CMPUT 466 Project Report

Alex Mak
ID: 1584710

April 25th, 2023

1. Introduction/ task background

From the second coding assignment, I was able to learn that linear regression performs poorly compared to logistic regression in classification tasks. I learned this idea when I observed that the training accuracies of linear regression classifiers were noticeably lower than the training accuracies of logistic regression classifiers.

To avoid overfitting and reduce variance in linear models, the concept of regularization is introduced with the purpose of restricting the capacity of the linear models' hypothesis class so they will have a lower variance and a higher bias. Although regularization might harm the model's training performance, it also has the possibility to improve the model's validation/test performance.

Altogether, the task I would like to test is whether linear regression with regularization can perform as good as logistic regression in terms of the test performances on binary classification tasks. I will test this task through a training-validation-test infrastructure to train, validate and test different machine learning linear models.

2. Problem Formulation

In order to test the task I mentioned above, I intend to formulate the problem in the following way:

Input and Output:
The designed input of the task would be the different machine learning linear models. They include linear regression and logistic regression in which they will serve as baseline of this task, as well as three linear regression models with regularization. They are lasso regression (linear regression with l1 penalty), ridge regression (linear regression with l2 penalty), and elastic net regression (linear regression with both l1 and l2 penalties). These 3 models will be compared with each other, as well as the other 2 baseline models in order to answer the task I assigned above.

They will be compared based on their training accuracies on binary classification tasks, which is also the designed output of this problem.

The training accuracies on binary classification tasks will be computed by how well the model fits the dataset itself. Specifically, it counts the proportion where the set of labels predicted for a sample from each linear model matches the corresponding set of labels in the dataset.

Dataset:
The dataset used to solve this problem is the Breast Cancer Wisconsin (Diagnostic) Data Set. This dataset is downloaded from the UCI Machine Learning Repository Website

(https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)), this dataset can be also found in Kaggle, or in scikit-learn because it is one of the built-in datasets.This dataset consist of different characteristics of a tumor (i.e radius, perimeter, area etc), and the diagnosis information which it classifies the tumor to either benign (B), or malignant (M). Altogether, this dataset is ideal for binary classification since the diagnosis label is classified based on 2 outcomes.

Within this dataset, there are a total of 569 samples. Specifically, 357 of them are classified as benign and 212 of them are classified as malignant.

3. **Approaches and baselines**

For this task, I plan to tune the regularization strength of each linear model. Therefore, I decided to tune the hyperparameters of learning rate (ɑ) because it directly affects the regularization strength of a model. Specifically, I will tune ɑ for the models of ridge regression and elastic net regression; and C (constant that is the inverse of regularization strength) for the models of logistic regression and lasso regression.

The reason I tuned ɑ for some of my approaches, and C for other models is because due to the constraint from Scikit learn's API, some models I implement (logistic regression and lasso regression) do not have ɑ as the hyperparameter to be tuned. Instead the hyperparameter C is available which is the inverse of regularization strength.

Nevertheless, all tuned models will be tuned in the same way as they are validated through a 5-fold cross validation in which the dictionary of tuned hyperparameter's values will also be the same. This means that every model that has been tuned will choose the best parameter value based on the same parameters' value list.

To resolve the issue of tuning models through 2 different hyperparameter (ɑ and C), the values in the parameters' value list will be inverted for models tuned by C. This is done to counteract the relationship between ɑ and C where $C = \frac{1}{\alpha}$

Additionally, the reason the linear regression model will not be tuned with ɑ is because linear regression itself does not have ɑ as ɑ is used in applying regularization.

4. **Codebase**

Aside from this project report, there are also 2 other files for this project. One of them is data.csv, it is the file of the dataset in .csv format,

The other one is code.py, which is the python code used to take different linear models as an input and return their training accuracies in a binary classification task as an output.

To run code.py completely, please install the following libraries:

- Numpy: (between versions 1.16.5 and 1.23.0 would be ideal)
    - https://www.binarystudy.com/2022/12/how-to-install-specific-version-of-Numpy-with-PIP.html
- Pandas
    - https://pandas.pydata.org/docs/getting_started/install.html
- scikit-learn
    - https://scikit-learn.org/stable/install.html

Then run the command “python code.py”. (Please ensure that the data.csv and code.py is within the same directory otherwise the code file cannot load the dataset properly)

Code Description and Summary

1. The code.py code will first read the dataset (data.csv), then it will separate the data into X and Y. X, being the characteristics that caused the tumor to be benign or malignant; Y, being the categorical variable that shows whether the tumor is actually benign or malignant.
2. Encode the categorical data (Y) into numbers (0 and 1) in order to perform binary classification.
3. Split the dataset into 3 portions: training (80%), validation (10%), testing (10%)
4. Standardize the features
5. For each model (linear, logistic, lasso, ridge, and elastic net regression)
    a. Train the model using the training dataset
    b. Perform hyperparameter tuning to obtain the best α/C that will compute the highest accuracy.
        i. Note: this step will be computed for every baseline and approach models except the baseline linear regression
    c. Train the model using the training dataset again, but with the optimal α/C. Obtain the testing accuracies after tuning from testing the trained model into the validation and testing datasets

5. Evaluation metric

The measure of success comes from the testing accuracies that are computed on the testing part dataset after each model (except linear regression) has finished hyperparameter tuning in the validation part of the dataset.

Precisely, the 3 test accuracies from the models in my 3 approaches (lasso, ridge, elastic net regression) will first compare with the testing accuracy after hyperparameter training for the logistic regression to answer the main question of this task about whether linear regression with regularization can perform as good as logistic regression in terms of the test performances on classification tasks.

Afterwards, the 3 test accuracies in my 3 approaches will be compared to each other to determine which one of the 3 yields the best performance, which is determined by the highest test accuracies.

The focus of my task is to determine whether there are any linear regression models with regularization that can match the test performance as the logistic regression model. Additionally if so, how many out of the 3 approaches can match the test performance.

As a result, the measure of success is an approximation since the test accuracies are being compared appropriately. By matching the test performance, this means that the test accuracy from a linear regression model should approximately share the same accuracy percentage as the test accuracy from a logistic regression model, or even higher if possible.

6. Results

The result of the 3 approaches after hyperparameter tuning are:

- Test accuracy for lasso regression: 0.9473684210526315
- Test accuracy for ridge regression: 0.9122807017543859
- Test accuracy for elastic net regression: 0.9298245614035088

The result from the baseline:

- Test accuracy for linear regression: 0.8947368421052632
- Test accuracy for logistic regression: 0.9649122807017544

Notes: the results are slightly varied for every computation of the code itself, but the overall inferences and assumptions can still be made.

In comparison to the baselines, the test accuracies from 3 approaches are all higher than the one with linear regression, meaning that the regularization in linear regression models does improve their test performance. With the improvement in test performance, the test accuracies for regularized linear regression models are much closer to the test accuracies for logistic regression models.

Within the 3 approaches, lasso regression has the highest test accuracy making it closest to logistic regression in terms of test performance, with elastic net regression following, then finally ridge regression.

However none of the 3 linear regression models from the approaches can match the accuracy from the logistic regression model (0.9649122807017544).

In conclusion, despite the fact that regularization brings noticeable improvement in terms of test performance for linear regression models which makes them better classifiers than the linear regression models without any regularization, linear regression with regularization are still unable to perform as good as logistic regression in terms of the test performances on binary classification tasks.

This proves that logistic regression models are still desirable over linear regression models in performing binary classification tasks.