Financial Technology: Assignment 1

Oct. 2020

B07703014 財金三 蔡承翰

[Problem 1: Linear Regression]

(a) To obtain the training and test set, we first read the data file into *pandas.Dataframe*. We extract the columns that we are interested in and get dummy variables. Later, we import *random*, setting the seed into *703014*. With the fixed seed, we can extract 200 samples from data pseudo-randomly. Let these chosen sample be the test set, and the rest be the training set. We then extract the target column $G3$ from theses sets respectively and transform them into column vector (*numpy.ndarray*). Finally, we normalize the two sets with the mean and standard deviation of the training set. That is, for $a_{,j}$ being a column in the training set, we do

$$a_{i,j} \coloneqq \frac{a_{i,j} - mean(a_{,j})}{std(a_{,j})}$$

And for $b_{,j}$ being a column of the test set, we do

$$b_{i,j} \coloneqq \frac{b_{i,j} - mean(a_{,j})}{std(a_{,j})}$$

Here, we use the notation $\coloneqq$ to distinguish assignment from equivalence. Finally, transforming $(a_{i,j})_{800\times 24}, (b_{i,j})_{200\times 24}$ into *numpy.ndarray* object should finish the preprocessing.

(b) We use normal equation to obtain the coefficient betas. Let X_{train} be the training set, Y_{train} be the target of the training set. We calculate

$$\hat{w} = (X^tX)^{-1}X^tY = pinv(X)Y$$

where $pinv(X)$ stands for pseudo-inverse of X. We apply the $betas$ to test set to do prediction. Denote X_{test}, Y_{test} as the test set and the target of it respectively. We write

$$predict = X_{test}\hat{w}$$

and

$$RMSE = \sqrt{mean((Y_{test} - predict)^2)}$$

Our program result gives us $RMSE = 11.60$.

(c) In this case, we consider

$$\min_{\hat{w}} J(\hat{w})$$

Notice that in this case,

$$J(\widehat{w}) = MSE_{train} + \frac{1}{2}\widehat{w}^t\widehat{w} = \frac{1}{n}\left||Y_{train} - X_{train}\widehat{w}|\right|^2 + \frac{1}{2}\widehat{w}^t\widehat{w}$$

$$= \frac{1}{n}(Y_{train} - X_{train}\widehat{w})^t(Y_{train} - X_{train}\widehat{w}) + \frac{1}{2}\widehat{w}^t\widehat{w}$$

$$= \frac{1}{n}\left(\left||Y_{train}|\right|^2 + \widehat{w}^t X_{train}^t X_{train}\widehat{w} - 2Y_{train}^t X_{train}\widehat{w}\right) + \frac{1}{2}\widehat{w}^t\widehat{w}$$

Consider the first-order condition. To simplified, let $X = X_{train}, Y = Y_{train}$

$$\nabla J(\widehat{w}) = \frac{1}{n}(2X^tX\widehat{w} - 2X^tY) + \widehat{w} \;=\; 0$$

$$\Rightarrow \left(\frac{1}{n}X^tX + \frac{1}{2}I\right)\widehat{w} = \frac{1}{n}X^tY \Rightarrow \widehat{w} = \left(\frac{1}{n}X^tX + \frac{1}{2}I\right)^{-1}\frac{1}{n}X^tY$$

where $n = 800$ is the count of samples in the training set. Since the coefficient vector is revealed, we are able to calculate $RMSE$ by the same operation from part (b). The result is $RMSE \;=\; 11.59$, which is almost the same from part (b).

(d) The coefficient vector w is the same as part (c), but in this model we consider

$$Y = Xw + b$$

where b is called bias. With w, we can find b

$$\hat{b} = mean(Y - X\widehat{w})$$

So now our prediction will be

$$predict \;=\; X\widehat{w} + \hat{b}$$

And the calculation of $RMSE$ are still the same. The result is $RMSE = 3.39$.

(e) Though the Bayesian Linear Regression model considers w as a random vector, in this case we only need to find the mean of the posterior distribution. From the textbook we write

$$\Lambda_m = (X^tX + \Lambda_0^{-1})^{-1}$$
$$\widehat{w} = \mu_m = \Lambda_m(X^tY + \Lambda_0^{-1}\mu_0)$$

In this case we fix $\mu_0 = 0,\ \Lambda_0 = \frac{1}{\alpha}I$. Calculating prediction and $RMSE$ are all the same as part (d). Our result shows that $RMSE \;=\; 3.43$.

(f) Predictions from part (b), (c), (d), (e) should give us the following figure.

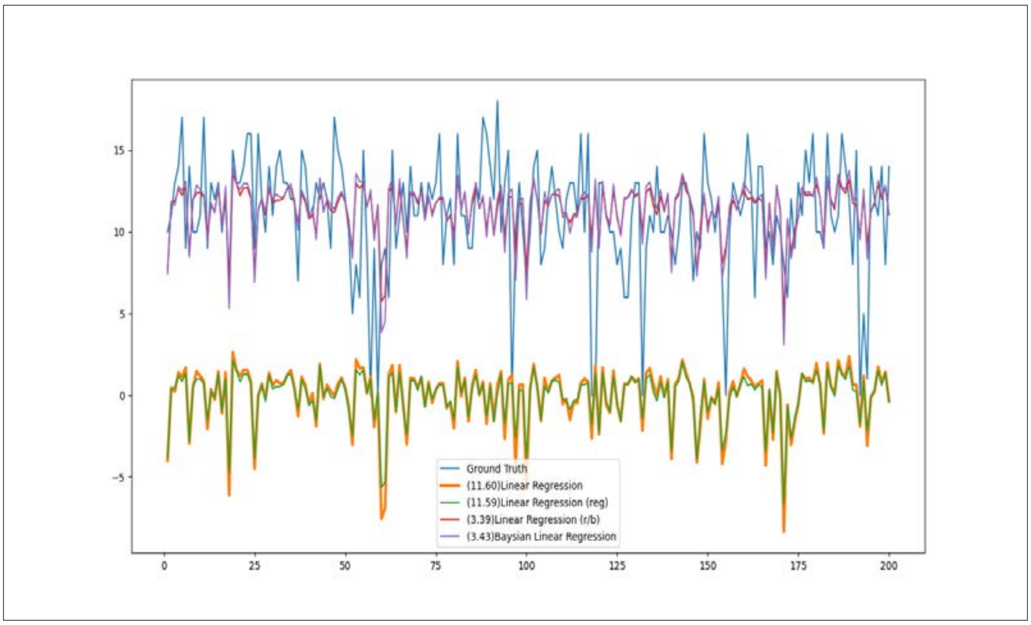

Figure 1

The main difference is because that part (d) and part (e) have considered the bias term. It can be proof that if the true model is with bias (intercept) and the bias hasn't been considered in estimation of $\hat{w}$, then $\hat{w}$ is a biased estimator. Since that

$$MSE = Bias(\hat{\theta})^2 + Var(\hat{\theta})$$

Hence, if $Bias(\hat{\theta})^2$ not zero, MSE is much larger, and the prediction is weaker.

(g) We perform a hyperparameter sweep to tune α, and obtain the following result.

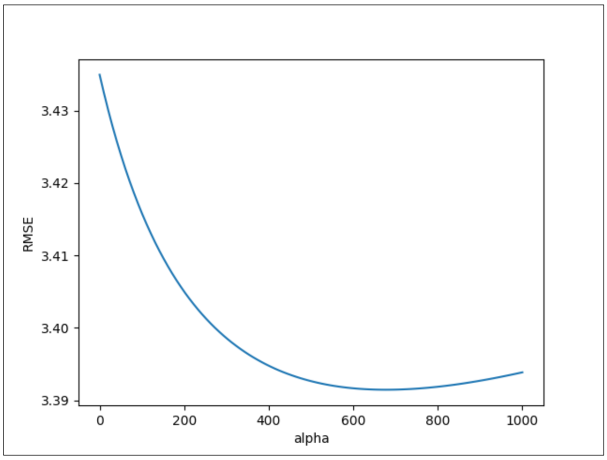

Figure 2

We find out that the optimal α is 676.8 and $RMSE = 3.39$.

[Problem 2: Census Income Data Set]

(a) Basically, the preprocessing is the same as problem 1. Nevertheless, owing to the inconsistency of the dummy variables, we should only focus on those variables in the intersection of the training set and the test set. Moreover, as the target itself is a dummy variable, we only need to predict the column “>50K”, where 1 means “>50K” and 0 means “<=50K”. Since the methods are all the same as problem 1. We simply give the result.

RMSE from part (b): 0.41

RMSE from part (c): 0.42

RMSE from part (d): 0.34

As of part (e), again, we perform a hyperparameter sweep to tune α, which gives us the following result.

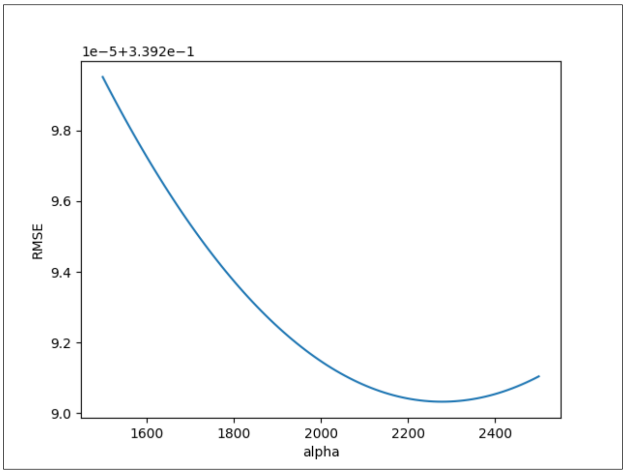

Figure 3

In this part the optimal α is 2277.78 and the corresponding $RMSE$ is 0.3393 (though tuning α brings only little improvement). In conclusion, we obtain the following graph to present the total result.

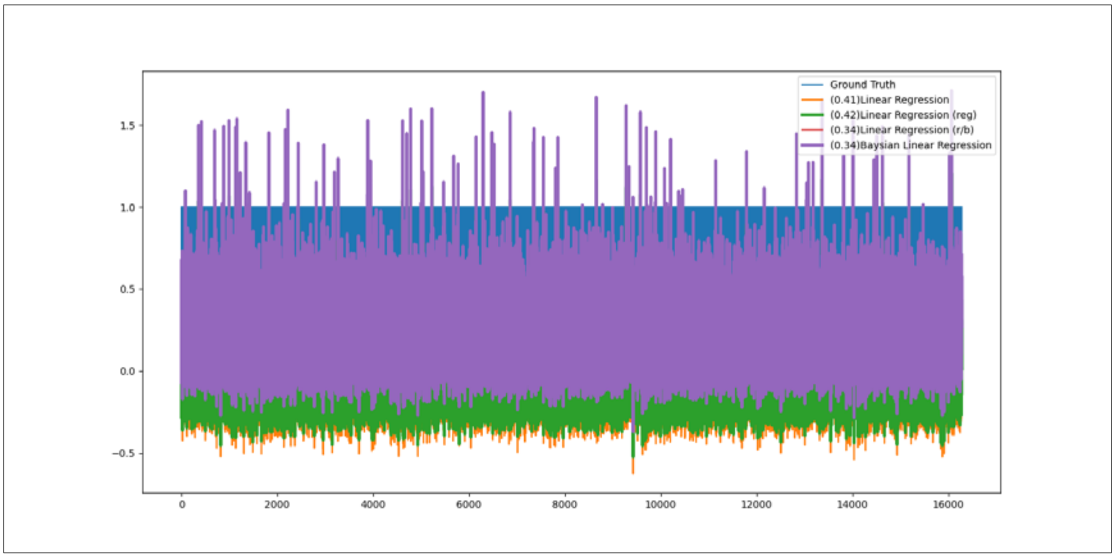

Figure 4

Again, we observe that considering bias in our model can greatly improve our prediction quality.