ENGINEERING DIVISION | NYU ABU DHABI

ENGR-UH 3332
Applied Machine Learning

Mini Project 1 – Linear Regression

Due Date: Refer to Brightspace

Introduction

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in many applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Linear Regression

Dataset

In this project, we are going to use "Boston house prices dataset" from Scikit Learn. The Boston house-price data has been used in many machine learning papers that address regression problems.
Boston house prices dataset has 506 instances and for each instance, it has 13 attributes and one target value.

- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940
- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in $1000's

Requirements

1. Explore and import **Boston house prices dataset:**
   a. Explore the dataset by using the Scikit Learn library and Numpy.

2. In this problem you will experiment with a linear regression problem based on real-world data. The data is from the Boston Housing dataset in scikit-learn. Your task isto estimate the price of a house in Boston using 13 attributes.

3. Fit a linear regression model using the closed-form solution presented in class. Use k-fold cross-validation to estimate the performance of this model. Print the average of your recorded scores for both the test set and training set.
   a. Only Numpy library is allowed. (you may use external API to extract samples for k folds)

4. Fit a ridge regression model using the closed solution. Use k-fold cross-validation to find the best $\lambda \in \left[10^1, 10^{1.5}, 10^2, 10^{2.5}, \ldots, 10^7\right]$.
   a. Use the Numpy function: np.logspace(1, 7, num=13) to get the different values for λ.

5. For the best λ you found, use k-fold cross-validation to estimate the performance of this model with this λ . Print the average of your recorded scores for both the test set and training set.

6. Repeat the previous exercise, but this time, by creating a polynomial transformation of degree 2 on the features of the dataset.
   a. Using PolynomialFeatures(degree=2) in sklearn library

7. Repeat Multivariate Linear Regression using the Gradient Descent method.

8. Implement Lasso Regression (Text book Chapter 4)
   a. Cost function of Lasso Regression:

$$J(\theta) = MSE(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

9. Implement Elastic Net (Text book Chapter 4)
   a. Cost Function of Elastic Net:

$$J(\theta) = MSE(\theta) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \theta^2$$

10. If you are given a choice of predicting future housing prices using one of the models you have learned above (those optimized with gradient descent), which one would you choose and why? State the parameters of that model.

Deliverables

A zip file containing the following:

1. a working project (source code, makefiles if needed, etc)
2. a report for the detailed description of the project
    a. Instructions on how to run your project
    b. Answers to the programming questions.

Before submitting your project, please make sure to test your program on the given dataset.

Notes

You may discuss the general concepts in this project with other students, but you must implement the program on your own. ***No sharing of code or report is allowed.*** *Violation of this policy can result in a grade penalty.*

Late submission is acceptable with the following penalty policy:
10 points deduction for every day after the deadline