Assignment 2: Linear Regression with One Variable

In this assignment, you will fit the linear regression parameters⌘to our dataset using gradient descent.

The objective of linear regression is to minimize the cost function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

where the hypothesis h⌘(x) is given by the linear model

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

Recall that the parameters of your model are the θ_j values. These are the values you will adjust to minimize cost J(θ). One way to do this is to use the batch gradient descent algorithm. In batch gradient descent, each iteration performs the update

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \qquad \text{(simultaneously update } \theta_j \text{ for all } j\text{).}$$

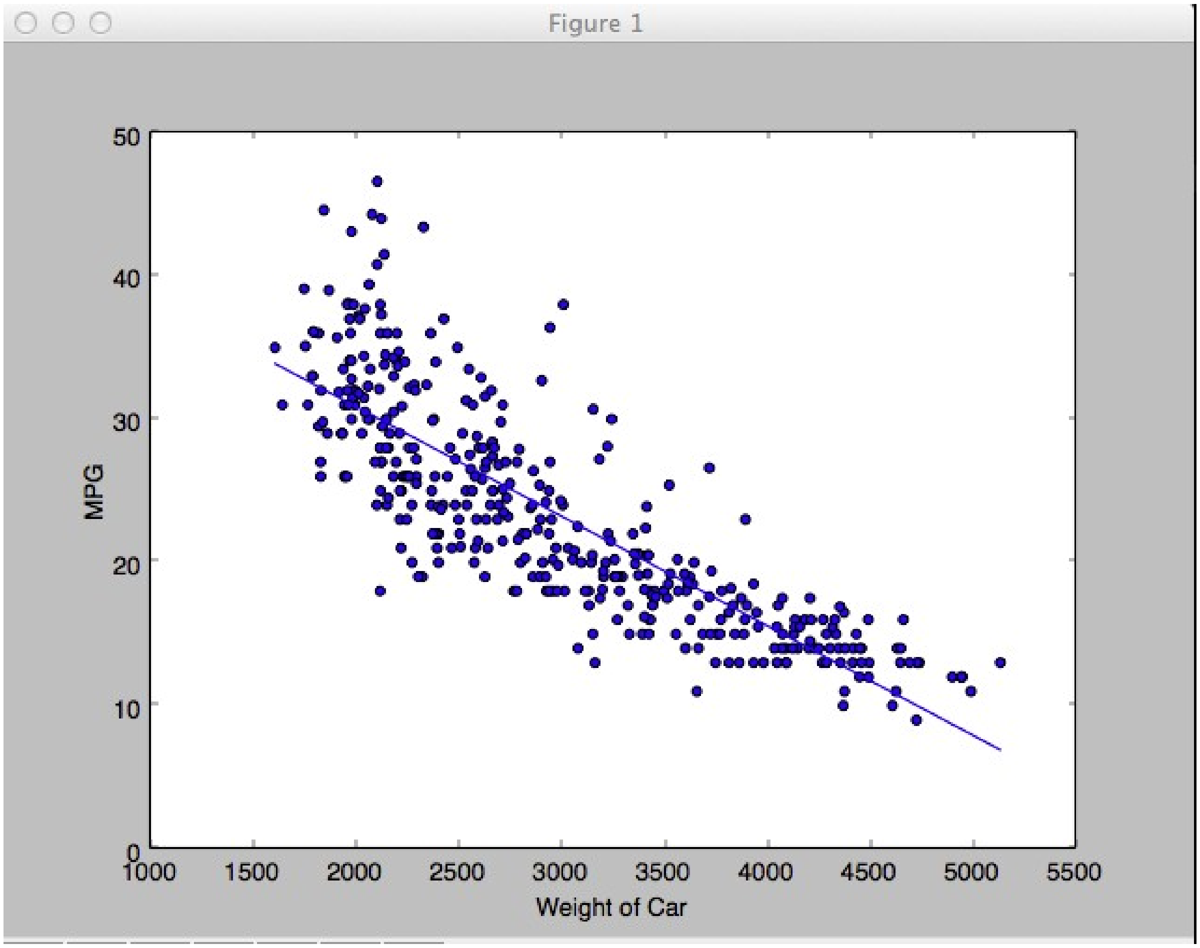
With each step of gradient descent, your parameters θ_j come closer to the optimal values that will achieve the lowest cost J(θ).

What are you going to do: Implement the functions in cost_function.py. This will involve computing the cost function, prediction function, and gradient descent for one variable.

I have included a data file comparing MPG with various car features (weight, horsepower, year, model, etc.). I have already implemented code that loads data from this file into ex1_main. This Python file calls the functions you will implement in cost_function.py at the appropriate times. It will print to your console:

1. The value of the cost function before gradient descent is performed. My value for this was 119652.7
2. The values of theta1 and theta1 in an array. My values were 23.44490618, -6.48710167.
3. The cost after gradient descent. MY value was 20157.6646036.

Then it will display a plot of your data with the line that fits your theta parameters. It should look like the one below:



Implement the functions so that your values are close to mine for full credit (10pts).

Extra Credit (5pts). Plot the value of J(θ) for each iteration number in the gradient descent (you will need to store these in a separate array. You should see the cost value go down until they approach an asymptote.

Implementation note: all the files (Python files, data files) need to be in the same directory for the code to work!

Question: Is a linear model the best for this data? Think about it.