MLHW1

yz5946 Sky Zhou

February 2024

1 Problem 1

a

For a individual data point, the squared distance can be written as

$$(y_i - \beta * x_i) \tag{1}$$

Therefore, the sum-of-squared distance of all data points can be written as the summation of every such squared distance

$$L = \sum_{i=1}^{n} (y_i - \beta * x_i) \tag{2}$$

where y_i is the actual value of data point i and $\beta * x_i$ is the predicted y_i based on the linear model with respect to parameter β

b

To minimize the loss function, we take derivative with respect to β and find β such that the derivative is 0.

$$\frac{\partial L}{\partial \beta} = \sum_{i=1}^{n} 2 * (y_i - \beta * x_i) * (-x_i) \tag{1}$$

$$= -2 \sum_{i=1}^{n} x_i y_i + 2\beta \sum_{i=1}^{n} (x_i)^2 \tag{2}$$

Set the equation above to be 0, we can solve for the value β

$$\beta = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} (x_i)^2} \tag{3}$$

We do not get exactly the same expression as we do in the linear model. However, the only difference is, in linear model, we subtract the average of x and y from x_i and y_i (since the nominator in linear model is $\sigma_{x,y}$ and the denominator is σ_x^2). This is because the information of average is needed for a linear model to adjust the interception with y-axis. In contrast, since the given model is restricted to bypass the origin, we do not subtract the average.

2 Problem 2

a

To compute the minimum value of $L(m)$, we take derivative of the loss function with respect to m and find the value of m such that the derivative is 0.

$$\frac{\partial L(m)}{\partial m} = \sum_{i=1}^{n} -2(y_i - m) \tag{1}$$

$$= -2 \sum_{i=1}^{n} y_i + 2nm \tag{2}$$

$$m = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{y} \tag{3}$$

We can see that the value m that minimize the loss function $L(m)$ is the average of y, $\bar{y}$

b

This loss function computes the largest distance between m and all data points y_i. Therefore, to minimize this function, we should make m as close to the maximum and minimum value among y_i as possible. So m should be set to the average of the max and min of y. $m = \frac{y_{min}+y_{max}}{2}$. In this case, if m goes closer to minimum value, $|y_{max} - m|$ would become bigger and vice versa.

Setting m to be the middle point of the max and min value of y would minimize this loss function.

c

We can prove this by contradiction. Suppose we now set m to be the median of y, and now the loss function evaluates the sum of difference of distance between m and all data points of y. If this does not minimize the loss function, then we try to move m to either side (closer to max and min of y). However, no matter how we move m, we cannot let the any of $|y_i - m|$ become smaller without increasing other $|y_j - m|$. For instance, if we move m a distance ϵ towards the minimum value of y, all $|y_i - m|$ where y_i are smaller than the median of y is decreased by ϵ. However, all $|y_j - m|$ where y_j is greater than the median of y is increased by the same amount. Since the number of data points that are greater and smaller than the median is the same by the definition of median, the two effects cancelled out. Plus, the distance between the median of y and m is also increased (from 0 to ϵ). Therefore, the sum of distance is increased.

According to the proof above, there is no value of m other than the median can make the loss function smaller.

d

The performance of the 3 loss functions is dependent on the distribution of data points.

For the loss function suggest in part a), it evaluates the sum-of-square distance between the data points and their mean. This function would be suitable for data whose distribution is close to normal distribution.

For the loss function suggest in part b), it evaluates the maximum absolute deviation. This will make m is not very far from any data point, especially for extreme values. However, this loss function will be greatly affected by the extreme values. But this loss function is helpful when the extreme values must be represented. For example, if there is a cost of largest error, we can use this loss function to minimize maximum cost.

For the loss function suggested in part c), it evaluates the sum of difference between median and all data points. This would be more robust compared to the loss function in part a) in the presence of outlier and non-normal distributions. The median is less affected by the extreme values, outliers and skewed distributions, making it a better measure of central tendency.

3 Problem 3

a

We take natural log for both sides of the equation:

$$log(f(t)) = log(z_0 e^{-\alpha t}) \tag{1}$$
$$= log(z_0) + log(e^{-\alpha t}) \tag{2}$$
$$= log(z_0) - \alpha t \tag{3}$$

It is clear that $log(f(t))$ is linear with respect to t since $log(z_0)$ is a constant and α is the coefficient of t

b

1. Convert data points to their natural log forms

```
for data in y:
    data = log(data)
```

Name the converted data be y and x

2. Apply formula in linear model and run regression

```
#run regression
```

$$\alpha = -\beta_1 = -\frac{\sigma_{x,y}}{\sigma_x^2}$$
$$\log z_0 = \beta_0 = \bar{y} - \beta_1 \bar{x}$$

3. Use exponent of natural number to convert back to the proposed form

$$z_0 = exp(\log z_0)$$

Or Mathematically

$$z_0 = e^{\bar{y} + \alpha \bar{x}}$$

Finally we have

$$f(t) = e^{\bar{y} + \alpha \bar{x} - \alpha t}$$

where $\bar{y}$ is the mean of $\log y$ and $\bar{x}$ is the mean of $\log x$

However in real code we will already have a specific value for $\log z_0$ and α after finishing the linear regression. Therefore we can simply compute

$e^{\log z_0}$ to get z_0 and plug in the numbers back to the original exponential model.

4 Problem 4

a

To compute $\nabla g(z)$, we take derivative of each term of $g(z)$ with respect to their variables

$$\nabla g(z) = \begin{bmatrix} \frac{\partial g(z)}{\partial z_1} \\ \frac{\partial g(z)}{\partial z_2} \\ \vdots \\ \frac{\partial g(z)}{\partial z_n} \end{bmatrix} = \begin{bmatrix} p \cdot |z_1|^{p-1} \cdot sign(z_1) \\ p \cdot |z_2|^{p-1} \cdot sign(z_2) \\ \vdots \\ p \cdot |z_n|^{p-1} \cdot sign(z_n) \end{bmatrix} \tag{1}$$

If p is even, we can further take out the absolute sign since z_i^p will always be positive. And the information provided by the *sign* function will already be included since $p-1$ is odd, raising z_i to an odd power will contain the *sign*.

$$\nabla g(z) = \begin{bmatrix} p \cdot z_1^{p-1} \\ p \cdot z_2^{p-1} \\ \vdots \\ p \cdot z_n^{p-1} \end{bmatrix} \tag{1}$$

b

To solve of $\nabla L_p(\beta)$, we can first create a new vector t, such that $t = y - \beta\mathbf{X}$. Use the result from part a), and then apply chain rule to the

inner function t, we can derive the result.

$$\frac{\partial L_p(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} p \cdot |t_i|^{p-1} \cdot sign(t_i) \cdot -\mathbf{X_{ij}} \tag{1}$$

$$\nabla L_p(\beta) == -p \cdot \mathbf{X^T} \begin{bmatrix} |t_1|^{p-1} \cdot sign(t_1) \\ |t_2|^{p-1} \cdot sign(t_2) \\ \vdots \\ |t_n|^{p-1} \cdot sign(t_n) \end{bmatrix} \tag{2}$$

5 Problem 5

a

To show that these two models are the same, we can first compare these two functions. First, we need to show that the two pieces of function still meets at $x = \lambda$. Then, we need to show that the value of the function evaluates to the same when $x_i \geq \lambda$ comparing to the original function since the first piece of the function appears to be the same.

$$a_1 + s_1\lambda = a1 + s_1\lambda - s_2\lambda + s_2\lambda \tag{1}$$

Therefore, when $x = \lambda$, the two pieces indeed meet. Then, to prove the next part, we need the information from the original function. We have

$$a_1 + s_1\lambda = a_2 + s_2\lambda \tag{1}$$

Therefore,

$$a_1 = a_2 + s_2\lambda - s_1\lambda \tag{2}$$

Substitute a_1 to the given model, we have

$$a_1 + s_1\lambda - s_2\lambda + s_2x_i = a_2 + s_2x_i \tag{3}$$

So the given model is the same as the original model as the function evaluates to be the same for all x_i in its domain.

b

1. Since there are 3 parameters to be solved, we need 3 columns for X. The first parameter should be the y-intercept of the first part of the piece-wise function. The second part should be the slope of the first part of the piece-wise function. The third part should be the slope of the second part of the piece-wise function.

1.0 Before data transformation, I would first sort the data based on X variable

1.1 Therefore, the first column of X will be a column of 1's. In this case the solved β would contain information of t-intercept as a constant.

1.2 The second column of X will be the column of original data.

1.3 The third column of X will be: for any value that is greater than λ, subtract λ; for any value that is smaller than λ, set it to 0. This column would compute a change in slope for values that are greater than λ.

1.4 Extract corresponding data to be Y.

2. Perform Multiple Regression Algorithm with squared loss to compute β

3. The coefficient β should be in this form:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \tag{1}$$

Where β_0 corresponds to y-intercept, which is a_1. β_1 corresponds to s_1. And $\beta_2 + \beta_1$ is s_2 since the second column is the change in slope for x_i greater than λ.

c

See attached notebook.

d

See attached notebook.