Brandon Langley

CPSC 4820

Linear Regression and Gradient Descent

Project 2: Report

September 30, 2019

Problem Description

An Entrepreneur has an idea for an app which will assist student decision making regarding drinking beer and studying. They have collected data from 300 students which consists of the time spent studying per week (in minutes), the volume of beer they consume per week (in ounces), as well as each students grade point average for the semester. Given the data set develop a linear regression model of the form $y=w_0+w_1x_1+w_2x_2$ and train the algorithm using gradient descent. The goal is to have the model be implemented in an app so that students can enter the amount of beer they consume and the time they spend studying and the app will predict the student's GPA.

Data Description

The data is given in a txt file, the fist line of which represents the count of the number of records in the file, the remainder of the file contains the data collected from each student. Each line of the file contains the record for one individual student consisting of 3-tab separated values corresponding to the time spent studying, oz of beer, and GPA, in that order.

Developing the Algorithm

Initially the data was read in and immediately randomized, it was then divided into a Training Set (70% or 210 record) and a Test Set (30% or 90 records). Then the linear regression model was trained using the training set. The initial values passed to the model were as follows; weights: $W_0=0$, $W_1=0$, $W_2=0$ and learning rate alpha=0.1. However, immediately after attempting to run the algorithm with these values it became apparent that this value for alpha was much too large, after testing a variety of alpha values alpha=0.00000001 was found to be more appropriate. With this value for alpha the initial value for J was 223.48994864 which is very high indicating the weights are still very far off. Next the algorithm was allowed to iterate and train the

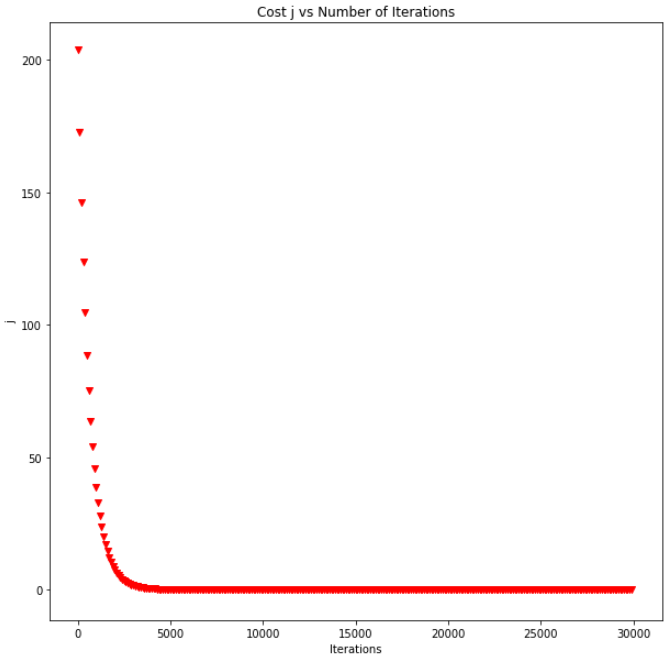

Figure 1: Graph Comparing J vs Number of iterations

weights until J(w_0,w_1,w_2) converged. After allowing the algorithm to iterate 50,000 times it was found that J converged around .008 after approximately 26,000 iterations. After 26,000 iterations J was equal to 0.00899617, it took another 20,000 iteration for J to drop another .001. For this reason, for the final training run J was taken to converge after 30,000 iterations. This can be seen in Figure 1. Additionally, to illustrate the convergence of J Figure 2 shows only the final 5,000 iterations. After the final training run the algorithm gave the following weights:
W_0=2.19845811e-05,
W_1=5.20852296e-02,
W_2=2.59243661e-03.

Results

Using these weight values to calculate the cost J on the Test Set gave a result of J=0.00266234

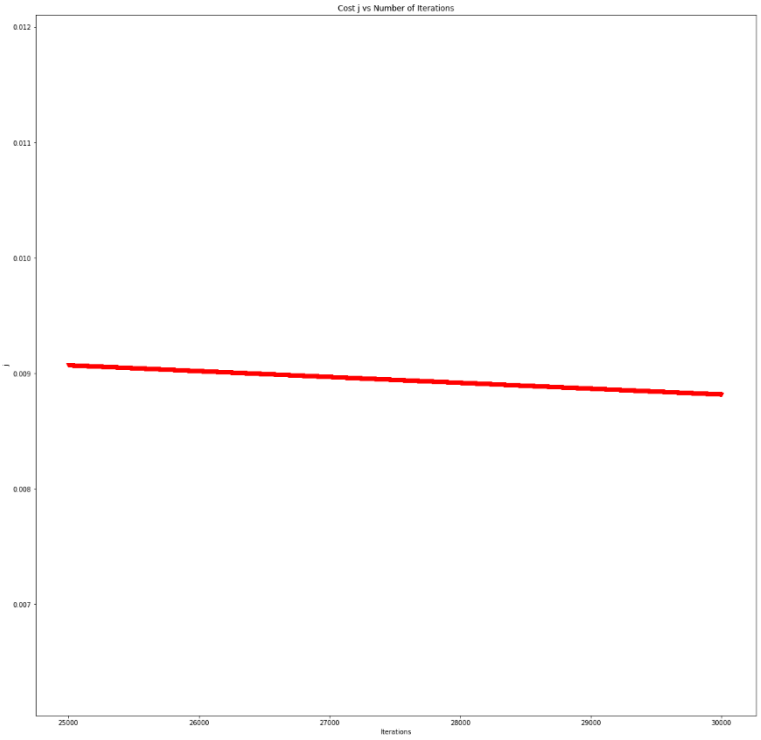

Figure 2: Graph of J values for the final 5,000 iterations