Machine Learning

Assignment 2 - Linear Regression
Summer 2023

Introduction

In this assignment you will explore closed-form (direct) linear regression on a dataset along with cross validation.

As with all homeworks, you cannot use any functions that are against the "spirit" of the assignment. For this assignment that would mean any linear regression functions. You *may* use statistical and linear algebra functions to do things like:

- mean
- std
- cov
- inverse
- matrix multiplication
- transpose
- etc...

Grading

Although all assignments will be weighed equally in computing your homework grade, below is the grading rubric we will use for this assignment:

Part 1 (Theory)	10pts
Part 2 (Closed-form LR)	50pts
Part 3 (Cross Validation)	40pts
TOTAL	100

Table 1: Grading Rubric

Datasets

Medical Cost Personal Dataset This dataset consists of data for 1338 people in a CSV file. This data for each person includes:

1. age
2. sex
3. bmi
4. children
5. smoker
6. region
7. charges

For more information, see https://www.kaggle.com/mirichoi0218/insurance

1 Theory

1. Consider the following supervised *training* dataset:

$$X = \begin{bmatrix} -2 \\ -5 \\ -3 \\ 0 \\ -8 \\ -2 \\ 1 \\ 5 \\ -1 \\ 6 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ -4 \\ 1 \\ 3 \\ 11 \\ 5 \\ 0 \\ -1 \\ -3 \\ 1 \end{bmatrix}$$

   (a) Compute the coefficients for closed-form linear regression using least squares estimate (LSE). Show your work and remember to add a bias feature. Since we have only one feature, there is no need to zscore it (6pts).

   $X\ Train = 1.02$
   $Y\ Train = -0.412$

   (b) Using your learned model in the previous part, what are your predictions, $\hat{Y}$, for the training data (2pts)?

   (c) What is the RMSE and SMAPE for this training set based on the model you learned in the previous part (2pts)?

2 Closed Form Linear Regression

In this section you'll create simple linear regression models using the dataset mentioned in the Datasets section. Use the first six columns as the features (age, sex, bmi, children, smoker, region), and the final column as the value to predict (charges). Note that the features contain a mixture of continuous valued information, binary information, and categorical information. It will be up to you to decide how to do any pre-processing of the features!

First randomize (shuffle) the rows of your data and then split it into two subsets: 2/3 for training, 1/3 for validation. Next train your model using the training data, and evaluate it for the training data, and for the validation data.

Implementation Details

1. So that you have reproducible results, we suggest that seed the random number generate prior to using it. In particular, you might want to seed it with a value of zero so that you can compare your numeric results with others.
2. **IMPORTANT** If you notice there's issues in computing the inverse of $X^T X$ due to sparcity, you maybe one to try one of the following:
   - Using the *pseudo-inverse* instead of the regular inverse. This can be more stable and accurate.
   - Adding some "noise" (i.e. very small values) to the binary features you made out of the enumerated features.

NOTE: Since your target values are relatively large, so too will your RMSE.

In your report you will need:

1. The root mean squared errors (RMSE) and symmetric mean absolute percent error (SMAPE) for the training **and** validation sets.
2. A list of any pre-processing of the dataset that you performed.

Training RMSE: 5911.310
Validation RMSE: 6345.477
Training SMAPE: 37.2396
Validation SMAPE: 38.1098

3 Cross-Validation

Cross-Validation is a technique used to use more data for training a system while maintaining a reliable validation score.

In this section you will do S-Folds Cross-Validation for a few different values of S. For each run you will divide your data up into S parts (folds) and build S different models using S-folds cross-validation and evaluate via root mean squared error. In addition, to observe the affect of system variance, we will repeat these experiments several times (shuffling the data each time prior to creating the folds). We will again be doing our experiment on the aforementioned dataset.

Write a script that:

1. Reads in the data, ignoring the first row (header) and first column (index).
2. 20 times does the following:
   (a) Seeds the random number generator to the current run (out of 20).
   (b) Shuffles the rows of the data
   (c) Creates S folds.
   (d) For $i = 1$ to S
      i. Select fold i as your validation data and the remaining $(S - 1)$ folds as your training data.
      ii. Train a linear regression model using the direct solution.
      iii. Compute the squared error for each sample in the current validation fold
   (e) You should now have N squared errors. Compute the RMSE for these.
3. You should now have 20 RMSE values. Compute the mean and standard deviation of these. The former should give us a better "overall" mean, whereas the latter should give us feel for the variance of the models that were created.

Implementation Details

1. Don't forget to add a bias feature!

In your report you will need:

1. The average and standard deviation of the root mean squared validation error for $S = 3$ over the 20 different runs.
   Avg. RMSE for 3-fold cross-validation over 20 runs: 6080.127
   Avg. SMAPE for 3-fold cross-validation over 20 runs: 38.166
2. The average and standard deviation of the root mean squared validation error for $S = 223$ over the 20 different runs.
   Avg. RMSE for 223-fold cross-validation over 20 runs: 5625.89
   Avg. SMAPE for 223-fold cross-validation over 20 runs: 38.10

3. The average and standard deviation of the root mean squared validation error for $S = N$ (where N is the number of samples) over 20 different runs. This is basically *leave-one-out* cross-validation.
   Avg. RMSE for 1338-fold cross-validation over 20 runs: 4199.78
   Avg. SMAPE for 1338-fold cross-validation over 20 runs: 38.113

Submission

For your submission, upload to Blackboard a single zip file with no spaces in the file or directory names and contains:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1: Your solutions to the theory question.
2. Part 2: Requested statistics and pre-processing decisions.
3. Part 3: Requested statistics and pre-processing decisions.