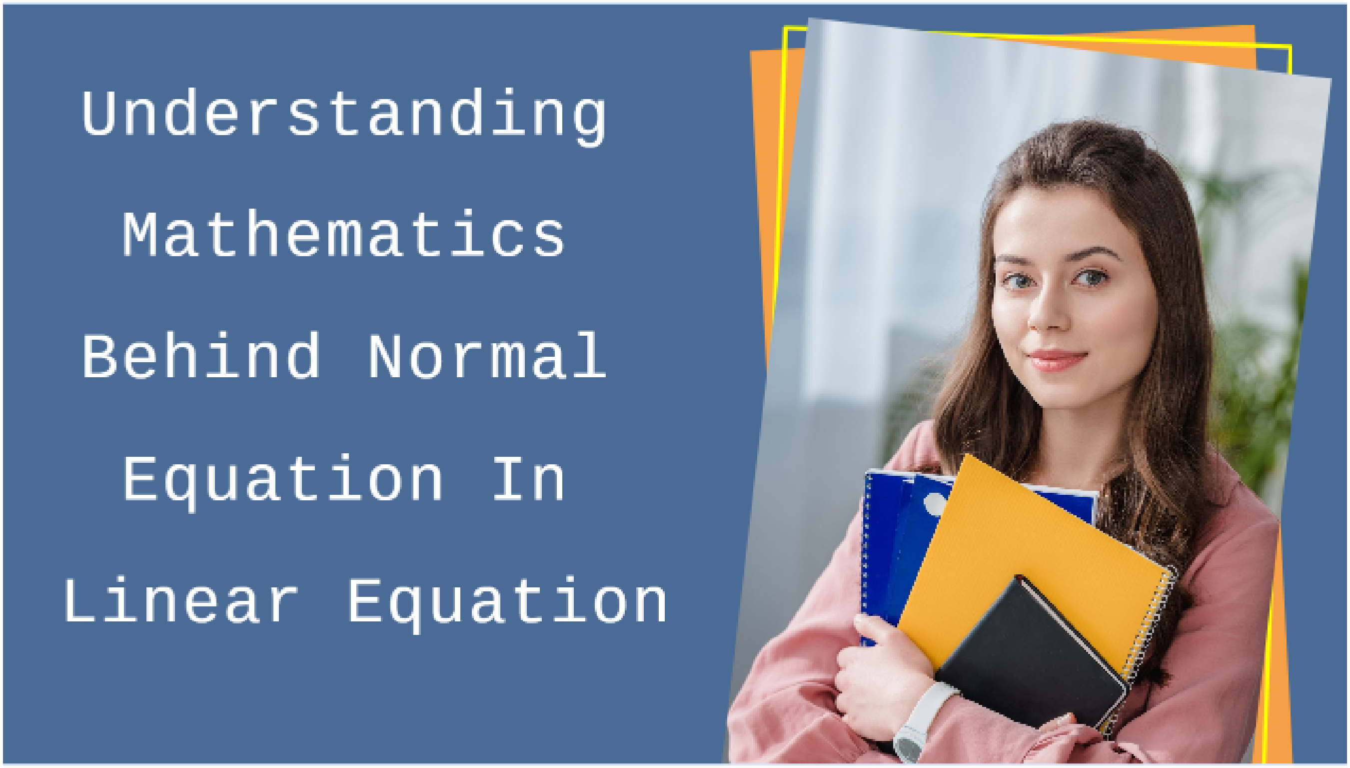
Published in The Startup

Pratik Shukla Follow

May 20, 2020 · 5 min read · · Listen

Linear Regression With Normal Equation Complete Derivation (Matrices)



Normal Equation is an analytic approach to Linear Regression with a least square cost function. We can directly find out the value of θ without using Gradient Descent. Following this approach is an effective and time-saving option when we are working with dataset with small features.

Normal Equation is as follows :

In the above equation :

θ : hypothesis parameters that define it the best.

X : input feature value of each instance

Y :Output value of each instance

Derivation Of Normal Equation:

(1) Hypothesis function :

$$h(\theta) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

Where,

n : number of features in the dataset.

X0 = 1 (for vector multiplication)

(2) Vector θ :

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

(3) Vector X:

$$\begin{bmatrix} \cdots \\ x_n \end{bmatrix}$$

(4) Notice that there is a dot product between θ and X. So we can write this as :

$$h(\theta) = \theta^T x$$

(5) Cost function :

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

Xi = the input value of ith trining example.

m : number of training instances.

n : number of dataset features

yi = the expected result for i th instance

(6) Representing the cost function in vector form:

$$\begin{bmatrix} h_\theta x^{(0)} \\ h_\theta x^{(1)} \\ \cdots \\ h_\theta x^{(m)} \end{bmatrix} - \begin{bmatrix} y^0 \\ y^1 \\ \cdots \\ y^m \end{bmatrix}$$

(7) We are going to ignore 1/2m here since it’s not going to make any difference in the derivation.

$$\begin{bmatrix} \theta^T(x^1) \\ \cdots \\ \theta^T(x^m) \end{bmatrix} - y$$

$$\begin{bmatrix} \theta_0(x_0^0) + \theta_0(x_1^0) + \dots + \theta_n(x_n^0) \\ \theta_0(x_0^1) + \theta_0(x_1^1) + \dots + \theta_n(x_n^1) \\ \cdots \\ \theta_0(x_0^m) + \theta_0(x_1^m) + \dots + \theta_n(x_n^m) \end{bmatrix} - y$$

x_j^i = Value of jth feature in ith training example

(8) This can further be reduced to :

$$X\theta - y$$

(9) But in our cost function there is a square. We can't simply square the above expression. As the square of vector/matrix is not equal to square of it of its values. so to get the squared values,multiply the vector/matrix with it's transpose.

Introduction To Matrices (For Machine Learning)

In this article I'm going to show some basic operations that we can perform on matrices like addition, multiplication…

medium.com

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y)$$

(10) Ignore the 1/2m since it's not going to make any difference in out derivation.

$$J(\theta) = ((X\theta)^T - y^T)(X\theta - y)$$

$$J(\theta) = (X\theta)^T X\theta - (X\theta)^T y - y^T(X\theta) + y^T y$$

(11)Now in the above equation 2nd and 3rd terms are same.(Explained in my previous article.) So add them.

$$J(\theta) = \theta^T X^T X\theta - 2(X\theta)^T y + y^T y$$

(12) Our ultimate goal is to minimize the cost function.Finding derivatives.

$$P(\theta) = 2(X\theta)^T y$$

(13) Writing it in a vector form :

$$P(\theta) = 2\left[\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & & & \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}\begin{pmatrix} \theta_1 \\ \theta_2 \\ \cdots \\ \theta_n \end{pmatrix}\right]^T \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix}$$

$$\begin{bmatrix} x_{21}\theta_1 + \dots + x_{2n}\theta_n \\ \dots \\ x_{m1}\theta_1 + \dots + x_{mn}\theta_n \end{bmatrix} \begin{pmatrix} \vdots \\ y_2 \\ \dots \\ y_m \end{pmatrix}$$

(14) Multiplying the vectors:

$$P(X) = 2(x_{11}\theta_1 + \dots + x_{1n}\theta_n)y_1 + \dots + 2(x_{m1}\theta_1 + \dots + x_{mn}\theta_n)y_m$$

(15) Simplifying it a bit :

$$P(x) = 2\sum_{r=1}^{m} y_r(x_{r1}\theta_1 + \dots + x_{rn}\theta_n)$$

$$= 2\sum_{r=1}^{m} y_r \cdot \sum_{c=1}^{n} x_{rc}\theta_c$$

(16) Find the derivatives :

$$\frac{\partial P}{\partial \theta_1} = 2(x_{11}y_1 + \dots + x_{m1}y_m)$$

$$\frac{\partial P}{\partial \theta_2} = 2(x_{12}y_1 + \dots + x_{m2}y_m)$$

$$\frac{\partial P}{\partial \theta_n} = 2(x_{1n}y_1 + \dots + x_{mn}y_m)$$

(17) In Conclusion:

$\partial \Theta$

(18) Finding the other derivative:

$$Q(\Theta) = \Theta^T X^T X \Theta$$

(19)Writing in vector form :

$$Q(\Theta) = (\Theta_1 \dots \Theta_n) \begin{pmatrix} x_{11} & x_{21} & \dots & x_{m1} \\ x_{12} & x_{22} & \dots & x_{m2} \\ \dots \\ x_{1n} & x_{2n} & \dots & x_{mn} \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \dots \\ \Theta_n \end{pmatrix}$$

$$Q(\Theta) = (\Theta_1 \dots \Theta_n) \begin{pmatrix} x_{11}^2 \Theta_1 + \dots + x_{1n}^2 \Theta_n \\ x_{21}^2 \Theta_1 + \dots + x_{2n}^2 \Theta_n \\ \dots \\ \dots + x_{mn}^2 \Theta_n \end{pmatrix}$$

(20) Simplifying :

$$Q(\Theta) = \Theta_1 (x_{11}^2 \Theta_1 + \dots + x_{1n}^2 \Theta_n) + \dots + \Theta_n (x_{n1}^2 \Theta_1 + \dots + x_{nn}^2 \Theta_n)$$

(21) Finding partial derivative :

$$\frac{\partial Q}{\partial \Theta_1} = 2\Theta_1 x_{11}^2 + 2\Theta_1 x_{12}^2 + \dots + 2\Theta_n x_{1n}^2$$

$$\frac{\partial}{\partial \theta} y^T y = 0$$

(23) Combining all together :

$$\frac{\partial J}{\partial \theta} = \frac{\partial Q}{\partial \theta} - \frac{\partial P}{\partial \theta}$$

$$\frac{\partial J}{\partial \theta} = 2X^T X\theta - 2X^T y = 0$$

(24) Simplifying :

$$2X^T X\theta - 2X^T y = 0$$

$$2X^T X\theta = 2X^T y$$

(25) Final step :

$$X^T X\theta = X^T y$$

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

$$\boxed{\theta = \left(X^T X\right)^{-1} . \left(X^T y\right)}$$

Open in app

Moving Forward,

In the next article I will show you how we can implement Linear Regression using the Normal Equation we just derived.

For more such detailed descriptions on machine learning algorithms and it's derivation and implementation, you can follow me on my blog:

All my articles are available on my blog : patrickstar0110.blogspot.com

Watch detailed videos with explanations and derivation on my youtube channel :

(1) Simple Linear Regression Explained With It's Derivation:
https://youtu.be/1M2-Fq6wl4M

(2)How to Calculate The Accuracy Of A Model In Linear Regression From Scratch :
https://youtu.be/bM3KmaghclY

(3) Simple Linear Regression Using Sklearn :
https://youtu.be/_VGjHF1X9oU

Read my other articles :

(1) Linear Regression From Scratch :
https://medium.com/@shuklapratik22/linear-regression-from-scratch-a3d21eff4e7c

(2) Linear Regression Through Brute Force :
https://medium.com/@shuklapratik22/linear-regression-line-through-brute-force-1bb6d8514712

(3) Linear Regression Complete Derivation:
https://medium.com/@shuklapratik22/linear-regression-complete-derivation-406f2859a09a

from-scratch-cb4a478c42bc

(5) Simple Linear Regression From Scratch :
https://medium.com/@shuklapratik22/simple-linear-regression-implementation-2fa88cd03e67

(6) Gradient Descent With it's Mathematics :
https://medium.com/@shuklapratik22/what-is-gradient-descent-7eb078fd4cdd

(7) Linear Regression With Gradient Descent From Scratch :
https://medium.com/@shuklapratik22/linear-regression-with-gradient-descent-from-scratch-d03dfa90d04c

(8) Error Calculation Techniques For Linear Regression :
https://medium.com/@shuklapratik22/error-calculation-techniques-for-linear-regression-ae436b682f90

(9) Introduction to Matrices For Machine Learning :
https://medium.com/@shuklapratik22/introduction-to-matrices-for-machine-learning-8aa0ce456975