share.coursera.org

ML:Gradient Descent - Coursera

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 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). This can be kind of confusing; we are moving up to a higher level of abstraction. We are not graphing x and y itself, but the parameter range of our hypothesis function and the cost resulting from selecting 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.

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 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, and the size of each step is determined by the parameter α, which is called the **learning rate**.

The gradient descent algorithm is:

$$\left\{ \begin{array}{l} \text{repeat until convergence:} \\ \left\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \right. \end{array} \right.$$

where

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

Intuitively, this could be thought of as:

repeat until convergence:

$\theta_j := \theta_j - \alpha[$
Slope of tangent aka derivative in j dimension$]$

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 (the derivation of the formulas are out of the scope of this course, but a really great one can be found here):

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i)x_i)$$

}

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 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.

Gradient Descent for Linear Regression: visual worked example

See this video that some may find useful as it visualizes

the improvement of the hypothesis as the error function reduces.

Frequently Asked Questions

- What is a **Gradient** (no calculus required)?

> A **gradient** is a mathematical operation that takes a scalar-valued function of multiple variables (e.g. the cost function) as an input, and computes a vector-valued output. The output of the gradient operator is called the **gradient of the input function**, or simply the **gradient vector**. The gradient vector contains information about the slope of the input function, and the direction of steepest ascent, at any location in the domain of the function. Specifically, the magnitude of the gradient vector is a measure of the slope of the input function, whereas the direction of the gradient vector is the same as the direction of steepest ascent.

- How is the gradient of the cost function used in **Gradient Descent**?

Next: Linear Algebra Review (Optional) Back to Index: Main