2019 Spring COM526000 Deep Learning - Homework 1

Machine Learning Basics: Regrsssion and Classification

Due: March 29, 2019

INSTRUCTIONS

1. In this homework, datasets from Student Performance Data Set from UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/datasets/student+performance`) are utilized to build various regression/classification models. Those two datasets were combined and shuffled into a single dataset. The last column, *cat*, represents the classes the students belong to.
2. **Please use *train.csv* to train/test your models and report regression/classification results generated from the hidden test set, *test_no_G3.csv*.** The following columns should be included as predictors: *school, sex, age, famsize, studytime, failures, activities, higher, internet, romantic, famrel, freetime, goout, Dalc, Walc, health, absences*, and you need to transform *binary* columns to one-hot encoding vectors. The target is *G3*.
3. **It is mandatory to build *ALL* functions with Python. Only *Numpy/Pandas* (for data preprocessing), *seaborn* (for plotting confusion matrix), and *matplotlib* (for plotting regression results) are allowed in this homework. *NO* machine learning platforms/packages are allowed such as *TensorFlow*, *scikit-learn*, *Keras*, etc.**
4. Name your source code that contains your *main* function as *hw1_StudentID.py* and your report as *hw1_StudentID.pdf*. You should provide your predictions for hidden test set following the format of example submissions (*StudentID_1.txt* and *StudentID_2.txt*). Please use *tabs* to separate IDs and predictions.
5. **You should write your own codes independently. Plagiarism is strictly prohibited.**

PROBLEMS

1. (40%) Linear Regression

(a) (10%) Split *train.csv* into training set (80%) and test set (20%). Both the training and test set should be normalized by subtracting the (column-wise) means of training set from them and then divided by the (column-wise) standard deviations of the training set. **Please elaborate on how you obtain your training and test sets in your report**. Notice that you should use identical training and test sets for (b) - (e).

(b) (5%) Implement a linear regression model *without* the bias term to predict *G3*. Use pseudo-inverse to obtain the weights. Record the root mean squared error (RMSE) of the test set.

(c) (5%) Regularization is often adopted to avoid over-fitting. Regularize your linear regression model by adding an additional term in your loss function, $\mathbf{J}(\mathbf{w})$:

$$\mathbf{J}(\mathbf{w}) = MSE_{train} + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}.$$

Implement a *regularized* linear regression model without the bias term where $\lambda = 1.0$. **Please describe how to find the optimal weights with maximum likelihood criterion in your report.** Record the RMSE of the test set.

(d) (5%) Repeat (c) but *include* the bias term in your model.

(e) (5%) Follow *Example: Bayesian Linear Regression* in the textbook (Chapter 5) and implement a Bayesian linear regression model *with* the bias term. Let $\mu_0 = \mathbf{0}$ and $\mathbf{\Lambda}_0 = \frac{1}{\alpha}\mathbf{I}$ in (5.78) where $\alpha = 1.0$. Use the mean of the posterior as weights for your model. Record the RMSE of the test set.

(f) (10%) Plot the ground truth (real *G3*) versus all predicted values generated by models (b) - (e) as examplified in Figure 1. **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).**

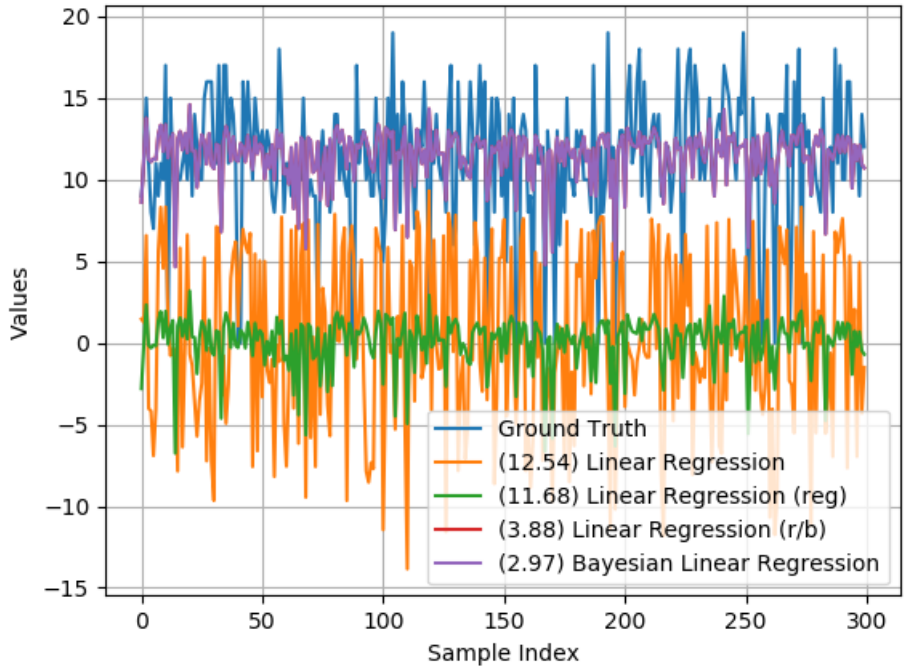

Figure 1: Regression result comparison.

2. (40%) Classification

(a) (5%) Create a new column to indicate whether *G3* is greater or equal to 10 (1 if this event is true; 0 if this event is not true) to serve as the labels for classification. Implement a linear regression model *with* regularization ($\lambda = 1.0$) and the bias term to predict the labels. Record the classification results when thresholds are set to 0.1, 0.5, and 0.9. Note that samples with model activations greater than the threshold are classified as class 1, and class 0 otherwise.

(b) (15%) Repeat (a) but use logistic regression. **Please elaborate on how to apply gradient descent algorithm to find the weights in your report**.

(c) (2%) For the case where the threshold is set to 0.5, **please plot confusion matrices** for both (a) and (b) as examplified in Figure 2.

(d) (3%) Repeat (c) for the case where the threshold is set to 0.9.

(e) (15%) From (c) and (d), **please list possible reasons to low accuracy when switching thresold from 0.5 to 0.9 in your report**. **Please summarize the accuracies and**

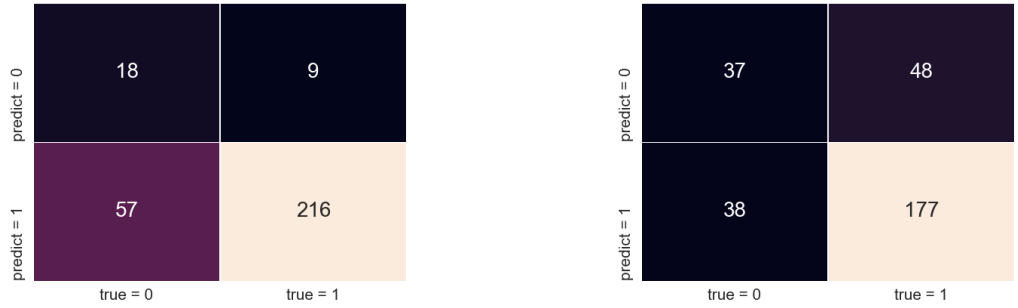

Figure 2: Example confusion matrix.

precisons for those 6 cases (2 models $\times$ 3 thresholds) in a table. Notice that precision represents the proportion of correctly predicted 1's in all 1's predicted.

3. (20%) Hidden Test Set

   (a) (10%) Apply the model from 1. (d) to *test_no_G3.csv* and save your results as *StudentID_1.txt*. You are allowed to tune α.

   (b) (10%) Apply the model from 2. (b) to *test_no_G3.csv* and save your results as *StudentID_2.txt*. You are allowed to tune hyper-parameters.