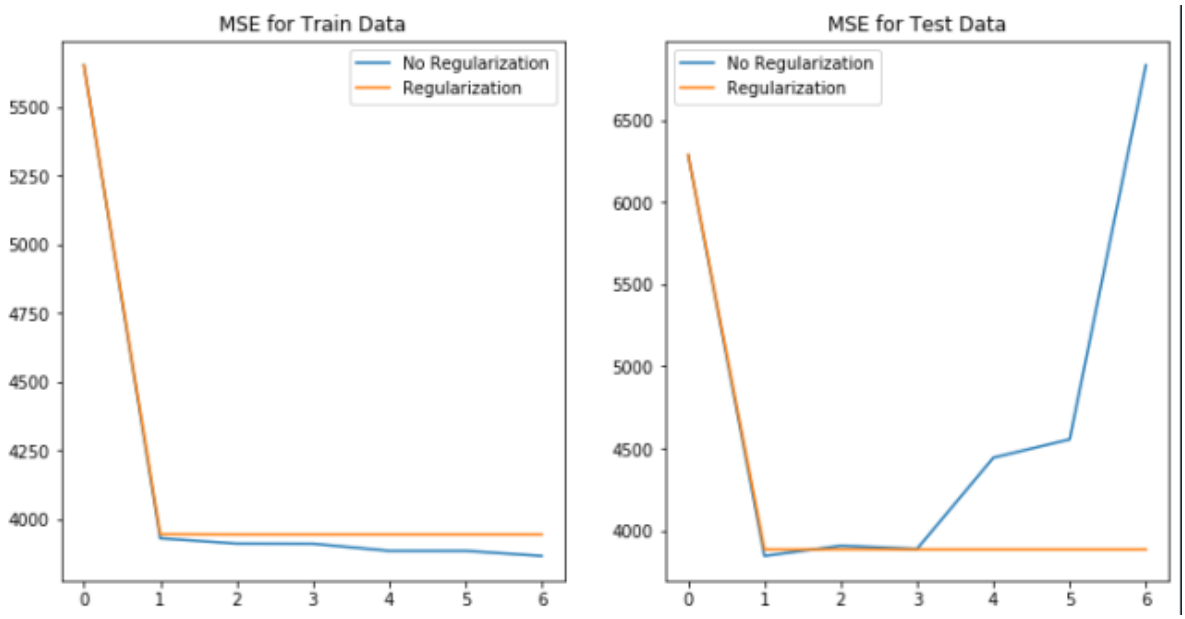
REPORT 5

Using λ = 0, as well as the optimal λ value obtained from Problem 3, and the same ridge regression loss function from Problems 3 and 4, we generalized the procedure in Problem 4 to fit to polynomials of degree p = 0-6. Errors on train and test data are summarized below.

MSE Values						
Lambda = 0.06						
Training Data						
Polynomial Degree						
0	1	2	3	4	5	6
5651	3952	3951	3951	3951	3951	3951
Test Data						
Polynomial Degree						
0	1	2	3	4	5	6
6287	3896	3896	3896	3896	3896	3896
Lambda = 0						
Training Data						
Polynomial Degree						
0	1	2	3	4	5	6
5961	3931	3912	3911	3885	3885	3867
Test Data						
Polynomial Degree						
0	1	2	3	4	5	6
6286	3845	3907	3888	4443	4555	6833

Both the regularized (λ = 0.06) and unregularized (λ = 0) cases performed comparably well on training and test data when fit up to polynomials of degree 3. As polynomial degree was increased beyond degree 3, overfitting was observed, with the unregularized case performing slightly better on training data, and the regularized case performing far better on test data, with respect to MSE.

Below, MSE is plotted versus polynomial degree.



REPORT 6

While our results pertained specifically to the task of predicting diabetes, we observed trends that should theoretically generalize well across many regression problems. Unregularized regression performed better at reducing MSE error on training data than regularized ridge regression, yet ridge regression performed far better on test data due to its ability to reduce overfitting. In general, this can be expected to hold better for datasets that have higher variances and lower sample sizes. In this project, sample size was quite low.

Similarly, fitting to polynomials of too low or too high degree was observed to generate significant underfitting and overfitting effects, respectively. Again, datasets with higher variances and lower sample sizes would be expected to be more vulnerable to this type of overfitting. However, this may not necessarily be true for datasets in which the test and training data are highly consistent.

Additionally, approximation via gradient descent, while fairly accurate, was not completely consistent with results generated by direct minimization.

In conclusion, the most accurate technique for predicting diabetes, according to our computations, was to use a ridge regression on a polynomial of degree 1 (linear), with the regression computed directly, rather than via gradient descent. With linear regression and with ridge regression results were comparable for degree 1 polynomial.

Results obtained with and without regularization were comparable for degree 1 polynomials.