Machine Learning

Lab 3 - Linear Regression
Fall 2024

Introduction

In this lab you will explore closed-form (direct) linear regression.

As with all labs, you cannot use any functions that are against the "spirit" of the lab. For this lab 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...

Your task will be to write a *single script* such so that we can run it in the command line and it output (and/or displays) the requested information/figures.

Grading

- +2pt Can run script properly.
- +2pts Parses dataset correctly.
- +1pt Adds in bias feature.
- +2pts Creates training and validation sets correctly.
- +1pt Generates some results.
- +2pts Generates correct statistics (within reasonable range. RMSE ≈ 6000, SMAPE $\approx 18\%$).

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 Closed Form Linear Regression

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. Don't forget to add a bias feature!
2. So that you have reproducible results, we suggest that you 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.
3. **IMPORTANT** If you notice there's issues in computing the inverse of $X^T X$ due to sparcity, you might want 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.

Print to the command prompt the following information:

1. The root mean squared errors (RMSE) and symmetric mean absolute percent error (SMAPE) for the training **and** validation sets. You might find different equations from SMAPE online, but use the one provided in the slides.