COMP2261 ARTIFICIAL INTELLIGENCE / MACHINE LEARNING

Multivariate Linear Regression

-- Cost Function & Gradient Descent

Dr SHI Lei

Durham University
learning lab

Supervised Learning

- To build a model represented as a hypothesis function $h_\theta(x)$.

Supervised Learning

income (input x, dependent variable)

↓

model (hypothesis function, mapping $x \rightarrow y$) ⇐ cost function (loss function)

↓

happiness (output y, independent variable)

Univariate Linear Regression

Hypothesis Function

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

Independent Variable

$$x$$

Parameters

$$\theta_0, \theta_1$$

Cost Function

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

Univariate Linear Regression → Multivariate Linear Regression

Hypothesis Function

$h_\theta(x) = \theta_0 + \theta_1 x$ $\longrightarrow$ $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \cdots + \theta_n \cdot x_n$

Independent Variable

x $\longrightarrow$ $\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\quad (x_0 = 1\,, constant)$

Model Parameters

θ_0, θ_1 $\longrightarrow$ $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\quad h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$

vectors

Cost Function

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

Univariate Linear Regression

$(n = 1)$

Multivariate Linear Regression

Cost Function

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

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

Partial derivative of J with respect to parameters

$$\frac{\partial J}{\partial \theta_0}(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)\boxed{x_0^{(i)}} \quad x_0^{(i)} = 1$$

$$\frac{\partial J}{\partial \theta_1}(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)\boxed{x_1^{(i)}}$$

$$\frac{\partial J}{\partial \theta_0}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)\boxed{x_0^{(i)}} \quad x_0^{(i)} = 1$$

$$\frac{\partial J}{\partial \theta_1}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)\boxed{x_1^{(i)}}$$

...

$$\frac{\partial J}{\partial \theta_n}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)x_n^{(i)}$$

Univariate Linear Regression

Gradient Descent

Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \cdot \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)}$$

learning rate

$\frac{\partial J}{\partial \theta_0}(\theta_0, \theta_1)$

$$\theta_1 := \theta_1 - \alpha \cdot \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)x^{(i)}}$$

learning rate

$\frac{\partial J}{\partial \theta_1}(\theta_0, \theta_1)$

} simultaneously update θ_0, θ_1

Multivariate Linear Regression

Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \cdot \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)x_0^{(i)}}$$

learning rate

$\frac{\partial J}{\partial \theta_0}(\boldsymbol{\theta})$

$$\theta_1 := \theta_1 - \alpha \cdot \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)x_1^{(i)}}$$

learning rate

$\frac{\partial J}{\partial \theta_1}(\boldsymbol{\theta})$

...

$$\theta_n := \theta_1 - \alpha \cdot \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)x_n^{(i)}}$$

learning rate

} simultaneously update $\theta_0, \dots, \theta_n$

$\frac{\partial J}{\partial \theta_n}(\boldsymbol{\theta})$

Takeaway Points

- Univariate linear regression is a special case of multivariate linear regression when the number of features $n=1$.
- n+1 dimensional column vectors to denote features and model parameters.
- Feature vectors and parameters vectors to express hypothesis function and cost function.
- Each iteration in gradient descent, all the n parameters (θs) need to be updated simultaneously.