≡   (Optional) Closed form Solution Math

(Optional) Closed form solution math

In this optional section, we'll develop the math of the closed form solution, which we introduced in the last video. First, we'll do it for the 2-dimensional case, and then for the general case.

2-Dimensional solution

Our data will be the values $x_1, x_2, \ldots, x_m$, and our labels will be the values $y_1, y_2, \ldots, y_n$. Let's call our weights w_1, and w_2. Therefore, our predictions are $\hat{y}_i = w_1 x_i + w_2$. The mean squared error is

$$E(w_1, w_2) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$

We need to minimize this error function. Therefore, the factor of $\frac{1}{m}$ can be ignored. Now, replacing the value of $\hat{y}$, we get

$$E(w_1, w_2) \quad = \quad \sum_{i=1}^{m} (\hat{y} - y)^2$$
$$= \quad \tfrac{1}{2} \sum_{i=1}^{m} (w_1 x_i + w_2 - y_i)^2$$

Now, in order to minimize this error function, we need to take the derivatives with respect to w_1 and w_2 and set them equal to 0.

Using the chain rule, we get

$$\frac{\partial E}{\partial w_1} \quad = \quad \sum_{i=1}^{m} (w_1 x_i + w_2 - y_i) x_i$$
$$= \quad w_1 \sum_{i=1}^{m} x_i^2 + w_2 \sum_{i=1}^{m} x_i - \sum_{i=1}^{m} x_i y_i$$

and

$$\frac{\partial E}{\partial w_2} \quad = \quad \sum_{i=1}^{m} (w_1 x_i + w_2 - y_i)$$
$$= \quad w_1 \sum_{i=1}^{m} x_i + w_2 \sum_{i=1}^{m} 1 - \sum_{i=1}^{m} y_i$$

$$w_1 \left(\sum_{i=1}^{m} x_i^2\right) + w_2 \left(\sum_{i=1}^{m} x_i\right) = \sum_{i=1}^{m} x_i y_i$$

$$w_1 \left(\sum_{i=1}^{m} x_i\right) + w_2(m) = \sum_{i=1}^{m} y_i$$

We can use any method to solve 2 equations and 2 variables. For example, if we multiply the first equation by $\sum_{i=1}^{m} x_i$, the second one by m, subtract them to obtain a value for w_1, and then replace this value in the first equation, we get the following:

$$w_1 = \frac{m\left(\sum_{i=1}^{m} x_i y_i\right) - \left(\sum_{i=1}^{m} x_i\right)\left(\sum_{i=1}^{m} y_i\right)}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2}$$

$$w_2 = \frac{\left(\sum_{i=1}^{m} x_i y_i\right)\left(\sum_{i=1}^{m} x_i\right) - \left(\sum_{i=1}^{m} x_i^2\right)\left(\sum_{i=1}^{m} y_i\right)}{m\left(\sum_{i=1}^{m} x_i\right)^2 - m\left(\sum_{i=1}^{m} x_i^2\right)}$$

That's our desired solution.

n-Dimensional solution

Now, let's do this when our data has n dimensions, instead of 2. In order to do this, we'll introduce the following notation. Our matrix X containing the data is the following, where each row is one of our datapoints, and $x_0^{(i)} = 1$ to represent the bias.

$$X = \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & \cdots & x_n^{(0)} \\ x_0^{(1)} & x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(m)} & x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

Our labels are the vector

$$y = \begin{pmatrix} \vdots \\ y_m \end{pmatrix}$$

and our weight matrix is the following:

$$W = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{pmatrix}$$

And so the equation for the mean square error can be written as the following matrix product:

$$E(W) = \frac{1}{m}((XW)^T - y^T)(XW - y).$$

Again, since we need to minimize it, we can forget about the factor of $\frac{1}{m}$, so expanding, we get

$$E(W) = W^T X^T X W - (XW)^T y - y^T (XW) + y^T y.$$

Notice that in the sum above, the second and the third terms are the same, since it's the inner product of two vectors, which means it's the sum of the products of its coordinates. Therefore,

$$E(W) = W^T X^T X W - 2(XW)^T y + y^T y.$$

Now, to minimize this, we need to take the derivative with respect to all values in the matrix W. Using the chain rule, as we used above, we get the following:

$$\frac{\partial E}{\partial W} = 2X^T X W - 2X^T y$$

And in order to set this equal to zero, we need

$X^T X W - X^T y = 0$, or equivalently,

$W = (X^T X)^{-1} X^T y.$

inverse of the matrix $X^T X$ is hard, if n is large. That's why we go through the pain of doing gradient descent many times. But if our data is sparse, namely, if most of the entries of the matrix X are zero, there are some very interesting algorithms which will find this inverse quickly, and that'll make this method useful in real life.

NEXT