Introduction to non-regularized regression

Jae Yun JUN KIM*

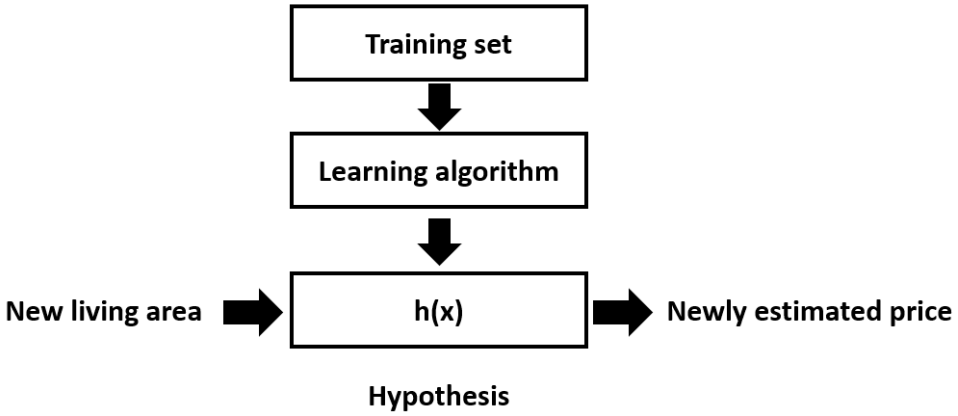
1 Motivation

Living Area [m^2]	nearest access to public transport [m]	Price [1000 Euros]
45.5	10.1	200.3
60.7	20.3	300.5
⋮	⋮	⋮

2 Notation

I	number of training examples
x	input variables/features
y	output variable / "target" variable
(x, y)	training example
$(x^{(i)}, y^{(i)})$	i^{th} training example

3 Hypothesis function



$$\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2, \tag{1}$$

where x_1 and x_2 are the living area size and the room number, respectively.

*ECE Paris Graduate School of Engineering, 37 quai de Grenelle 75015 Paris, France; jae-yun.jun-kim@ece.fr

For conciseness, define $x_0 = 1$ and

$$\hat{y} = h_\theta(x) = \sum_{n=0}^{N} \theta_n x_n = \overline{x}^T \theta, \tag{2}$$

where N is the number of features, and θ are called **model parameters**.

4 Cost minimization problem

What we would like to do is to find θ's that minimize

$$E(\theta) = \frac{1}{2} \sum_{i=1}^{I} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2. \tag{3}$$

That is,

$$\min_\theta E(\theta). \tag{4}$$

Now that the cost minimization problem is stated, we turn our attention to the question to how we can solve this problem.

There are mainly three approaches to solve this problem.

- Batch gradient descent
- Stochastic gradient descent
- Closed-form solution

5 Batch gradient descent (BGD)

$$\theta_n \leftarrow \theta_n - \alpha \frac{\partial}{\partial \theta_n} E(\theta), \tag{5}$$

where

$$\begin{aligned}
\frac{\partial}{\partial \theta_n} E(\theta) &= \frac{\partial}{\partial \theta_n} \left[\frac{1}{2} \sum_{i=1}^{I} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2 \right] \\
&= \frac{1}{2} \sum_{i=1}^{I} \frac{\partial}{\partial \theta_n} \left[\left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2 \right] \\
&= \frac{1}{2} \sum_{i=1}^{I} 2 \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) \frac{\partial}{\partial \theta_n} \left[\theta_0 x_0 + \cdots + \theta_n x_n + \cdots + \theta_N x_N \right] \\
&= \sum_{i=1}^{I} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) x_n^{(i)} \\
&= \sum_{i=1}^{I} \left(\left(\overline{x}^{(i)} \right)^{\mathrm{T}} \theta - y^{(i)} \right) x_n^{(i)}
\end{aligned} \tag{6}$$

Hence, the **update rule** is

$$\theta_n \leftarrow \theta_n - \alpha \sum_{i=1}^{I} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_n^{(i)}. \tag{7}$$

For more details, look at the lecture note that you took in class.

6 Stochastic gradient descent (SGD)

However, when $I >> 1$, the *batch gradient descent* may be very inefficient. Alternatively, one can use the **stochastic gradient descent**.

Hence, the **update rule** is

$$\begin{aligned} i &\sim \mathcal{U}[1, I] \\ \theta_n &\leftarrow \theta_n - \alpha \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_n^{(i)}. \end{aligned} \tag{8}$$

This will give some approximate convergence, but it will converge more quickly. Hence, the choice between the *batch gradient descent* and the *stochastic gradient descent* is a trade-off between the accuracy and efficiency, respectively.

For more details, look at the lecture note that you took in class.

7 Closed-form solution(CFS) or Ordinary Least Square (OLS)

The optimal model parameters can be computed as

$$\theta^* = (\overline{X}_{\text{train}}^T \overline{X}_{\text{train}})^{-1} \overline{X}_{\text{train}}^T Y_{\text{train}}. \tag{9}$$

For more details, look that the lecture note that you took in class.

8 Using the optimal model parameter values

From Section 5, Section 6 and Section 7, you should be able to obtain θ^*. Now, using θ^* (from any of the above three methods), you can predict the optimal output values for both training and test.

For more details, look that the lecture note that you took in class.