Programming Assignment 1

University at Buffalo

Introduction to Machine Learning CSE 474/574

Jose A. Barajas
Majd Abu ghazaleh

Question 1

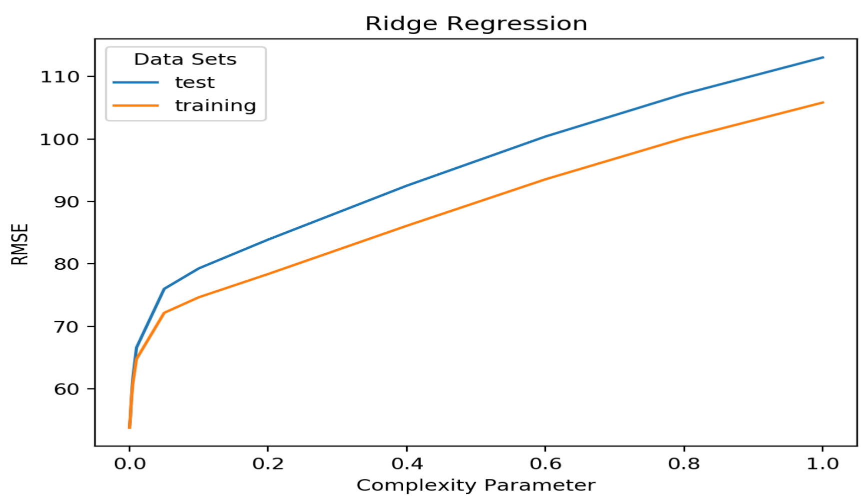
For this problem, a linear least squares model was used on the diabetes data set. The weights were learned manually by solving the exact solution as noted below:

$$w = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

The main issue with the exact solution involves the inverse term. Given the nature of the inputs, the inverse can be unstable. This was noticed as in the order of operations had a significant impact on the results. Anyway, the following formulation yielded an RMSE of 159 and 168 for the training sets and testing sets respectively. This makes sense, it is typical that the training set RMSE will be lower than the test RMSE. Though the discrepancy in our case is no cause to assume complete overfitting. More significantly, after adding a bias (which was done by adding a column of ones into the feature matrix) yielded a RMSE of 53.75 and 53.94 for the training and test sets respectively. Not only is overfitting virtually non-existent, but the RMSE is significantly much lower.

Question 2

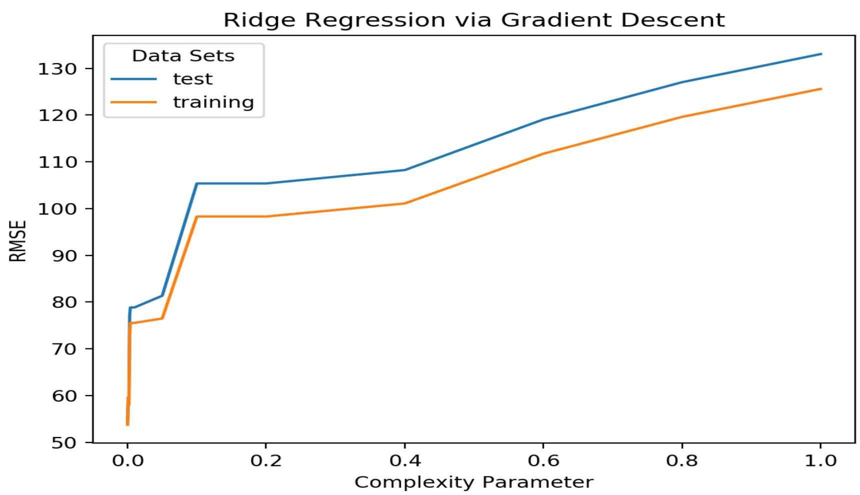
Similarly, for this problem, the RMSE was calculated like before, except ridge regression was implemented. For this case, ridge regression was tested with a bias over a spectrum of lambda values. Lambda values impacting the "penalty" on the model. The results are outlined below in the following figure:



Just like before, it is evident that the training set performed better than the test set. Though it seems as though the best model is one in which lambda is zero or close to zero. Thus, the ridge regression model is comparable to the linear one since a ridge regression model with lambda equals 0 is the same model.

Question 3

Approaching problem three in a different manor, instead of using the exact solution, gradient descent was implemented to find the optimal solution. This was done using scipy's minimize function which implements conjugate gradient descent. The following plot outlines the result using gradient descent:



Actually, the RMSE is quite comparable to to RMSE using the exact solution, though this method showed less issues involving singular matrices and instability.

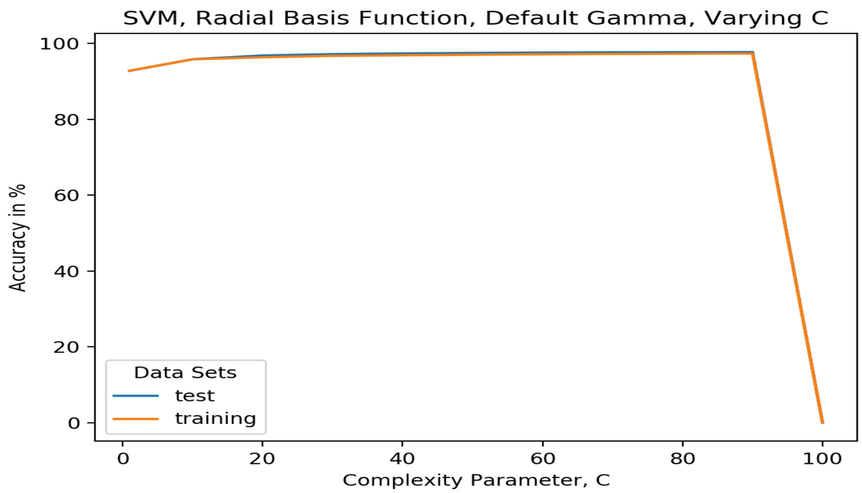
Question 4

For logistic regression. Though it was most certainly fast to implement, when (as we will note below) compared to SVM, it does not perform as well. Our logistic regression model was implemented by minimizing the error via conjugated gradient descent. Out model had a maximum accuracy of 92% while SVM had a maximum of around 98 %. Though we for certain would consider logistic regression to be viable given the type of data you are working with. For instance, given a high dimensional binary classification problem with high number of observations, logistic regression may be the preferred option.

Question 5

A support vector machine (SVM) was trained in python to classify the ijcnn dataset. SVM models were assessed by changing kernel type, gamma setting, and cost. The following table and plot summarizes the results:

Kernel	Gamma	Cost	Accuracy (%)
Linear	N/A	0	92.1
RBF	1	0	98.4
RBF	Default	1	92.7
RBF	Default	10	92.8
RBF	Default	20	95.8
RBF	Default	30	96.8
RBF	Default	40	97.2
RBF	Default	50	97.37
RBF	Default	60	97.5
RBF	Default	70	97.6
RBF	Default	80	97.7
RBF	Default	90	97.72
RBF	Default	100	0



SVM methods as seen above when tested with the ijcnn dataset are not exhibiting overfitting and overall perform well. The best model was one that uses a radial basis function as the kernel. Specifically one that had a gamma of 1 while keeping all other parameters at default. Though after varying the cost, it can be noted that increasing the cost increases the accuracy (to a point).