CS 383 - Machine Learning

Assignment 4 - Gradient Descent
Winter 2017

Introduction

In this assignment you will perform linear regression via gradient descent.

You may **not** use any function from the Matlab ML library in your code. And as always your code should work on any dataset that has the same general form as the provided one.

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 (GD LR)	30pts
Report	5pts
TOTAL	35

Table 1: Grading Rubric

Datasets

Fish Length Dataset (x06Simple.csv) This dataset consists of 44 rows of data each of the form:

1. Index
2. Age (days)
3. Temperature of Water (degrees Celsius)
4. Length of Fish

The first row of the data contains header information.

Data obtained from: http://people.sc.fsu.edu/ jburkardt/datasets/regression/regression.html

1 Gradient Descent

Download the dataset *x06Simple.csv* from Blackboard. This dataset has header information in its first row and then all subsequent rows are in the format:

$$ROWId, x_{i,1}, x_{i,2}, y_i$$

Your code should work on any CSV data set that has the first column be header information, the first column be some integer index, then D columns of real-valued features, and then ending with a target value.

As discussed in class Gradient Descent (Ascent) is a general algorithm that allows us to converge on local minima (maxima) when a closed-form solution is not available or is not feasible to compute.

In this assignment you are going to implement a gradient descent algorithm to find the parameters for linear regression on the same data used for the previous sections. You may **NOT** use any function for a ML library to do this for you.

Implementation Details

1. Seed the random number generator prior to your algorithm.
2. Don't forget to add in the offset feature!
3. Initialize the parameters of θ using random values in the range [-1, 1]
4. Do **batch** gradient descent
5. Terminate when absolute value of the percent change in the RMSE on the **training** data is less than the Matlab defined *eps* or after $1,000,000$ iterations have passed (whichever occurs first).
6. Use a learning rate $\eta = 0.01$.
7. Make sure that you code can work for an arbitrary number of observations and an arbitrary number of features.

Write a script that:

1. Reads in the data, ignoring the first row (header) and first column (index).
2. Randomizes the data
3. Selects the first 2/3 (round up) of the data for training and the remaining for testing
4. Standardizes the data (except for the last column of course) base on the training data
5. While the termination criteria (mentioned above in the implementation details) hasn't been met
   (a) Compute the RMSE of the *training* data
   (b) While we can't let the testing set affect our training process, also compute the RMSE of the testing error at each iteration of the algorithm (it'll be interesting to see).

(c) Update each parameter using *batch* gradient descent

6. Compute the RMSE of the testing data.

What you will need for your report

1. Final model
2. A graph of the RMSE if the *training* and *testing* sets as a function of the iteration
3. The final RMSE *testing* error.

Your graph should look similar to Figure 1.

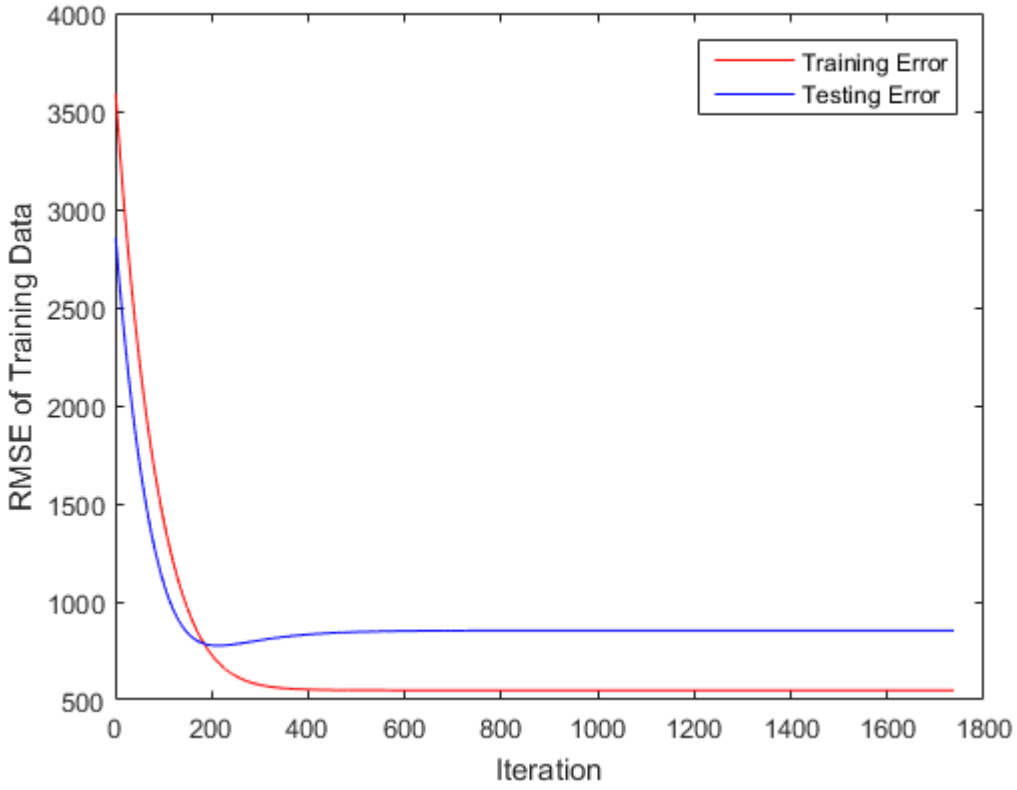

Figure 1: Gradient Descent Progress

RMSE:	853.38

Table 2: Gradient Descent Regression Evaluation

Submission

For your submission, upload to Blackboard a single zip file containing:

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:
   (a) Final Model
   (b) RMSE
   (c) Plot of RMSE for Training and Testing Data vs Gradient Descent iteration number