CS 383 - Machine Learning

Assignment 3 - Linear Regression

Introduction

In this assignment you will explore gradient descent and perform closed-form linear regression on a dataset.

As with all homeworks, you cannot use any functions that are against the “spirit” of the assignment. For this assignment that would mean an 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)	20pts
Part 2 (Gradient Descent)	30pts
Part 3 (Closed-form LR)	50pts
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).

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

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

2. For the function $J = (x_1 w_1 - 5x_2 w_2 - 2)^2$, where $w = [w_1, w_2]$ are our weights to learn:

   (a) What are the partial gradients, $\frac{\partial J}{\partial w_1}$ and $\frac{\partial J}{\partial w_2}$? Show work to support your answer (6pts).

   (b) That are the value of the partial gradients given current values of $w = [0, 0], x = [1, 1]$ (4pts)?

2 Gradient Descent

In this section we want to visualize the gradient descent process on the function

$$J = (x_1 w_1 - 5 x_2 w_2 - 2)^2$$

You should have already derived (pun?) the gradient of this function in the theory section. To bootstrap the process, initialize $w_1 = 0$ and $w_2 = 0$. In addition, we'll assume only a single observation: $x = [1, 1]$.

Write a program to perform gradient descent on this function, terminating when the change J from one epoch to another is less that 2^{-32}. In addition, we'll use a learning rate of $\eta = 0.01$. You'll want to keep track of the values of J, w_1, and w_2 during learning in order to generate the plots mentioned below:

In your report you will need

1. Plot epoch vs J as a line graph.
2. Create a 3D line plot of w_1 vs w_2 vs J.
3. Report your final values of w_1, w_2 and J, in addition to the number of epochs needed to reach your termination criteria.

3 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.

First randomize (shuffle) the rows of your data and then split it into two subsets: 2/3 for training, 1/3 for validation.
Now let's train **four** models using *closed-form linear regression.* As you'll see below we will be exploring the effects of pre-processing and incorporation of bias features.

1. The first system will change categorical features to enumerated ones (but will **NOT** convert the *region* feature into a set of binary features. In addition it will **NOT** have a bias feature.

2. The second will do the same pre-processing as above, but will now include a bias feature.

3. The third **WILL** convert the *region* feature into a set of four binary features, but will **NOT** have a bias feature.

4. And finally we'll train a system using the pre-processing in the previous part, but **WILL** include a bias term.

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. The closed-form of linear regression doesn't benefit from zscorring your data, so it's up to you whether you want to or not.

3. **IMPORTANT** Converting enumerated features to a set of binary features introduces *sparsity* to our matrix X. Since the closed-form solution requires computing the inverse of X^TX, this sparsity, combined with adding a feature of all ones (the bias feature), can cause issues in finding the inverse of X^TX. To overcome this you can try:

   - 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.

In your report you will need:

1. The root mean squared errors for the training **and** validation sets for each of your four models.

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 questions.
2. Part 2:
    (a) Your plot of epoch vs J.
    (b) Your 3D plot of w_1 vs w_2, vs J.
    (c) The final learned values of w_1, w_2, and J, and the number of epochs required to get there.
3. Part 3: The RMSE for the training and validation sets for your four models.