Yixiao Zhang
31353282
yz3a19@soton.ac.uk

Regression and Regularization

COMP6245 Lab 4

1 Linear Least Squares Regression

442 samples from diabetes dataset is loaded using sklearn package. This dataset includes 10 features and a label. The label of this dataset is numeric which is from 20 to 350 as Figure 1 shows. Scatter of two input is helpful to understand the data better. Figure 2 shows 7th and 8th feature of the dataset.

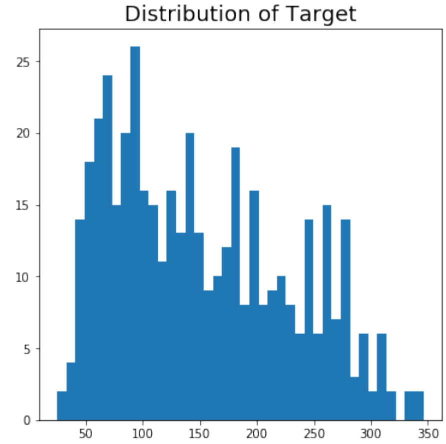

Figure 1: Distribution of label

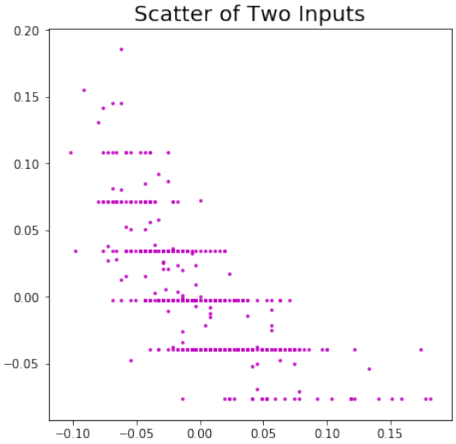

Figure 2: Scatter of 7th and 8th feature

Two linear predictor are trained using dataset above. Figure 3 shows the prediction result of sklearn predictor, whereas Figure 4 shows the prediction result of pseudo-inverse method predictor which implement by formula $a = (Y^tY)^{-1}Y^tf$. In Figure 3 and Figure 4, one axis is ground truth and another axis is prediction result which means that the point which near the diagonal the prediction is more accuracy. From the experiment, the error of sklearn predictor is 1.264×10^6 and the error of pseudo-inverse predictor is 1.266×10^6. On the other words, the accuracy of these two predictor have no significant difference.

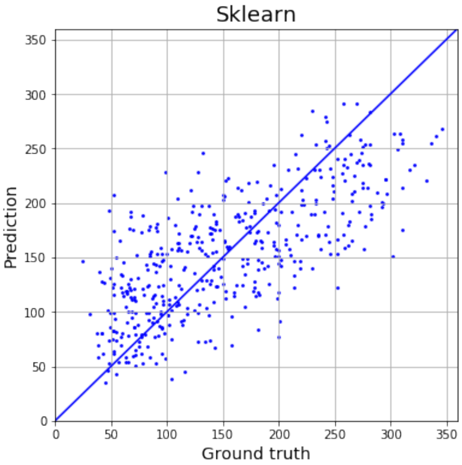

Figure 3: sklearn predictor

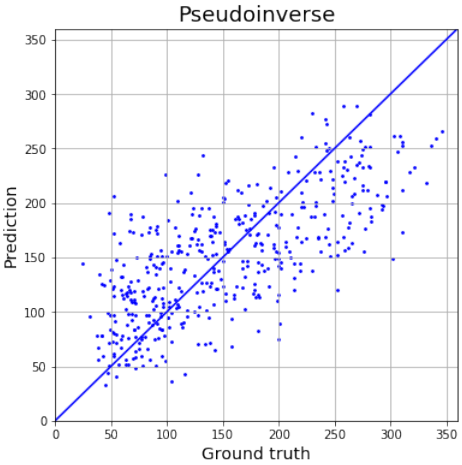

Figure 4: pseudo-inverse predictor

2 Regularization

The parameters of pseudo-inverse predictor may very big which may difficult to calculate and may increase computing time. Tikhonov regularization is a good way to minimize the mean squared error with a quadratic penalty on the weights.

Figure 5 shows the parameters of pseudo-inverse predictor, the 5th and 9th parameter are extremely large. Figure 6 shows the parameters after Tikhonov regularization, almost every parameters smaller than before. This regularization helps predictor perform quicker but it leads error bigger as well. The error of pseudo-incerse predictor is 1.27×10^6 and the error of regularized pseudo-incerse predictor is 1.35×10^6.

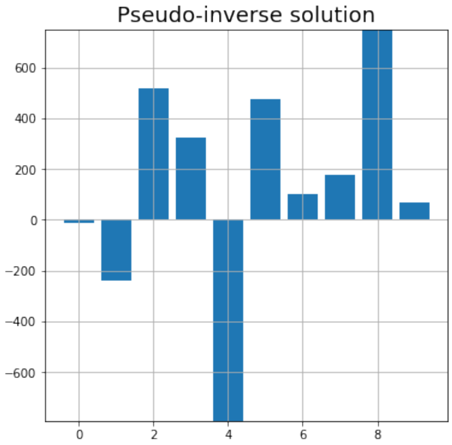

Figure 5: Parameters of pseudo-inverse predictor

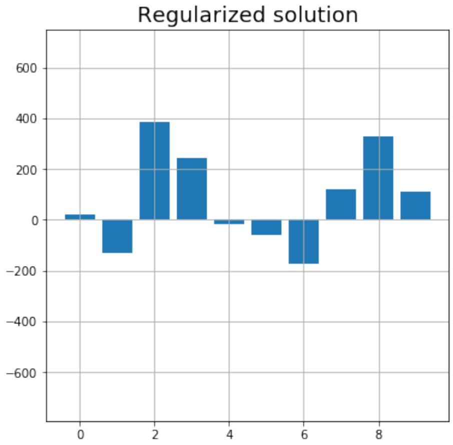

Figure 6: Parameters after regularization

3 Sparse Regression

Sparse regression helps predictor to decrease the number of parameters. Figure 7, Figure 8 and Figure 9 shows the non-zero weights change with the regularization parameters alpha 0.1, 0.5 and 0.9 respectively. With the increasing of alpha, the number of non-zero weights dropped whereas the error increased. The error of these three solutions are 1.29×10^6, 1.43×10^6 and 1.62×10^6 respecitvely.

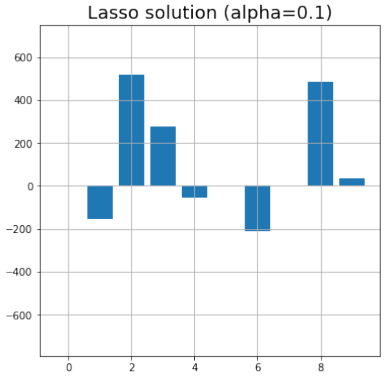

Figure 7: Parameters of solution (alpha=0.1)

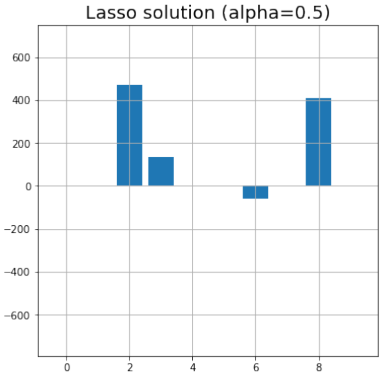

Figure 8: Parameters of solution (alpha=0.5)

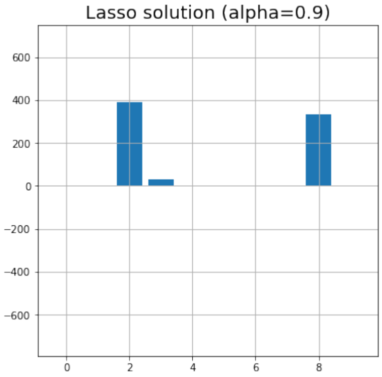

Figure 9: Parameters of solution (alpha=0.9)

To better understand the process of regularization, a lasso solution regularization path is generated as Figure 10 shows. From Figure 10, with increasing of alpha, the value of each parameters decreased and the number of non-zero weights increased.

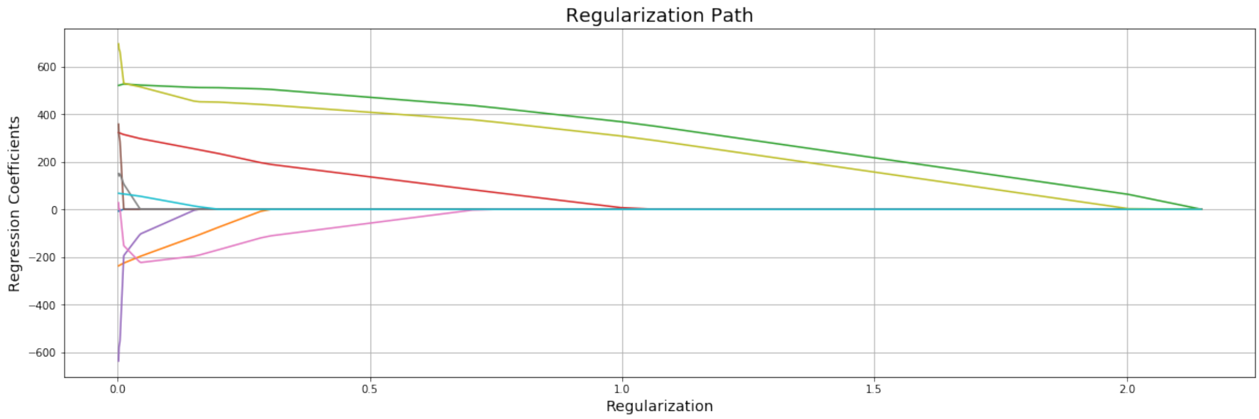

Figure 10: Regularization path

4 Solubility Prediction

A dataset related to chemical compounds is used in this section. Package pandas is used to load data from excel file. The distribution of Log Solubility is shown as Figure 11. The whole dataset is split to training set (70%) and testing set (30%) randomly.

A predictor is trained using training set. Figure 12 shows the prediction result of training set as well as Figure 13 shows the prediction result of testing set. The result of training set is much better than testing set, it may happen because the parameters are too much which lead over fitting problem. To solve this problem, it will be better if there is more data and split the whole data to training set, testing set and validation set or reduce the numbers of parameters(increase number of non-zero weights).

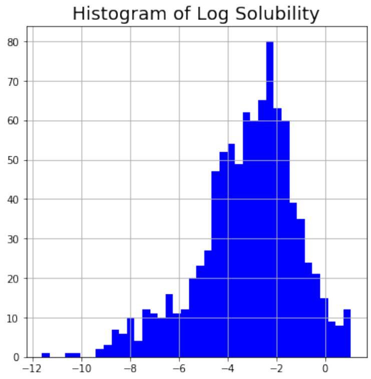

Figure 11: Distribution of Log Solubility

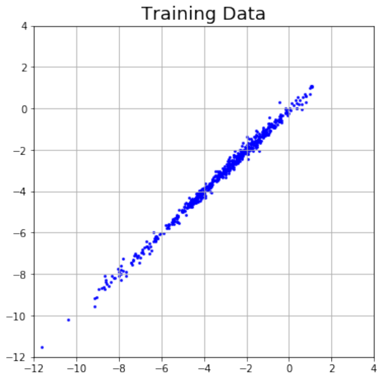

Figure 12: Prediction result of training set

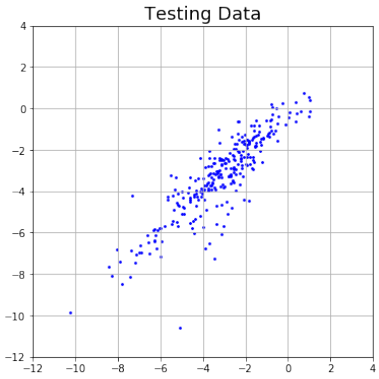

Figure 13: Prediction result of testing set

The number of non-zero weights and error in a lasso solution regularization process is described by Figure 14 and Figure 15. With the increasing of alpha, number of non-zero weights dropped dramatically and over-fitting problem reduced. From the Figure 15, the error of training is much lower than testing which is in over-fitting, but the gap between the training and testing bacame smaller with decrease of number of non-zero weights.

In addition, Decreasing number of non-zero weights could helps predictor runs quickly and save computation resources, but it could also increase the error in prediction. It is very important to find a balance between efficient and accuracy. The balance point is depend on which accuracy target and how much computation resources do users have.

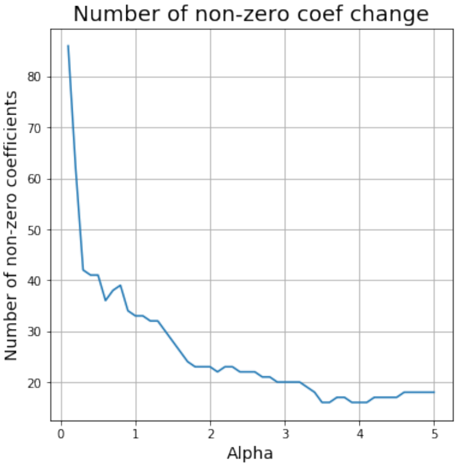

Figure 14: Number of non-zero weights change with increasing regularization

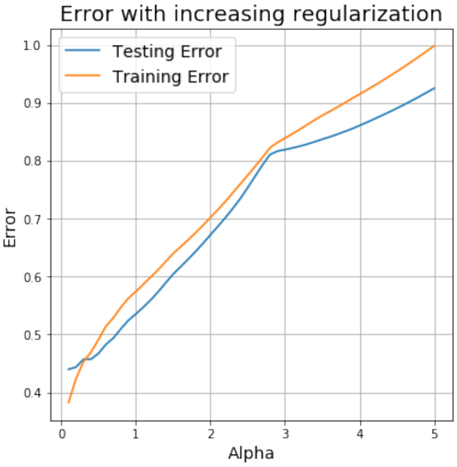

Figure 15: Error change with increasing regularization