National University of San Agustin
System Engineer School

AI - Lab 02 - Gradient descent

Christian E. Portugal-Zambrano

April 08, 2019

1 Introduction

Until now we just applied an *iterative* and *brute force* manner to find the *optimal* values for θ_0, θ_1. There exists another ways to found these values without using high computational cost, the intuition is based on the change of the predicted thetas after each iteration, we can avoid the selection of a range of values for θ to search using a derivate, so we need to remember what a derivate is? and how can help us to make better values of θ, then we will be presenting an elegant method to perform the same task with less variables, this is know as the normal equation.

1.1 Objectives

- Understand the gradient descent as a better method to find the optimal θ.
- Understand the fundamentals of partial derivates as a better approach and its applications into machine learning.
- Understand how to apply gradient descent and normal equation to our linear models.

1.2 Pre-requisites

You need some linear algebra and calculus for this assignment.

2 Gradient Descent

So we have our hypothesis function and we have a way of measuring how well it fits into the data. Now we need to estimate the parameters in the hypothesis function. That's where gradient descent comes in. Imagine that we graph our hypothesis function based on its fields θ_0 and θ_1 (actually we are graphing the cost function as a function of the parameter estimates). We are not graphing x and y itself, but the parameter range of our hypothesis function and the cost resulting from selecting a particular set of parameters. We put θ_0 on the x axis and θ_1 on the y axis, with the cost function on the vertical z axis. The points on our graph will be the result of the cost function using our hypothesis with those specific theta parameters. The Figure 2.1 represents our intuition: We will know that we have succeeded when our cost function is at the very bottom of the pits in our graph, i.e. when its value is the minimum. The red arrows show the minimum points in the graph.

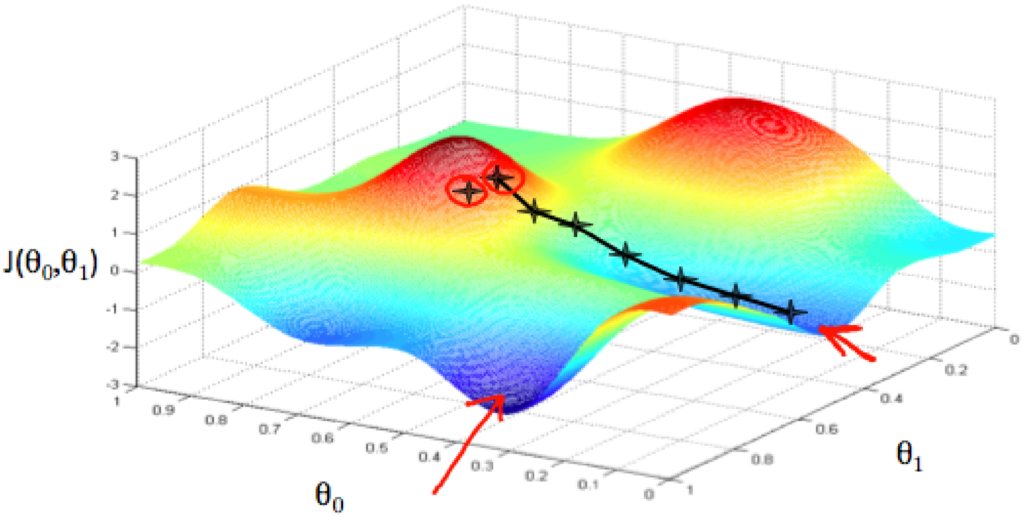


Figure 2.1: Cost function with a path of thetas selected.(Figure has been taken from Machine Learning Course from Andrew NG for Coursera, you can visit at Coursera-Machine Learning)

The way we do this is by taking the derivative (the tangential line to a function) of our cost function. The slope of the tangent is the derivative at that point and it will give us a direction to move towards. We make steps down the cost function in the direction with the steepest descent.

The size of each step is determined by the parameter α, which is called the learning rate, remember from the iterative process? the step we choose to find the next theta is the most closed representation of our learning rate.For example, the distance between each **black star** in the graph above represents a step determined by our parameter α.

A smaller α would result in a smaller step and a larger α results in a larger step. The direction in which the step is taken is determined by the partial derivative of $J(\theta_0, \theta_1)$. Depending on where on starts on the graph, one could end up at different points. The Figure 2.1 shows us two different starting points that end up in two different places. To formalize our gradient descent intuition we present the algorithm:

$$\text{repeat until convergence: } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$
$$\}$$

where $j = 0, 1$ represents the feature index number.

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived. We can substitute our actual cost function and our actual hypothesis function and modify the equation to:

$$\begin{aligned} &\text{repeat until convergence: } \{ \\ &\qquad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) \\ &\qquad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i) \\ &\} \end{aligned}$$

where m is the size of the training set, θ_0 a constant that will be changing simultaneously with θ_1 and x_i, y_i are values of the given training set (data).

Note that we have separated out the two cases for θ_j into separate equations for θ_0 and θ_1 and that for θ_1 we are multiplying x_i at the end due to the derivative. The following is a derivation of $\frac{\partial}{\partial \theta_j} J(\theta)$ for a single example :

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 \\ &= 2.\frac{1}{2} (h_\theta(x) - y) \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\ &= (h_\theta(x) - y) \frac{\partial}{\partial \theta_j} (\sum_{i=0}^{m} \theta_i x_i - y) \\ &= (h_\theta(x) - y) x_j \end{aligned} \tag{2.1}$$

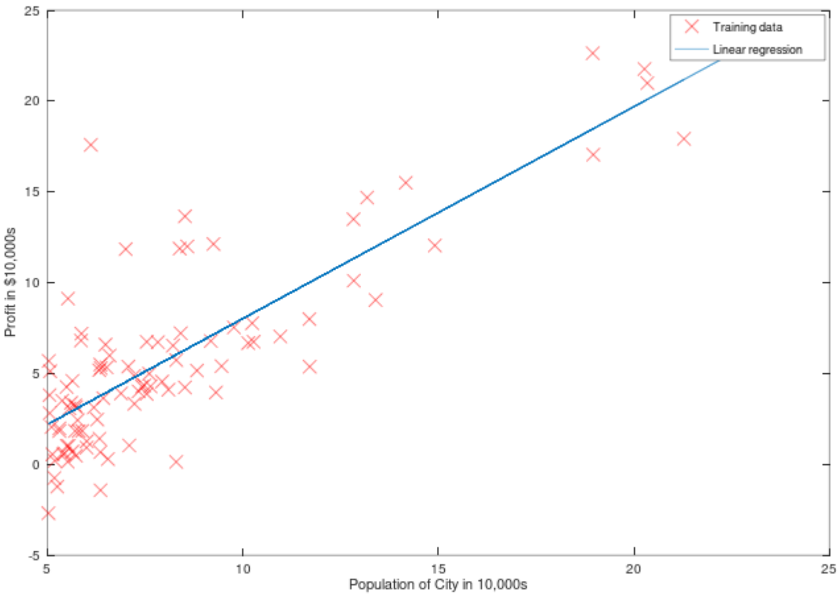
The point of all this is that if we start with a guess for our hypothesis and then repeatedly apply these gradient descent equations, our hypothesis will become more and more accurate.

So, this is simply gradient descent on the original cost function J. This method looks at every example in the entire training set on every step, and is called batch gradient descent. Note that, while gradient descent can be susceptible to local minima in general, the optimization problem we have posed here for linear regression has only one global, and no other local, optima; thus gradient descent always converges (assuming the learning rate α is not too large) to the global minimum.

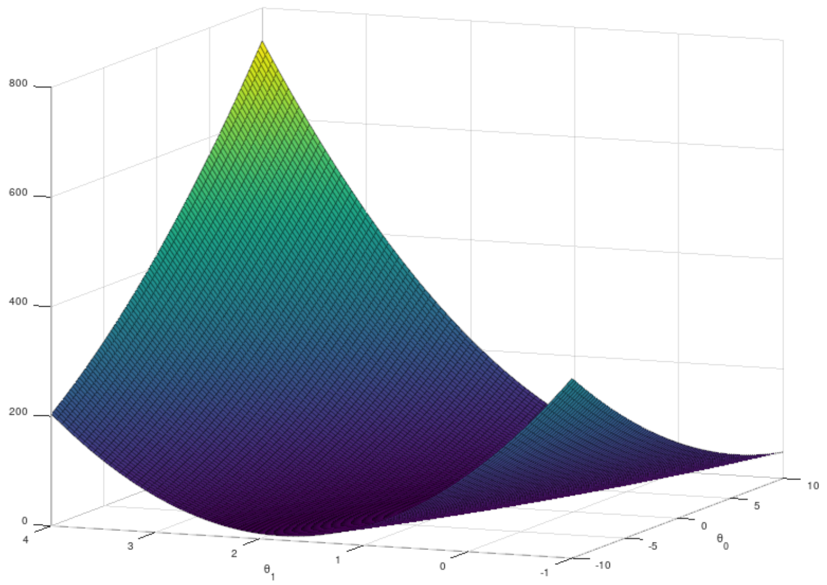
3 To do

You have to use the data01.txt which is a dataset of linear regression with one variable, you must perform the next tasks:

- Plot the data.
- Perform a manual method to find the Thetas that minimizes the cost.
- Implement the gradient descent algorithm.
- Choose and α of 0.01 and select 1500 iterations to find the best θ_1 and θ_0.
- Some results you must find are:
- - $\theta_0 = 0$ and $\theta_1 = 0$ Cost $=$ 32.073
  - Theta found by gradient descent: -3.630291 1.166362
  - For population = 35,000, we predict a profit of 4519.767868
  - For population = 70,000, we predict a profit of 45342.450129
- Plot the linear regression, you must have some similar to:



- Plot the surface that represents the cost for this problem, some similar to:



- (Optional but considered to evaluation) Plot the Thetas obtained after each iteration performed of gradient descent, you can do this into the surface above or use curves (level curves) to plot it.

4 Deadline

The deadline to make your report is 10 minutes before the next class arrive for each group. The report must be sent out to the instructor's email. You can use the TeX model from here All question and doubts must be done to the same email. For the report pay attention to the orthography and references used, be careful with all the ways of plagiarism, you must include the code into the report with all of its graphics generated, the only file to be sent is a pdf with your CUI as identification, no other files will be considered for the note.