Regularization & Cross Validation

Friday, November 8, 2024 6:20 PM

$w^T . x + w_0$

L = (y - y_pred)^2 $= w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + w_0$

L = (y - (w^T . x + w0))^2 $= w_1 x_1 + w_2 x_1^2 + w_3 x_1^3 + ... + w_n x_1^n + w_0$

x_1	x_1^2
0.3	0.09
0.9	0.81

Here "x" are our features which will be between 0 to 1 as we have applied MinMaxScaler.
If the actual y is very large, to keep our error small, how should be the values of w^T?

Example:
Let y = 10000 so to minimize the error, y_pred should also be near to 10k
Let y_pred = 9900.
But to get y_pred = 9900, w^T should be large because x is between 0 to 1.

Now if w^T is very large, small change in 'x' will result in big change in y_pred. This leads to high variance situation. Hence it is leading towards overfit model.

What is the solution to this?
Ans: If we add some sort of penalty to our model that increases by increase in values of 'w' vector, we can control it to reach to overfit situation.

An example of our new loss function with such penalty is:

L = (y - y_pred)^2 + ($w1^2 + w2^2 + w3^2$ + …)

L = (y - y_pred)^2 + Ew^2

L = (y - y_pred)^2 + a * Ew^2

L2 Regularizer (Ridge)

Then what is L1 Regularizer?

Why can't we use |w| in place of w^2?
Ans: Because it is not differentiable!

But, is |w| not differentiable at every point?
No! It is not differentiable only at w = 0.

But the w^j is weight of a feature. If weight of feature is 0 that means it is totally useless feature & we can drop it.

At the points other than 0,
d/dx (w) = 1 for w > 0
d/dx (w) = -1 for w < 0

Therefore, we can create a loss function with |w| as well.

L = (y - y_pred)^2 + a * E |w|

L1 regularizer (Lasso)