Linear Models

Rohan Saha

About me

MSc student at UofA - Computing Science

Research : Machine Learning and Neuroscience

Volunteer at GDG Edmonton working as a technical speaker

I love bodybuilding and cooking (sometimes fail).

Prediction Problem

Given some input **X**, predict an output **Y**.

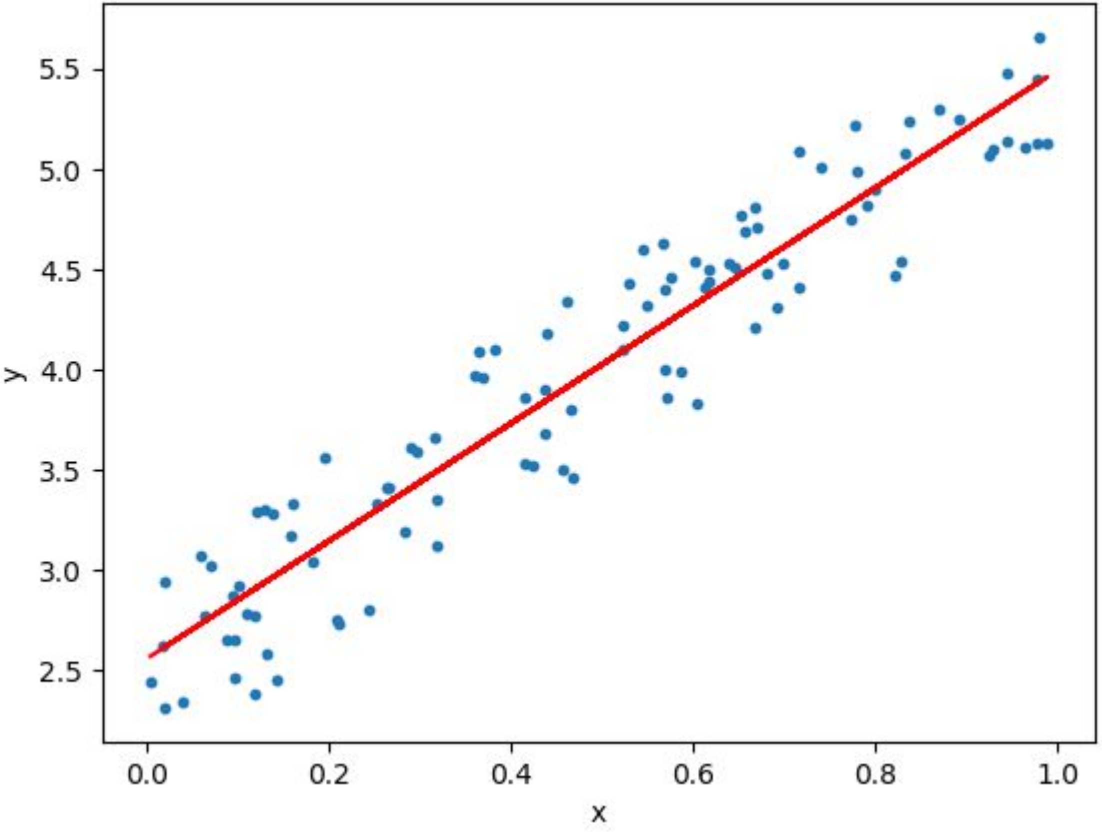
$$\mathbf{Y} = f(\mathbf{X})$$

F can be linear or non-linear.

X is (usually) a matrix.

Y is (usually) a value of set of values.

We will focus on Linear functions for now.



$$Y = \Theta \, . \, X$$

Here Θ is the transformation applied on X.

Task: Find (best) Θ

https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2

Goal : find Θ.

Hypothesis: $Y = f(X) = \Theta.X$

For simplicity: Let X be one valued variable.

Therefore: $Y = f(X) = \Theta_1.X$ is our hypothesis.

Θ's are called parameters.

Different Θ -> Different hypothesis

Θ can take values like.

$\Theta = 0.5$

$\Theta = 1.4$

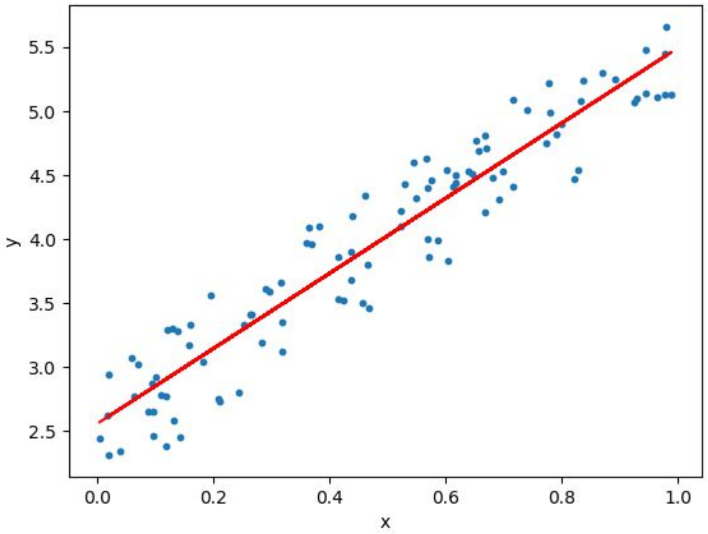
$\Theta = 2$

Θ = any real value.

Cost Function

In the hypothesis, we also add a bias term for the intercept.

So hypothesis is now: $Y = f(X) = \Theta_0 + \Theta_1.x$, where Θ_0 is the bias.



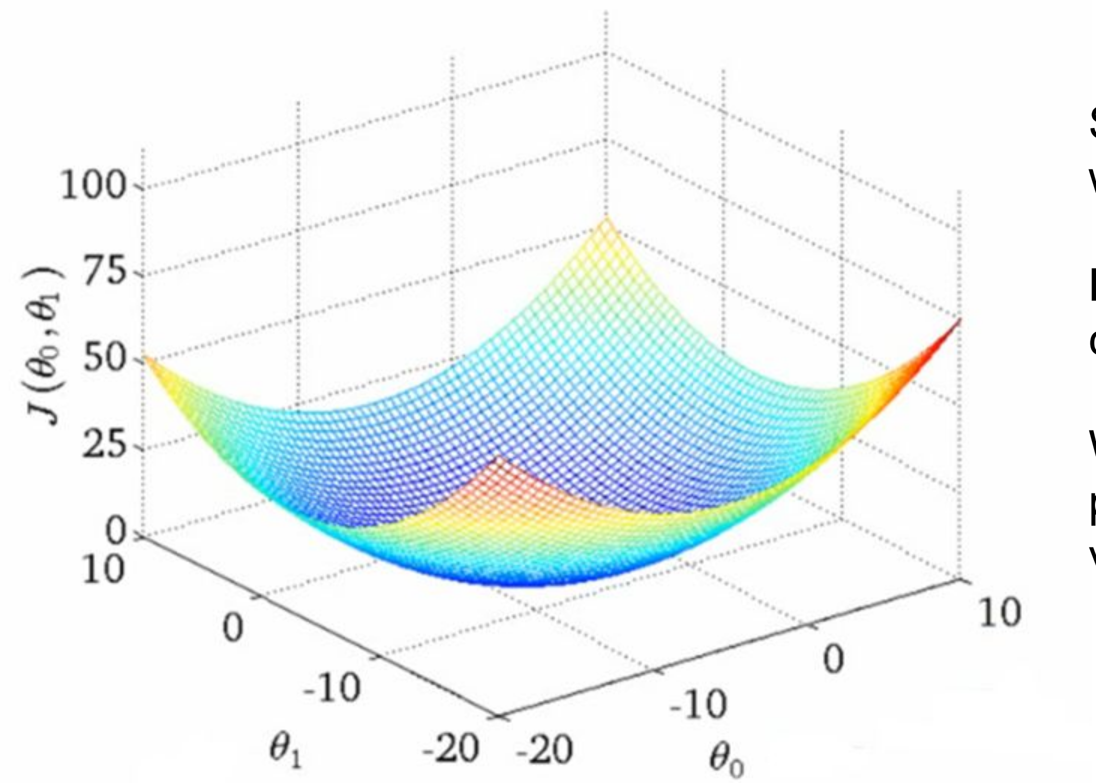
$$MSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

MSE = Mean Squared Error cost function.

$MSE = J(\Theta_0, \Theta_1)$

Goal: Minimize cost function.

Visualizing Cost Function



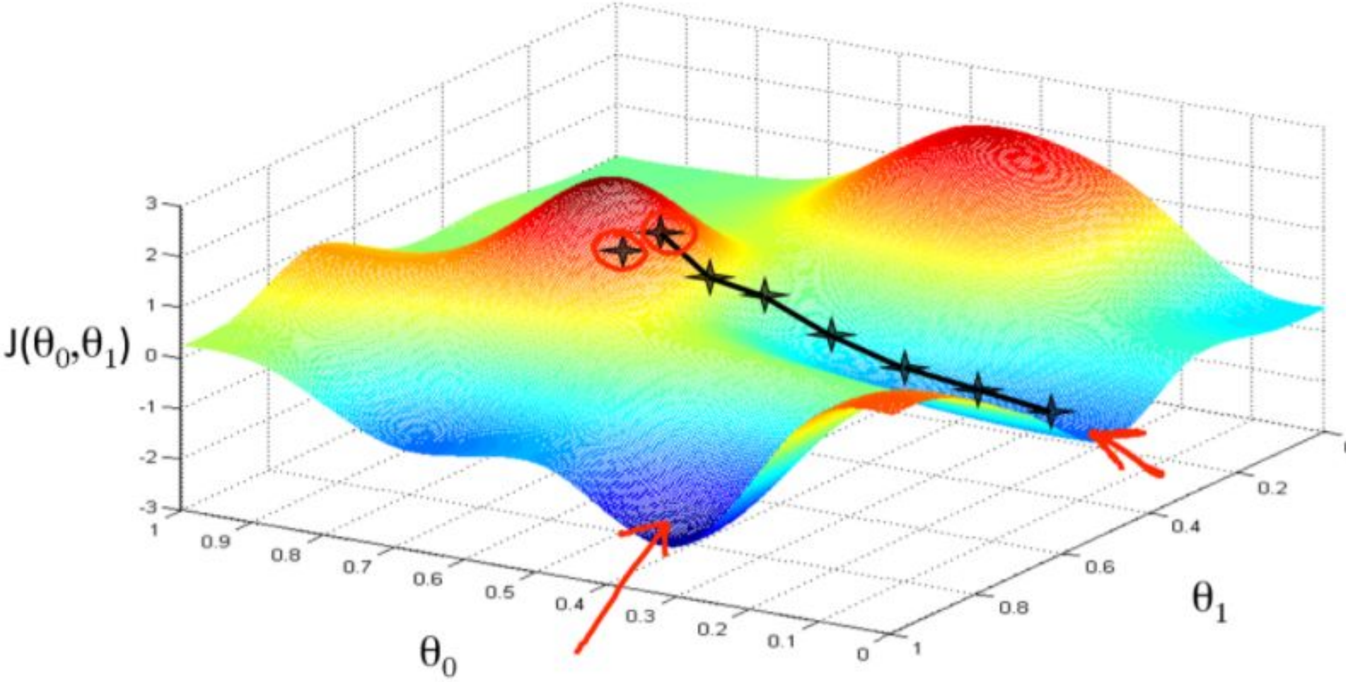
Shows how the cost function varies with different parameter values.

Linear Regression has a convex cost function.

We want to find values of parameters to obtain the minimal J value.

https://medium.com/analytics-vidhya/linear-regression-hypothesis-function-cost-function-and-gradient-descent-part-2-730b13959b3c

Gradient Descent



Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

α is the learning rate.

J(Θ) is the cost function.

https://hackernoon.com/gradient-descent-aynk-7cbe95a778da

Types of Gradient Descent:

- Stochastic Gradient Descent
  - Update after each sample
- Batch Gradient Descent
  - Update after going through all the samples
- Mini-Batch Gradient Descent
  - Update after going through a subset of samples

Linear Regression with Multiple Variables

- Input data has more than one variable (feature)
  - **X** = (x1, x2, x3, x4, x5, … , xn)
- More parameters
  - $\mathbf{\Theta} = (\Theta_1, \Theta_2, \Theta_3, \Theta_4, \dots, \Theta_n)$
- Helps to Vectorize using Linear Algebra!

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

$$\Rightarrow X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}, \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\Rightarrow h_\theta(x) = \theta^T x,$$

Q: What is the value of x_0?

Cost function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

Closed-form solution

Usually used when we have many features (for the input data)

No iterations required.

No step size selection

Can be computationally intensive if #features is large.

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