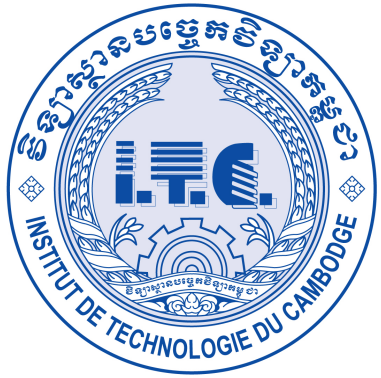


Gradient Descent Method

PHOK Ponna

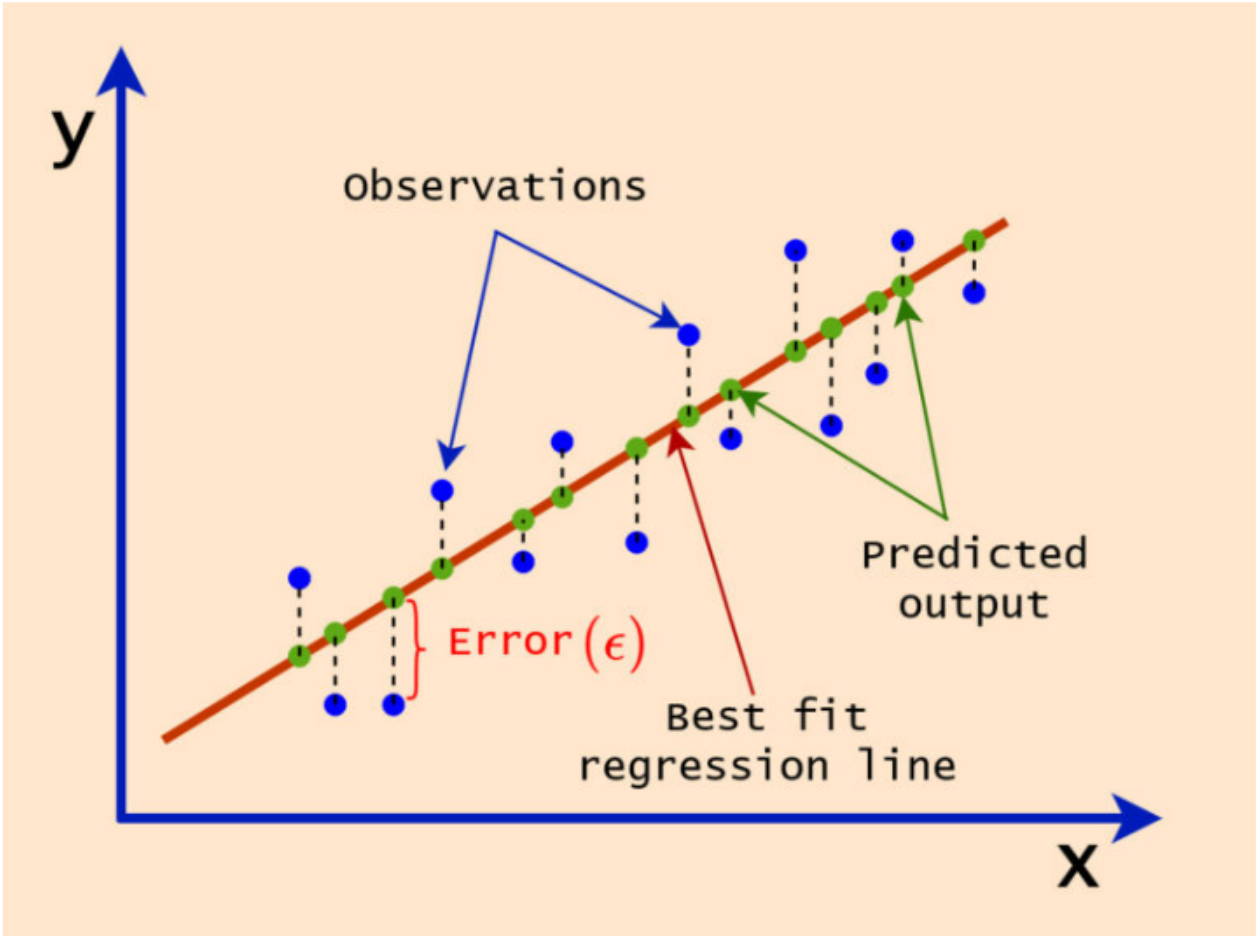
Institute of Technology of Cambodia
Department of Applied Mathematics and Statistics (AMS)

2022–2023

Contents

Contents

Linear regression is a supervised algorithm that draws a linear relationship model between the independent variables (inputs, sometimes it is called regressors or predictors) and the dependent variable (output or outcome). In simple words, the idea of Linear regression is to find the line that fits the observed data and use it to predict the future outcome from new inputs or unseen data. The observed data can be a historical data that has been collected for a specific purpose within a certain duration.
The simple Linear regression consists of only one independent variable (x) and one dependent variable (y) as illustrated in the following figure. What we actually do in Linear regression is basically to find the best parameters that fits the straight line to the observations by minimizing the errors between the predicted outcomes and the observations. Meaning that, the best model in Linear regression is the linear equation model with the best parameters.


y
Observations
Predicted
output
Error (ϵ)
Best fit
regression line
X

$(x^T)_{1\times(n+1)}\ \theta_{(n+1)\times 1}$

A simple Linear regression with one independent variable and one dependent variable is defined as:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

where $h_\theta(x)$ is the hypothesis function, x is the input, and θ_0 and θ_1 are the parameters of the model. The subscript θ means that the hypothes is depends not only on the input x but also on the parameters θ_0 and θ_1. We use θ_0 and θ_1 to represent the y-intercept and the slope, respectively.

In Linear regression, we usually have one dependent variable as a target output. The inputs, however, can consist of one or more independent variables. We've just looked at a simple Linear regression with only one input variable. Now, we're going to look at multi-variable Linear regression.

Let

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \text{and} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

Then the hypothesis of Linear regression given x and θ can be written as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n = \sum_{j=0}^{n} \theta_j x_j \qquad \text{where } x_0 = 1 \ .$$

So, the hypothesis of Linear regression can be written it in a vectorized form as follows:

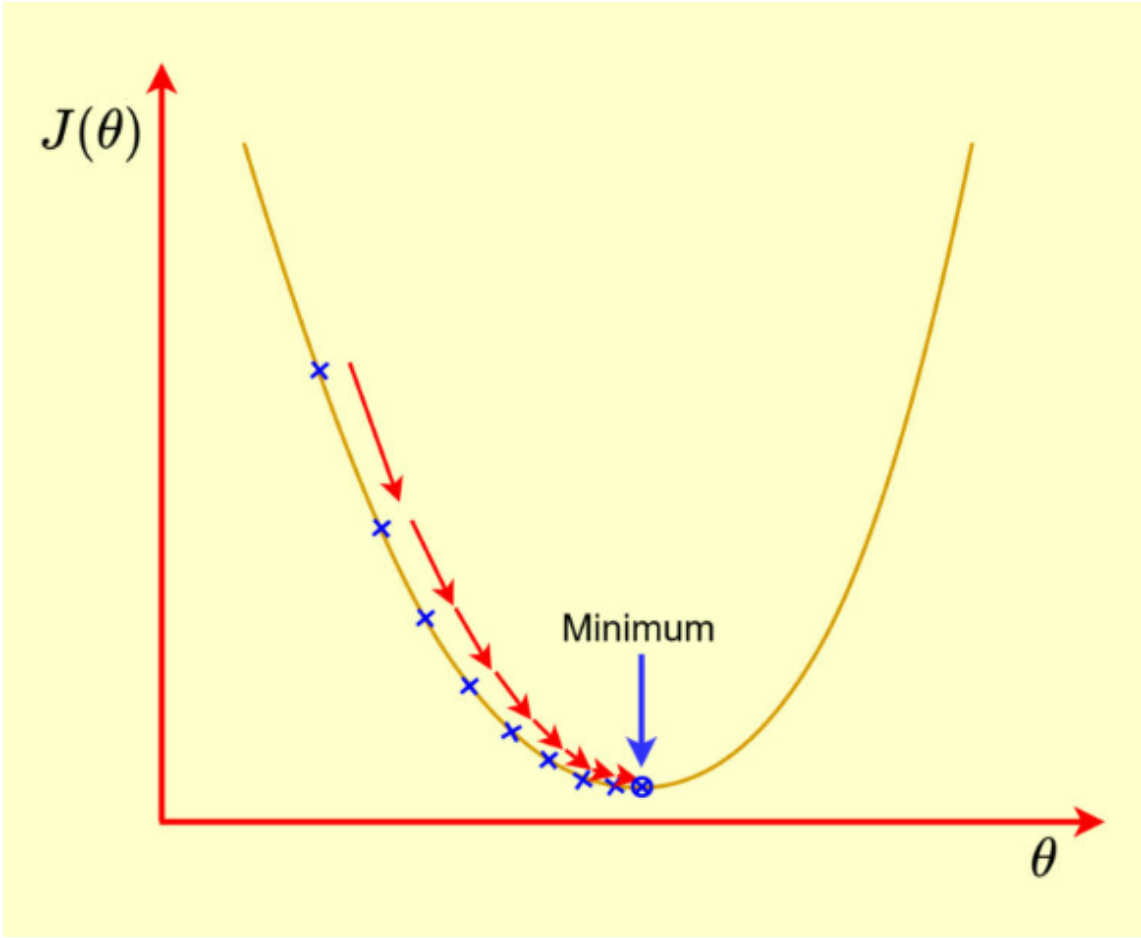
$$h_\theta(\mathbf{x}) = \mathbf{x}^T.\theta, \qquad \text{where } x_0 = 1 \ .$$

Remark 1

The input $\mathbf{x}$ contain $n+1$ variables, $x_0, x_1, x_2, ..., x_n$. These variables are called as features. So, $\mathbf{x}$ contains $n+1$ features.

Contents

Gradient descent is an optimization algorithm that is commonly used in machine learning to learn a model by focusing on minimizing the cost function $J(\theta)$ w.r.t the parameters θ . The cost function is nothing but the error between the hypothesis $h_\theta(\mathbf{x})$ and the observed data $\mathbf{y}$. During the learning process, we iteratively calculate the gradient of the cost function and update the parameters of the model. The parameters are updated in the opposite direction of the gradient of $J(\theta)$.
There are several types of Gradient descent, including batch-Gradient descent, stochastic Gradient descent (SGD), and mini-batch Gradient descent. Each type has its own trade-offs in terms of computational efficiency and the accuracy of updating the parameters. However, in section, we'll be focusing only on the vanila Gradient descent or better known as batch Gradient descent.


$J(\theta)$
Minimum
θ

What we do basically in Gradient descent is to update the parameters by minimizing the cost function $J(\theta)$. We iterate the updating process until we have the smallest error possible. To do so, we need to define our cost function first. So in this tutorial we define our cost function $J(\theta)$ as a half-Mean Square Errors $(\frac{1}{2}MSE)$ between the hypothesis and the observations.
For m number of training set $\{(x^{(i)}, y^{(i)}); i = 1, 2, .., m\}$, where $x \in \mathbb{R}^{m\times(n+1)}, y \in \mathbb{R}^{m+1}$, and $\theta \in \mathbb{R}^{n+1}$, the cost function $J(\theta)$ can be written as follows:

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

If we just want to consider for only one training example (x, y), $m = 1$, that makes $\mathrm{x} \in \mathbb{R}^{n+1}$, $\theta \in \mathbb{R}^{n+1}$, and $y \in \mathbb{R}$, the cost function $J(\theta)$ can be written as:

$$J(\theta) = \frac{1}{2}\Big(h_\theta(\mathrm{x}) - y\Big)^2$$

When we are talking about the Gradient descent, we need to know first about what the gradient is. Actually, the gradient of a function f, denoted as ∇f, is nothing but a vector of its partial derivative.
Gradient of a multi-variable function $f(\theta_0, \theta_1, \theta_2, ..., \theta_n)$ is writen as in the following form:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial \theta_0} \\ \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_n} \end{bmatrix}$$

Where ∂ is the partial derivative operator that differenciates a function with respect to one variable and treats the others as constants.
So, the gradient of $J(\theta)$ can be written as $\nabla J(\theta)$.

We already know that the parameters will be updated iteratively in the opposite direction of $\nabla J(\theta)$ during the learning process. Therefore, we can define the update rule for the parameters as the following:

$$\theta := \theta - \alpha \nabla J(\theta)$$

Where α is the **learning rate** that defines how far to go for each step. α can range from 0 to 1 $(0 < \alpha < 1)$.
If we let θ be a column vector and $\theta \in \mathbb{R}^{n+1}$, then we have

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \qquad \forall j = 1, 2, \ldots, n$$

Gradient Descent for a Single Training Example $(m=1)$

- The cost function $J(\theta)$ for a single training example (x, y), where $\mathrm{x} \in \mathbb{R}^{n+1}, y \in \mathbb{R}$, and $\theta \in \mathbb{R}^{n+1}$: $h_\theta(x) = \theta_0 + \theta_1 x$

$$J(\theta) = \frac{1}{2}\left(h_\theta(\mathrm{x}) - \mathrm{y}\right)^2 \qquad , \forall j = 1, 2, \ldots, n$$

- Updated parameters

$$\theta_j := \theta_j - \alpha . \left(h_\theta(\mathrm{x}) - y\right) x_j \qquad , \forall j = 1, 2, \ldots, n$$

$\theta_j = \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta)$, since $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j}\left[\frac{1}{2}\left(h_\theta(x) - y\right)^2\right]$

$= 2 \times \frac{1}{2}\left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j}\left(h_\theta(x) - y\right)$

$$\frac{\partial}{\partial \theta_j} J(\theta) = (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

$$= (h_\theta(x) - y) \cdot x_j \quad , \forall j = 1, 2, \ldots, n$$

Then $\theta_j = \theta_j - \alpha \cdot (h_\theta(x) - y) \cdot x_j \quad , \forall j = 1, 2, 3, \ldots, n$. □

Gradient Descent for Multiple Training Examples

- Since we are dealing with multiple training examples $\{(\mathrm{x}^{(i)}, \mathrm{y}^{(i)}); i = 1, 2, .., m\}$, where $\mathrm{x} \in \mathbb{R}^{m \times (n+1)}$, $\mathrm{y} \in \mathbb{R}^{m}$, and $\theta \in \mathbb{R}^{n+1}$, we rewrite the cost function $J(\theta)$ as:

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

- Updated parameters

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left(h_\theta(\mathrm{x}^{(i)}) - \mathrm{y}^{(i)} \right) x_j^{(i)} \qquad \forall j = 1, 2, \ldots, n$$

$x_0 = 1$

$0, 1, \ldots, n$

$i = 1, 2, \ldots, m.$

Because $\theta_j = \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta) = \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} \left[\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \right]$

$= \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x^{(i)}) - y^{(i)})$

$$\theta_j = \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)} \quad . \square$$

Gradient Descent in Vectorized Form

If we have $\mathrm{x} \in \mathbb{R}^{m \times (n+1)}$, $\mathrm{y} \in \mathbb{R}^{m \times 1}$, and $\theta \in \mathbb{R}^{(n+1) \times 1}$, then $\mathrm{x}, \mathrm{y}, \theta$ are written as follows:

$$\mathrm{x} = \begin{bmatrix} x_0^1 & x_1^1 & x_2^1 & x_3^1 & \dots & x_n^1 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 & \dots & x_n^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_0^m & x_1^m & x_2^m & x_3^m & \dots & x_n^m \end{bmatrix}, \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_n \end{bmatrix}, \text{ and } \mathrm{y} = \begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ \vdots \\ y^m \end{bmatrix}$$

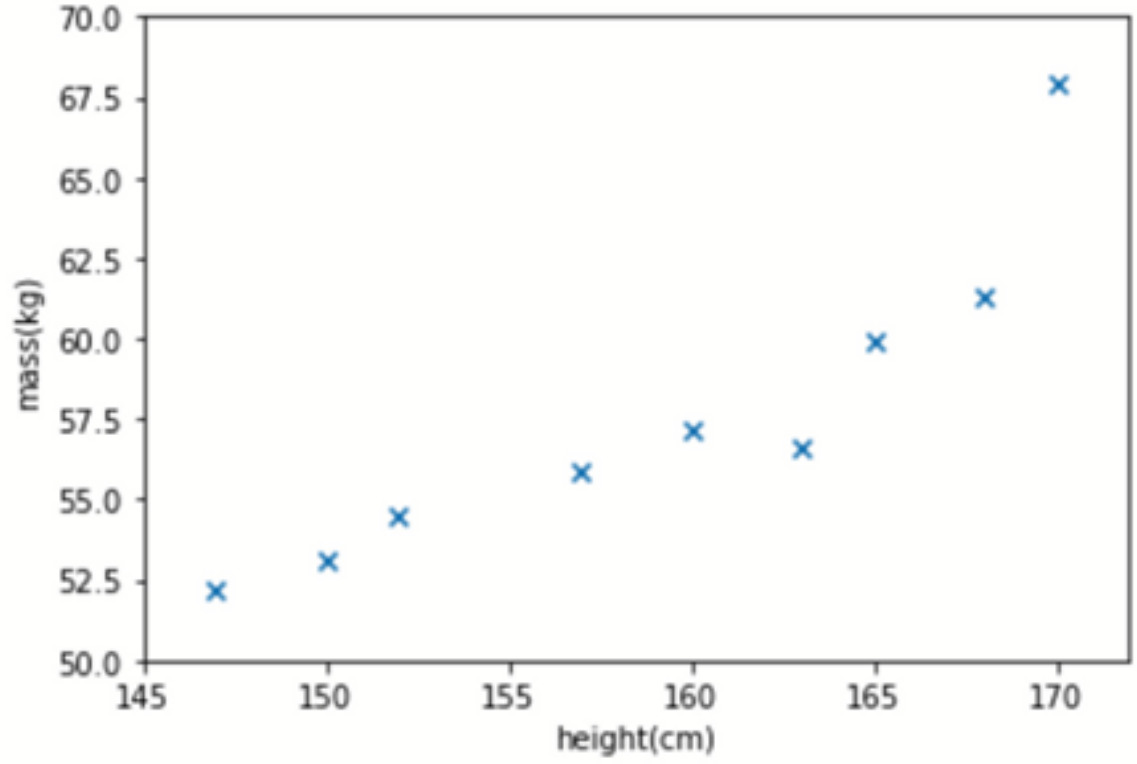
$$\theta := \theta - \frac{\alpha}{m} \left(\mathrm{x}^T (\mathrm{x}\theta - \mathrm{y}) \right)$$

Contents

content...

តារាងទី១ ទិន្នន័យកម្ពស់(cm)និងម៉ាស(kg)មនុស្ស៩នាក់

កម្ពស់(cm)	152	157	160	163	150	147	165	168	178
ម៉ាស(kg)	54.48	55.84	57.20	58.57	53.12	52.21	59.93	61.29	69.92



y

x

$y = \theta_0 + \theta_1 x$ or $y = ax + b$ $\Rightarrow h_\theta(x) = ax + b$

- Hypothesis function: $h_\theta(x) = \theta_0 + \theta_1 \cdot x$
- Cost function: $J(\theta) = \frac{1}{2}(h_\theta(x) - y)^2$
- Initialize $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix}$, $\alpha = ?$
- Gradient of $J(\theta)$: $\frac{\partial}{\partial \theta_j} J(\theta)$, $j = 1, 2$
- $\theta_j = \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta)$
- $h_{a,b}(x) = ax + b$
- $J(a,b) = \frac{1}{2}(h_{a,b}(x) - y)^2 = \frac{1}{2}(ax + b - y)^2$

  $\frac{\partial}{\partial a} J(a,b) = (ax + b - y) \cdot x$, $\frac{\partial}{\partial b} J(a,b) = (ax + b - y)$