COMP 6245 Report for Lab three

RUNYING JIANG (rj1u20@soton.ac.uk)

January 20, 2021

1 Linear Least Squares Regression

We deploy Linear regression model on **Diabetes** dataset from UCI Machine Learning repository. [?], which is imported from the package **sklearn**.And we take columns *LogS.M.* as target.

We first plot a few histograms of the targets and scatters of the features to have a better understanding of the dataset. which is **Figure 1** as follow.

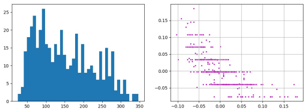

Figure 1: Diabetes dataset: The left colomn shows the distribution of diabetes target data while the right colomn shows the scatters of columns 6 and columns 7 of data to predict

We implement a linear predictor, which is solved by the pseudo-inverse method.

$$\mathbf{w} = (X^tX)^{-1} wX^t\mathbf{t}$$

Derived from

$$\min_{w} ||t - Xw||_2^2$$

Then we compare the linear predictor with the linear regression model imported from sklearn and got the following result **Figure 2** :

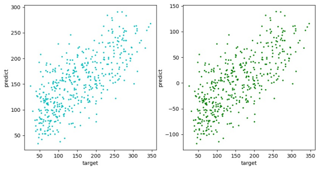

Figure 2: Comparison of two linear models: These two plots shows the scatters of predict data in horizontal axis and the result data in vertical axis. The left colomn shows the prediction result of sklearn while the right colomn shows the prediction result of pseudo-inversed linear predictor

Then we found there is a gap between the prediction and target data in pseudo-inversed linear model. To solve this problem we add an intercept which is the same as bias in the linear model. Following **Fig: 3**is the advanced result:

```
# Linear regression using sklearn
lin = LinearRegression()
lin.fit(X, t)
th1 = lin.predict(X)

# Pseudo-increase solution to linear regression
w = np.linalg.inv(X.T @ X) @ X.T @ t
th2 = X @ w


# Pseudo-increase solution
# to linear regression with intercept
O = np.ones((len(X), 1))
X2 = np.append(X, O, axis=1)
w2 = np.linalg.inv(X2.T @ X2) @ X2.T @ t
th3 = X2 @ w2
```

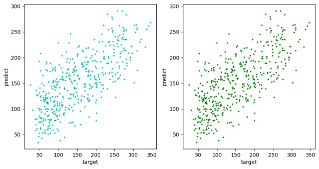

Figure 3: Advanced Comparison of two linear model:These two plots shows the scatters of predict data in horizontal axis and the result data in vertical axis, which is an advanced result compared to the last fig.

2 Regularization

Tikhonov regularization which is L2 regularization minimizes the mean squared error with a quadratic penalty on the weights:

$$\mathbf{w} = (\lambda I + X^t X)^{-1} w X^t \mathbf{t}$$

Derived from

$$\underset{w}{min}||t - Xw||_2^2 + \frac{\lambda}{2}||w||_2^2$$

We implement Ridge regularizers and Tikhanov(quadratic regularizer) separately. By setting gamma parameter in Tikhanov(quadratic regularizer) to 0.2, which equals to the alpha parameter in Ridge regularizers imported from sklearn , we get the same result **Fig: 4** as we expected.

```
# Tikhanov (quadratic) Regularizer
gamma = 0.2
wR = np.linalg.inv(X2.T @ X2 +
gamma*np.identity(NumFeatures+1)) @ X2.T @ t

# Ridge Regularizer
l2 = Ridge(alpha=0.2)
l2.fit(X, t)
th_ridge = l2.predict(X)
```

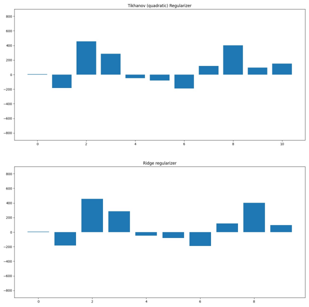

Figure 4: Comparison of L2 regularizers These two plots shows the bars of L2 regular weights. The first bar shows the weights got from derived formula while the second shows the weights from L2 regularizer of sklearn

3 Sparse Regression

L1 regularization which is Lasso regularization minimize the mean squared error with a penalty on weights:

$$\underset{w}{min}||t - Xw||_2^2 + \frac{\lambda}{2}||w||_1$$

We plot the resulting weights of Pseudo-inverse solution, Lasso solution,Regularized solution as follow **Fig: 5**

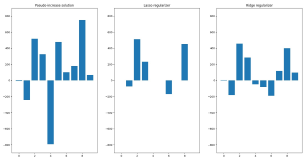

Figure 5: Comparison of different regularizers These plots shows the weights of three different regularizers used in **Diabete Dataset**.The horizontal axis is the index of the features while the vertical axis is the weights of the features.

After a deep look at the source of the data and the variables. I found the weights are partly meaningful.

feature	meaning of features
age	age in years
sex	sex
bml	body mass index
bp	average blood pressure
s1	tc, T-Cells (a type of white blood cells)
s2	ldl, low-density lipoproteins
s3	hdl, high-density lipoproteins
s4	tch, thyroid stimulating hormone
s5	ltg, lamotrigine
s6	glu, blood sugar level

Table 1: Characteristics of Diabetes datasets

From the result of **Fig :5**, we find the diabetes is highly related to *body mass index*,*average bood pressure*, *lamotrigine*.The weights of *body mass index*,*average bood pressure* is reasonable. But the weights of*lamotrigine* which is a medicine used to treat epilepsy in common do not make sense. Meanwhile, the plot shows diabetes has a certain relationship with *sex*, which is beyond our common sense. Besides, the weight of *T-Cells* is quite high in Preusdo-increase solution regularizer while it is almost zero in Lasso and Ridge regulatizers.

Regularization Path

We implement lasso regularizer on random Dataset X and y. Then we plot the regularization path in **Fig: 6** which indicates how regression coefficients change with the hyperpatameter.

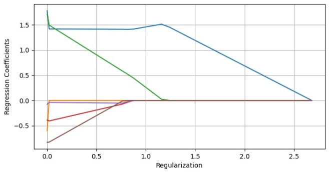

Figure 6: Regularization Path The horizontal axis is the hypterparameter while the vertical axis is the coefficients

4 Solubility Prediction

We first load the data of the excel Spread sheet **Husskonen_Solubility_Features.xlsx** and split into training set (80% in portion) and test set(20% in portion). To optimize, we use MinMaxScale on the data. Then we implement a linear regression and the scatter plots of training and test sets are as follow**Fig: 7**. The results indicate the training model is over-fitting which means it behaves quite well on training data but has a bad performance on test data.

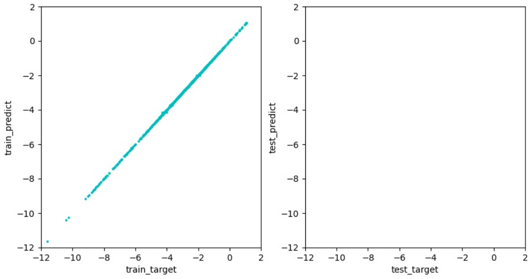

Figure 7: Linear regularizer result after scale:The left plot shows the scatter of training set while the right one shows the scatter of test set

By implementing a lasso regularized solution, there is an improvement on the performance of the test data.

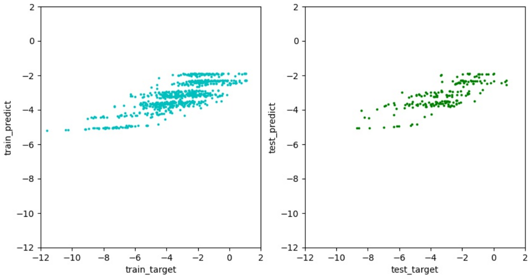

Figure 8: Lasso regularizer after scale:Scatter Plot after scale:The left plot shows the scatter of training set while the right one shows the scatter of test set

We iterate from zero to two with the step of 0.01 to see how the prediction error and the corresponding number of non-zero coefficients change with increasing regularization. From **Fig: 9**, we can conclude that the error of prediction increases discontinuously with the alpha value. Meanwhile,**Fig: 10** indicates that there is an continuous decrease in the number of none-zero coefficients as the increase of alpha value.

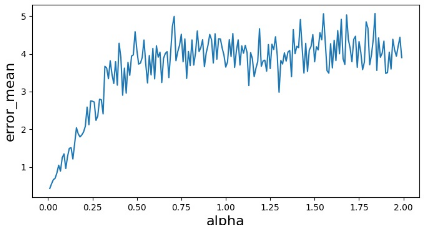

Figure 9: Error of prediction with lasso:The horizontal axis is the alpha value range from zero to two at an interval of 0.01 while the vertical axis is the prediction error of lasso

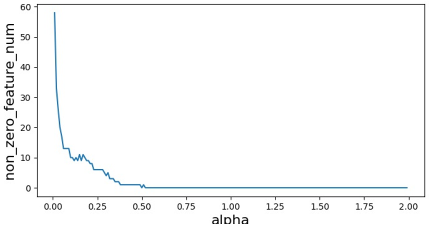

Figure 10: None-zero coefficients with lasso:The horizontal axis is the alpha value range from zero to two at an interval of 0.01 while the vertical axis is None-zero coefficients with lasso

We select the top ten features which weigh most to predict solubility.

The features are **MLOGP2**, **B07[C-C]**, **B01[C-O]**, **B01[C-Cl**, **B03[C-N]**, **B02[C-O]**, **piPC10**, **B02[C-Cl**, **B04[C-Cl]**, **B05[C-Cl**

We reduce the dataset to the shape of (932,10). Then, we implement and compare the result of Linear regularizer, Ridge regularizer and Lasso regularizer on the new dataset. The prediction error are as follow **Table: 3**

solution	error before	error after
linear regularizer	81873374130880.66	0.7926
Ridge regularizer	10.5549	0.6256
Lasso regularizer	4.3244	2.3766

Table 2: My prediction accuracy over the feature selected

We could see the accuracy has been improved in these three models which means our selected feature worked. Besides, there is a huge improvement in linear regression. The scatters of the results are as follow which shows the improvement in a more direct way. **Figure : 11 12 13**

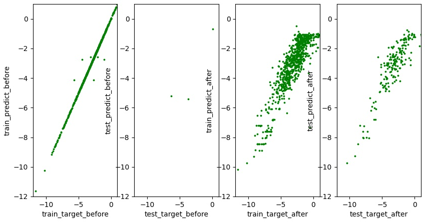

Figure 11: The scatter of linear model: the horizontal axis is the target data while the vertical axis is the predict data.

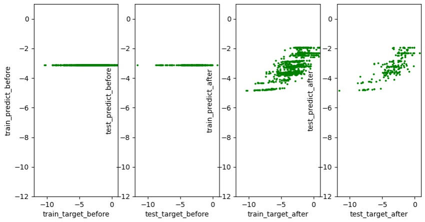

Figure 12: The scatter of lasso model: the horizontal axis is the target data while the vertical axis is the predict data

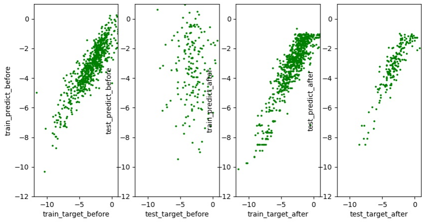

Figure 13: The scatter of ridge model: the horizontal axis is the target data while the vertical axis is the predict data

4.1 Comment on result

After having a brief look at the reference given , we compare to the result of reference[3] test set, a predictive r2 of 0.86 and s of 0.53 (log units) were achieved, which shows we get a compared good result by selecting feature by L1 model and run in L2 model.

solution	r2 before	r2 after
linear regularizer	-9535561319750.785	0.7568
Ridge regularizer	-1.2634	0.8125
Lasso regularizer	-0.0055	0.4722

Table 3: My prediction accuracy over the feature selected