Università della Svizzera italiana

Institute of Computational Science ICS

Example of Linear Regression with one variable using Utopia

With linear regression, we are searching for parameters, such that our model gives the most reasonable prediction. To do this, we need to minimise the cost function $J(\Theta) : \mathbb{R}^2 \to \mathbb{R}$, which basically tells us the difference between the actual value y and the value predicted by our hypothesis function.

For this example, we rely on a dataset consisting of s samples. Each sample is defined by an input x_i and an output y_i. The input x_i expresses a population and the output y_i expresses the profit earned by a foodtruck of a certain company. Supposing that we would like to have a foodtruck in a city with a certain population, we would be able to predict the profit for the food truck, thanks to our model parametrized by $\Theta \in \mathbb{R}^2$. For examples with a population of 43'000, we will be able to write a vector [1, 4.3], - with 1 being the bias, - and by multiplying this vector by Θ we will obtain an estimation of the profit. The cost function is defined as:

$$J(\Theta) = \frac{1}{2s} \sum_{i=1}^{s} (h_\Theta(\Theta^{(i)}) - y^{(i)})^2$$

In our implementation, each population example corresponds to a row in our vector $x \in \mathbb{R}^s$, to which we add an additional columns and we set it to all ones to add the bias. The vector x then becomes a matrix $X \in \mathbb{R}^{s \times 2}$. Similarly, $y \in \mathbb{R}^s$ is a vector with each row correspondig to an example of the profit.

Also, the mean si halved since it makes more convienent to compute the gradient descent: the derivative term will delete the $\frac{1}{2}$ term. Last but not least, $h_\Theta(x, \Theta)$ is the hypothesis function $h_\Theta : \mathbb{R}^2 \to \mathbb{R}$ and it is given by the following linear model:

$$h_\Theta(\Theta) = \Theta X$$

The Θ values are the paramater of our model and we try to find optimal values which minimise the cost function $J(\Theta)$, which is our goal. So we start with some Θ and we keep updating them until we find a minumum.

For example, to reach this minimum, we can use a gradient based algorithm which update Θ at each iteration using gradient information.

For $j = \{0, 1\}$, the gradient has the following form:

$$\frac{\partial}{\partial \Theta_0} J(\Theta) = \frac{1}{s} \sum_{i=1}^{s} (h_\Theta(\Theta^{(i)}) - y^{(i)}),$$

$$\frac{\partial}{\partial \Theta_1} J(\Theta) = \frac{1}{s} \sum_{i=1}^{s} (h_\Theta(\Theta^{(i)}) - y^{(i)})x^{(i)}$$