DAC Excercise 1

Sameer Dhiman

July 20, 2020

1 Description

We are going to implement linear regression and gradient descent in this excercise. For single variable problems we already know linear regression takes the from

$$f(x) = mx + b$$

and Gradient Descent will be

$$m = m - \alpha * \frac{\delta f(x)}{\delta m} - (1)$$

$$and$$

$$b = b - \alpha * \frac{\delta f(x)}{\delta b} - (2)$$

After putting the values of derivative we get

$$m = m - \alpha * (y_{true} - y_{pred}) * x - (3)$$

$$and$$

$$b = b - \alpha * (y_{true} - y_{pred}) - (4)$$

Note: Remember that above equations (3 and 4) are in case of linear regression only. When in doubt always prefer to use to the derivative from of gradient descent (1 and 2).

Your task is to implement this in any programming language of your choice. Here is pseudocode for the above operations.

```
y_pred(m, b, x):            #Return the predicted value of y from current values of m and b
    return m*x + b

Gradient(X, Y_true, m, b):    # Will run gradient descent iteration once for each data point
    for each x in X and y_true corresponding to that x:
        m = m - alpha*(y_true - y_pred(m,b,x))*x
        b = b - alpha*(y_true - y_pred(m,b,x))

Gradient2(X, Y_true, m, b):  # may run gradient descnet iteration multiple times for each
    m_old = infiniy          # data point till the change in value of m is less than 0.001
    while(|m_old - m| > 0.01): # i.e change is very small that means we are near the minima
        m_old = m
        for each x in X and y_true corresponding to that x:
            m = m - alpha*(y_true - y_pred(m,b,x))*x
            b = b - alpha*(y_true - y_pred(m,b,x))
```

Note: I could not get the algorithm module to work properly in latex that's why I wrote the pseudocode in python (=_=) . Read it as it is written in english. This is not a proper python code.
Note 2: Here you will start with a radom value for m and b.

Now here as you can see there are multiple ways to defining the end condition for gradient descent, each one has their own benifits and shortcoming for eg. here *Gradient* will finish much faster but will have lower test accuracy, while *Gradient2* will have a higher text accuracy but can also cause overfitting.

2 Questions

Hints for these questions are given below but first try to do it without them.

1. Run the Gradient descent to find the approximate value of x for which $x^2 - 24x + 145$ is minimum.

2. Run the Gradient descent to find the approximate value of x_1, x_2, x_3 and x_4 for which $x_1^2 + 12x_1 + x_2^2 - 10x_2 + 145$ is minimum.

3. Run linear regression on the following data:

X	Y
3.0	57.0
4.4	53.8
5.8	57.6
7.3	68.6
8.7	71.4
10.1	74.2
11.5	75.0
12.9	70.8
14.4	76.8
15.8	76.6
17.2	82.4
18.6	85.2
20.1	86.2
21.5	91.0
22.9	95.8
24.3	98.6
25.7	103.4
27.2	102.4
28.6	111.2
30.0	108.0

3 Hints

1. Gradient Descent equation will be

$$x = x - \alpha * \frac{\delta(x^2 - 24x + 145)}{\delta x}$$

2. Run the same equation for all of them,

$$x_i = x_i - \alpha * \frac{\delta(x_1^2 + 12x_1 + x_2^2 - 10x_2 + 145)}{\delta x_i}$$

   for all i in {1,2,3,4}

3. I don't think you will need any.

4 Solutions

1. x = 12
2. $x_1 = -6$ and $x_2 = 5$
3. m = 2 and b = 50

Note: Your answers might not exactly match but they must be close to these answers with a tolerance of about 0.001