Computing Parameters Analytically - Normal Equation

August 6, 2021

1 Introduction

So far we have been using the gradient descent algorithm to find values for the parameters. The normal equation is a better way to solve for parameters of some linear regression problems.

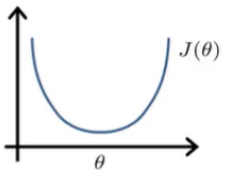
The normal equation solves for parameters analytically (in one step) as opposed to the iterative way of the gradient descent algorithm.

2 Normal Equation

Consider the parabolic cost function shown below

Intuition: If 1D ($\theta \in \mathbb{R}$)

$\rightarrow J(\theta) = a\theta^2 + b\theta + c$



To minimize such a function requires that we find the gradient function and set it equal to 0 and find the value for θ.

For more complex cost functions we perform the same but we use partial derivatives to arrive at the parameter in question that will minimize the cost function

$\theta \in \mathbb{R}^{n+1}$ $\quad J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$ (for every j)

Solve for $\theta_0, \theta_1, \ldots, \theta_n$

The normal equation uses this principle of partial derivatives and minimizing the cost function as part of its proof, however the proof will be skipped here and only the result of the normal equation is presented.

The normal equation relies on the x_0 feature and the construction of matrices for the input and output of the training set as well as for the parameters. Consider the example training set and corresponding features below

Examples: $m = 4$.

x_0	Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

constructing matrices for the input and output of this set below

Examples: $m = 4$.

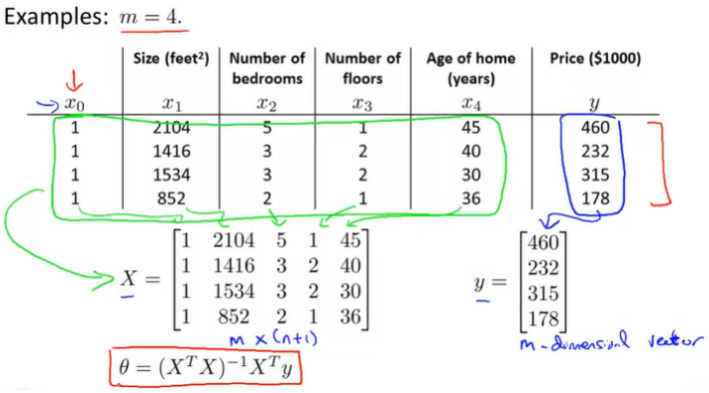
x_0	Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

m x (n+1)

m-dimensional vector

The normal equation used in finding the parameters resulting in a minimized cost function is given below



The beauty of using the normal equation to find the parameters of the hypothesis function is that it does not require feature scaling, don't need to define an arbitrary learning rate and it is not iterative (we arrive at our solution at once!). But of-course there are also some caveats with the normal equation. Obviously the matrices to be inverted need to be invertible (a matrix might not be invertible if there are redundant features - linear dependency, or if there are too many features - number of features are more than the number of training examples, although this can be solved via regularization or reducing the number of features but more on these in sections to come). Also using the normal equation can be slow for a large number of features (typically anything more than 10000 features from practice).

So the question now is, when to use what. Use what is more in favour of which algorithm as depicted below.