Multiple Regression

Created	@Jan 30, 2021 11:48 AM
Created By	K Khanh Vương
Last Edited By	K Khanh Vương
Last Edited Time	@Feb 13, 2021 11:49 AM
Module	Regression
Status	Archived
Type	Regression Problem

How can we determine whether a model is Linear ?

- A **Linear Regression** model is always linear in the `parameters`, but may use non-linear `feature`

- An example of a **NOT Linear Model**

$$y = w_0 w_1 log(w_1) x$$

What is Multiple Regression ?

- **Multiple Regression** is **Linear Regression** when we have multiple `feature`
- Each `feature` is a function of a single or multiple `input`

$$y = \sum_{i=0}^{N} w_i h_i(x) + \epsilon_i$$

  - x_i are those `input`
  - $h_i()$ are the `function` of those `input` or `feature`
  - w_i are the **Regression Coefficient** or **Parameter**

What is the form of Polynomial Regression ?

$$y = \sum_{i=0}^{N} w_i x^i + \epsilon$$

- **Features**:

$$[1(constant), x, x^2, ..., x^N]$$

- **Parameters**:

$$[w_0, w_1, w_2, ..., w_N]$$

- **Polynomial Regression** is a form of regression analysis in which the relationship between the `Independent Variable` and the `Dependent Variable` is modeled as an n^{th} degree polynomial in `Independent Variable`
- **Polynomial Regression** fits a nonlinear relationship between the value of x and the corresponding conditional $mean$ of y

What is the Seasonality ?

- **Seasonality** is a term that describe the up-and-down of the y in a specific range of x

- We can easily generate **Seasonality** through sin and cos function and treat them like `parameter` of the model
- We can combine **Seasonality** to our **Linear Regression Line** to make the line become more flexible and easily show the trend of the data

Why should we use Multiple Regression ?

- **Simple Linear Regression** tend to be too simple to describe the data
- So even if we add **Polynomial Feature** to our model, we may not archive better result due to the fact that there're maybe some other `features` that may affect the result
- So we must add those `feature` to our model to archive there's affect on the result. By adding them, we increased the number of dimensions of our model

What are the general notation ?

Notation

Name	Notation	Type
Output	y	Scalar
Output of the i Observation	y_i	Scalar
Input of the i Observation	$x_i = (x[0], x[1], ..., x[d])$	Vector
Value of the j Feature	$h_j(x_i)$	Scalar
Value of the j input of the i Observation	$x_i[j]$	Scalar

What is the equation for Regression with Features of Multiple inputs ?

$$y_i = \sum_{j=0}^{D} w_j h_j(x_i) + \epsilon_i$$

- $h_j(x_i)$ is the j `feature` of the `input`
- w_j is the **Regression Coefficient** or **Weight** associated with that `feature`.
- The **Regression Equation** represents a (hyper)plane in a $k+1$ dimensional space in which k is the number of `Independent Variables` plus one dimension for the `Dependent Variable` Y.

How can we Interpreting the Coefficient of the fitted function ?

- When we think about interpreting any given `coefficient` of our fitted model, we're gonna fix all the other `input` in the model, and just gonna look at that one that we can vary
- We will then see how much this `feature` affect our model, mean the impact of this `feature` on the `output` when other `feature` are fixed
- But one thing to be noticed is that, we must consider the context of the model. Think about the `coefficient` and the context of what we've put into the model. Sometime, the model that contain that `coefficient` with some other `sub-features` can behave very different from the model that doesn't have those `sub-features`

Can we interpret the Coefficient in a Polynomial Regression model ?

- We can't interpret the `coefficient` in a **Polynomial Regression Model** because if we want to interpret, the first thing we need to do is to fixed all other `coefficient` in the model. But in **Polynomial Regression Model**, everything else is a power of this one `input` so we can't do that because if we change one `input`, then we can't hold all other `features` fixed

How can we rewrite our model in Matrix Notation ?

$$Y = WH(X_i)$$

- Y is the output vector
- W is the `parameter` vector
- $H(X_i)$ is the `feature` vector

What is the Cost Function of Multiple Regression ?

$$RSS(W) = \sum_{i=1}^{N}(y_i - h^T(X_i)W)^2$$

- We can then rewrite this function in **Matrix Notation**

$$RSS(W) = (Y - HW)^T(Y - HW)$$

  - By multiply the **Residual** vector with the transposed of itself, we will get $RSS(W)$, which is a **Scalar** that emphasize the **Residual Sum of Square** for N observations

What is the Gradient of the Cost Function ?

$$\bigtriangledown RSS(W) = -2H^T(y - HW)$$

- To interpret this derivative equation, think of the corresponding equation for the loss function in 1-dimensional space.

$$\begin{aligned} \tfrac{\mathrm{d}}{\mathrm{d}w}(y - hw)(y - hw) &= \tfrac{\mathrm{d}}{\mathrm{d}w}(y - hw)^2 \\ &= 2(y - hw)(-h) \\ &= -2h(y - hw) \end{aligned}$$

What is the close-form approach for estimating the parameter of the Multiple Regression Problem ?

- We can simply set the **Gradient** of our **Cost Function** equal 0 to estimate the W

$$
\begin{aligned}
\bigtriangledown RSS(W) &= -2H^T(y - HW) = 0 \\
&\leftrightharpoons -2H^Ty + 2H^THW = 0 \\
&\leftrightharpoons H^THW = -H^Ty \\
&\leftrightharpoons W = (-H^Ty)(H^TH)^{-1}
\end{aligned}
$$

  - Note that: $AA^{-1} = I$
  - But there will be some problem with this type of approach
    - **Invertible Problem**: As we see here, this approach will require us to find the **Inverse** of the H^THW term
      - We can only take the **Inverse** of DD matrix if $N > D$, mean the number of **Linearly Independent Observations** that needs to be greater than the number of `features`
    - **Complexity of Inverse**: $O(D^3)$

What is the Gradient Descent approach for estimating the parameter of the Multiple Regression Problem ?

- While the **Function** is not **Converged**:

$$
W^{(t+1)} = W^t - \eta \bigtriangledown RSS(W^t) = W + 2\eta H^T(Y - HW^t)
$$

How can we interpret the update of each feature in the Matrix Form ?

- As we know

$$RSS(W) = \sum_{i=1}^{N}(y_i - h^T(X_i)W)^2$$
$$= \sum_{i=1}^{N}(y_i - h_0(x_i)w_0 - h_1(x_i)w_1 - ... - h_D(x_i)w_D)^2$$

- In the **Matrix Notation**, we will update these all `feature` at once by subtracting the `paramater` vector by the **Gradient** of the **RSS**. So split the **Gradient** out, we will have the partial derivate with respect to each of those `feature`

$$\frac{\partial}{\partial w_j} RSS = -2\sum_{i=1}^{N} h_j(x_i)(y_i - h^T(X_i)W)$$

- So each time we update the j^{th} `feature`

$$w^{(t+1)} = w^t + 2\sum_{i=1}^{N} h_j(x_i)(y_i - h^T(X_i)W)$$

- If we under estimate that j^{th} `feature`, $y_i - h^T(X_i)W$ will be `positive,` then the `Coefficient` w_j with respect to that `feature` will be increased, or will be decreased on the opposite

How can we apply Multiple Regression in actual project ?

- Notice that we must also create `training data` for those `features`
- For example, if we want to add one special `feature` like `a * b`, then we will need to add this type of result to the `training set` and `test set`