Linear Regression

Boyan Dafov

March 3, 2020

I Solving the 1-D problem

First we are going to define and solve the linear regression problem for one independant and one dependant variable. We are looking for the "best fitting line" for random 2D data points. The equation we are looking to build looks like that:

$$\hat{y} = a * x + b$$

Where "y" and "x" are given for every data point. Lets the fine the "Error" function, assuming that we have "n" data points given :

$$E = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \text{ where } \hat{y}_i \text{ is the prediction for } x_i$$

Now we are looking for the model, that gives us the minimum "Error function" result. First we are going to use that:

$$\hat{y}_i = a * x_i + b$$

Substitute it in the "Error function" above and get :

$$E = \sum_{i=1}^{n} (a * x_i + b - y_i)^2, \text{ where we are given } y_i \text{ and } x_i \text{ for every i}$$

To find the values for "a" and "b", we are going to take the derivatives of the function with respect first to "a" and than to "b":

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} 2 * (y_i - (a * x_i + b)) * (-x_i)$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} 2 * (y_i - (a * x_i + b)) * (-1)$$

Now we are going to simplify and set them to 0 and get the following system with two equations and two unknowns :

$$a * \sum_{i=1}^{n} x_i^2 + b * \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i * x_i$$

$$a * \sum_{i=1}^{n} x_i + b * n = \sum_{i=1}^{n} y_i$$

Let $\sum_{n=1}^{n} x_i^2 = c$, $\sum_{n=1}^{n} x_i = d$, $\sum_{n=1}^{n} x_i * y_i = e$, $\sum_{n=1}^{n} y_i = f$ and get the following system :

$$\begin{cases} a * c + b * d = e \\ a * d + b * n = f \end{cases}$$

There are many ways to solve it for both "a" and "b", but as a result we get $b = \frac{e*d - f*c}{d^2 - n*c}$ and $a = \frac{e*n - f*d}{n*c - d^2}$

II Solving the multidimensional problem

We are given set of pairs - vector of independant variables and dependant (prediction) variable $\{(x_1, y_1), (x_2, y_2), \dots\}$. Lets define the letter d - dimensionality. Our model for the multidimensional solution should look like that :

$$\hat{y} = w^T * x + b$$

Keep in mind that "w" should be "d" - dimensional vector, x is "DxN" matrix, where N - number of samples, D - number of features (dimensionality). We make the following transformations :

$\hat{y} = w^T * x + b \Leftrightarrow \hat{y} = b + w_1 * x_1 + \dots + w_d * x_d$, let $b = w_0$ and $x_0 = 1$ and get :

$$\hat{y} = w_0 * x_0 + w_1 * x_1 + \dots + w_d * x_d \implies \hat{y} = \hat{w}^T * \hat{x}$$

We define the problem like that, because it makes the calculation of multiple predictions clearer (and more optimal when it comes to computation, especially if you use library that supports matrix arithmetics like python's numpy).
Now we are going to define the "Error" function :

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{w}^T * x_i)^2$$

We still can take the derivative with respect to any component of "w" - $\hat{w}_j$, where $j = 1, \dots, d$. Let's see how does the derivative $\frac{\partial E}{\partial w_j}$ look like:

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{n} 2*(y_i - w^T * w_j)*(-\frac{\partial w^T * x_i}{\partial w_j}) = \sum_{i=1}^{n} 2*(y_i - w^T * x_i)*(-x_{ij}) = \sum_{i=1}^{n} y_i *(-x_{ij}) - \sum_{i=1}^{n} w^T * x_i *(-x_{ij})$$

Since $j = 1, \dots, d$, we set all derivatives to 0 and get "d" equations with "d" unknowns

$$\sum_{i=1}^{n} y_i * (-x_{i1}) = \sum_{i=1}^{n} w^T * x_i * (-x_{i1})$$

$$\dots$$
$$\dots$$
$$\dots$$

$$\sum_{i=1}^{n} y_i * (-x_{id}) = \sum_{i=1}^{n} w^T * x_i * (-x_{id})$$

We have that $a^T * b = \sum_{i=1}^{n} a_i * b_i \implies w^T * (X^T * X) = y^T * X$, we transpose the equation :

$$[w^T (X^T * X)]^T = [y^T * X]^T \implies (X^T * X) * w = X^T * y \implies w = (X^T * X)^{(}-1) * X^T * y$$

III R-squared value

We are going to define the R-squared value, which is used to determine how good our model is :

$$R^2 = 1 - \frac{SSres}{SStot}, \text{ where}$$

$$SSres = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

$$SStot = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

Where $\bar{y}$ is the mean value of all y_i given. When R^2 is closer to 1, our SSres is closer to 0, which means that our model is good. When R^2 is closer to 0, it means that our model predicted just the mean value of all y_i.