2019 Spring COM526000 Deep Learning - Homework 1 Report

Machine Learning Basics: Regression and Classification

105061110 周柏宇

Problem 1

(a) Please elaborate on how you obtain your training and test sets in your report.

First, I load the csv file with “pandas.read_csv()”. According to the instruction, we only need specific columns as our predictors, in which there might be some categorical features. In order to apply regression methods, we need to convert categorical columns into one-hot encoded columns, and we can simply achieve this by using “pandas.get_dummies()”.

Now we are ready to split the data into training and testing set, to do so, I use “dataframe.sample(frac=.8, random_state=42)”. We need to specify two parameter: “frac=.08” means we will randomly draw 80% of data as out training set. Since we need to use the same training set throughout the problem, we will use a fix random seed to generate the same sample, that’s what random_state does. Testing set is obtain by dropping the training set from the original set. Finally, it would be convenient to separate predictor columns from target column. Thus, I define X_train as our predictor columns by dropping the target column “G3”.

(c) Please describe how to find the optimal weights with maximum likelihood criterion in your report.

We derive the normal equation for the regression with regularization:

$$J(w) = MSE_{train} + \frac{\lambda}{2}w^T w = \frac{1}{n}\sum_{t=1}^{n}\left[y^{(t)} - w \cdot x^{(t)}\right]^2 + \frac{\lambda}{2}w^T w$$

The partial derivatives with respect to the weights should be 0:

$$\frac{\partial}{\partial w_i}J(w) = \frac{2}{n}\sum_{t=1}^{n}\left[y^{(t)} - \left(x^{(t)}\right)^T w\right]\left(-x_i^{(t)}\right) + \lambda w_i = 0$$

$$\rightarrow \frac{2}{n}\sum_{t=1}^{n} x_i^{(t)}\left(x^{(t)}\right)^T w + \lambda w_i = \frac{2}{n}\sum_{t=1}^{n} x_i^{(t)} y^{(t)}$$

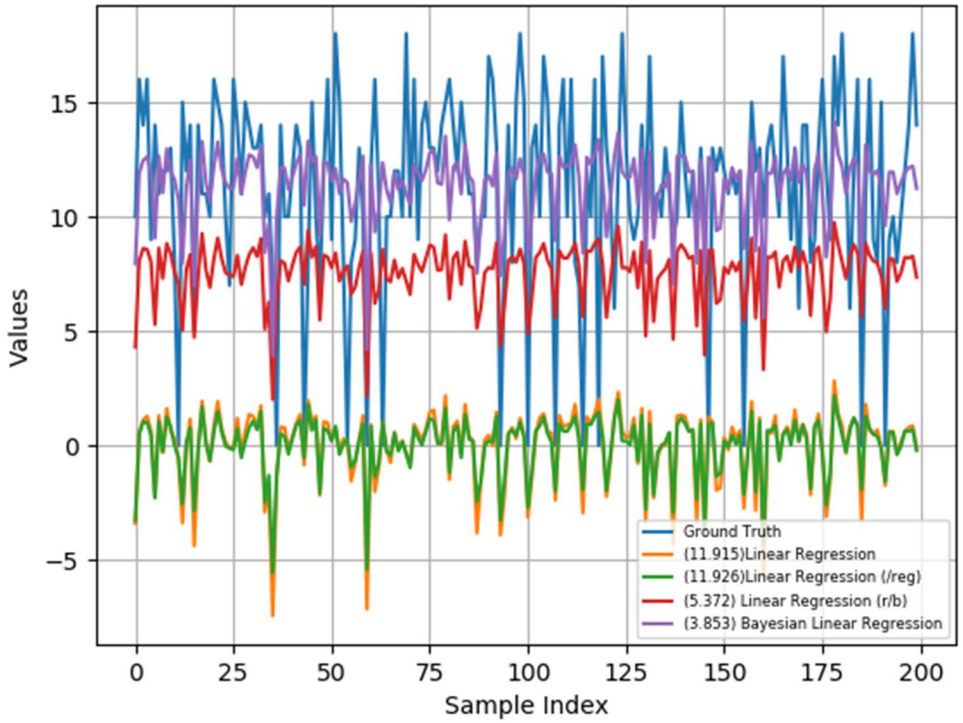
Collapse the equations for all partials:

$$\frac{2}{n}\sum_{t=1}^{n} x_i^{(t)}\left(x^{(t)}\right)^T w + \lambda w = \frac{2}{n}\sum_{t=1}^{n} x^{(t)} y^{(t)}$$

$$\rightarrow \frac{2}{n}X^TX + \lambda w = \frac{2}{n}X^TY$$

$$w = [X^TX + \frac{n\lambda}{2}I]^{-1}X^Ty$$

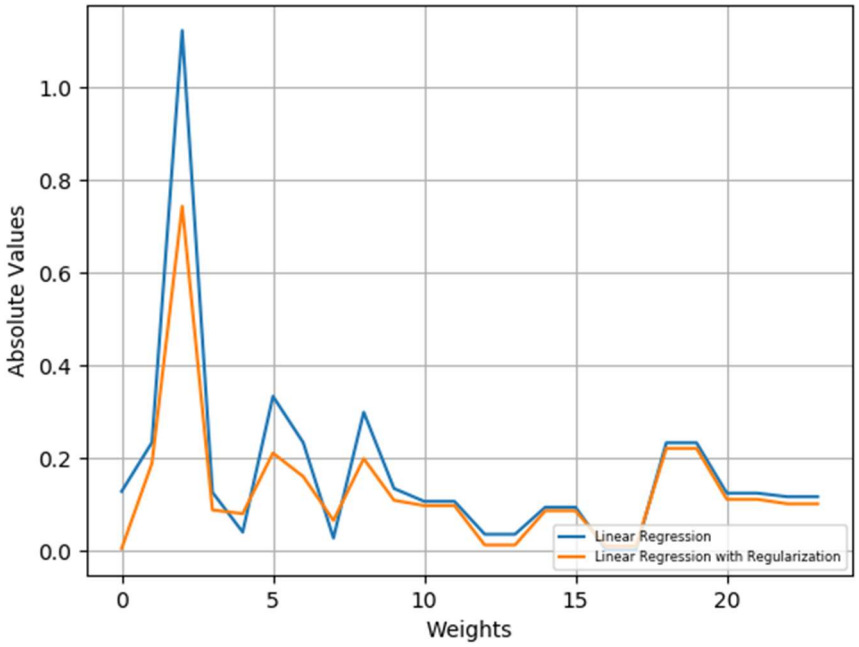
(f) Please compare the RMSE's and predicted G3 values in your report. Also, please explain mathematically why predicted G3 values are closer to the ground truth for (d) and (e).

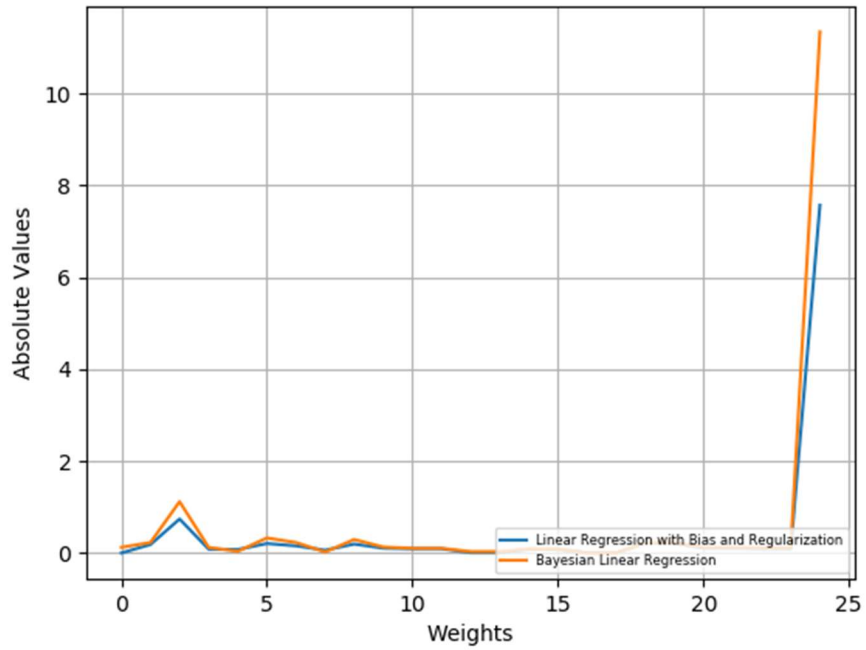


For the Linear Regression (lr) line and the Linear Regression with Regularization (lr/reg) line, they both have similar RMSE, but the predicted values for lr/reg are slightly smaller due to the regularization term. We can also observe the effect of regularization by plotting the weights, as illustrated below, lr/reg's weights are noticeably smaller.

For the Linear Regression with Regularization and Bias (lr/r/b) and Bayesian Linear Regression (blr), the RMSEs are significantly reduced, which can be attributed to the bias terms. The inclusion of the bias term adds another degree of freedom, which enhances our model's capability. Plus, there is no justification to assume the data points are distributed unbiased. We can further conclude that the RMSEs are mainly causes by the absence of bias by plotting the bias out, which is surprisingly large compared to coefficients.

Since we have proved that with the specific selection of μ_0 and Λ_0 for blr, we will get an equivalent result as the frequentist linear regression with weight decay penalty of $\alpha w^T w$. The explicit expression for w is $[X^T X + \alpha I]^{-1} X^T y$. The only explanation for the performance difference between lr/r/b and blr is the hyper-parameter selection of λ and α.





Problem 2

(b) Please elaborate on how to apply gradient descent algorithm to find the weights in your report.

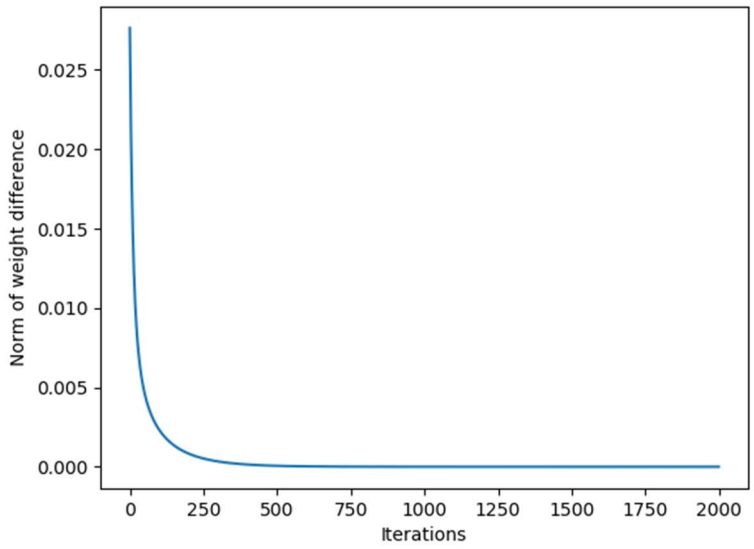
I use the following update rules:

$$w \leftarrow w - \eta \frac{\partial}{\partial w} J(w)$$

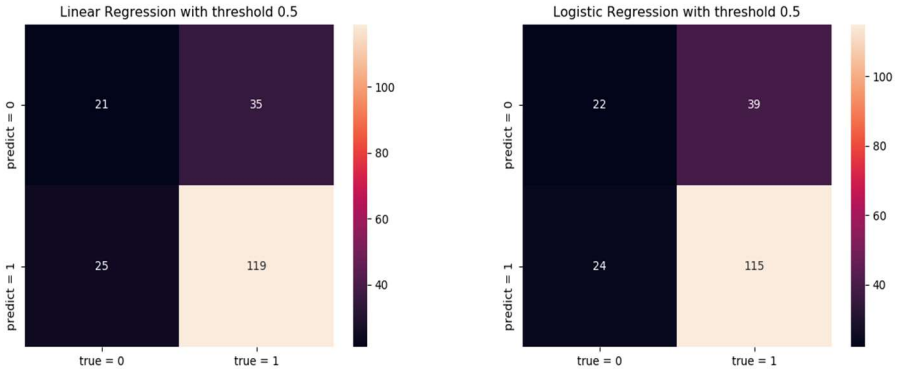
$$\frac{\partial}{\partial w} J(w) = \frac{1}{n} X^T (sigmoid(Xw) - y)$$

where n is the number of data and η is the step size, 0.1 for my choice.

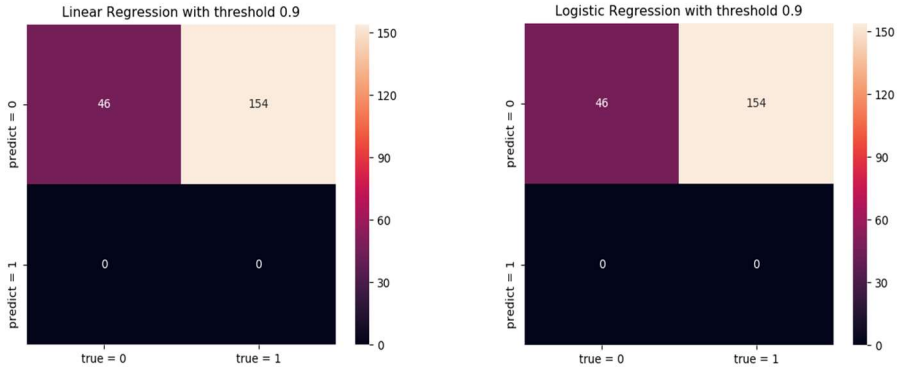
After 2000 iterations, we can see that the gradient has achieve zero.



(c) Please plot confusion matrices for both (a) and (b) as exemplified in Figure 2.



(d) Repeat (c) for the case where the threshold is set to 0.9.



(e) Please list possible reasons to low accuracy when switching threshold from 0.5 to 0.9 in your report. Please summarize the accuracies and precisions for those 6 cases (2 models × 3 thresholds) in a table.

Accuracy Table

	Threshold =0.1	Threshold =0.5	Threshold =0.9
Linear Regression	0.77	0.7	0.23
Logistic Regression	0.78	0.685	0.23

Precision Table

	Threshold =0.1	Threshold =0.5	Threshold =0.9
Linear Regression	0.7755102	0.82638889	No positive label predicted
Logistic Regression	0.78947368	0.82733813	No positive label predicted

The low accuracy when switching threshold from 0.5 to 0.9 because the threshold for the value to be greater than 0.9 is too demanding, hence no model predicts positive label in this case.

Reference:

Class Material

http://web.mit.edu/zoya/www/linearRegression.pdf

https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24