Simple Linear Regression

Created	@Jan 21, 2021 1:32 PM
Created By	K Khanh Vương
Last Edited By	K Khanh Vương
Last Edited Time	@Feb 12, 2021 8:01 PM
Module	Regression
Status	Archived
Type	Regression Problem

What is Regression ?

- A subset of **Supervised Learning**
- Used for
    - Forecasting by predicting a **Continuous Value** from the `input`
    - Indicates the **strength of impact** of multiple `Independent Variable` on a `Dependent Variable`

How can we distinguish between Regression techniques ?

- Number of `Independence Variables`
- Relationship between the `Independent` and `Dependent` variables.

What are the differences between Simple and Multiple Regression ?

- **Simple Regression**:
    - Have only one `Independent` and one `Dependent` variables.
    - The shape of the line is `straight`
- **Multiple Regression**:
    - Have multiple `Independent Variable`
    - The shape of the line can be more complex

- This will result in a `hyperplane` base on the number of the `features`

What are the constraints of Dependence and Independence Variable ?

- **Independence Variable**:
    - Can be `discrete` or `continous`
- **Dependence Variable**:
    - Must be `continous`, or else it would be `Classification` or `Logistic Regression` problem
        - **Explanation**: Because we will want the output of the model is a line, so it must be `continous`

What is the equation for Simple Linear Regression ?

$$y_{(W)} = w_0 + w_1 x$$

- With $W(w_0, w_1)$ is the `parameter`, the `weight`, or the `coefficient` of our model
    - w_0 is the `intercept`, represent the value of y when $x = 0$
    - w_1 is the `slope`, represent the impact of x on y, means how much `Dependent Variable` will change when we change the value of `Independent Variable`

How can we change the shape of the Regression Line to fix more complex data ?

- If we add more `feature` to our `model`, meaning add more `coefficient`, then our `model` will be increased in the number of dimensions

$$y_{(W)} = w_0 + w_1x_1 + w_2x_2 + ...$$

- If we want our `model` to fit more complicated `data`, then we can add `polynomial` features, meaning increase the flexibility of the curve

$$y_{(W)} = w_0 + w_1x_1 + w_2x_2^2$$

What is the Bias-Variance Trade off ?

- **Simple Model**
  - Well-behave
  - Too simple to describe the complex relationship
- **Complex Model**
  - Flexible, so it will have the potential to fit really complex relationship that describe really what happen with the data
  - Can have strange behaviors
- **Bias-Variance trade off** is the progress that consider the trade off between two of them to get the best model

What will happen with the `Dependent Variable` when we scale the `Independent Variable` ?

- The `intercept` will always remain the same
- The magnitude of the `slope` will change base on the correlation between new and old unit
  - If we change the scale of the `Independence Variable`, then the **Regression Line** will become longer

- If we change the scale of the `Dependence Variable`, then the slope of the **Regression Line** will be increased with the factor of the `slope`

How can we make use of the Regression Line ?

- Every `prediction` will lie on the **Regression Line**, but will have some `error`. Meaning that

$$y_{actual} = y_{predict} + \epsilon$$

  - With ϵ is the `error coefficent`
  - We will want our $\epsilon = 0$
- Because we won't know the **Regression Line** will lie above or below the actual value for each `data point`, so the $y_{predict}$ will be the best `guess` that we can generate

What is the main task over a Simple Linear Regression model ?

- We will want to optimize our model through optimize it's `parameter`
  - The model can be fully described by it's `parameter`
  - We will have fixed x and y, meaning that we will already have the `data` and `label`
  - Each pair of (x, y) will be a `data point`, and will be described by the model through $W(w_0, w_1)$
- Meaning that we will need to optimize the `Loss Function` by optimizing the `parameter`

What is Convex and Concave function ?

- **Concave** is a function where the line that connect two random points on the curve lie below the curve everywhere
- **Convex** is a function where the line that connect two random points on the curve lie above the curve everywhere
- The place where the derivate of both **Concave** and **Convex** function equal to 0 are **Unique**

What is the Hill-Climbing Algorithm ?

- We will check while our function is not converged

$$w^{(t+1)} = w^{(t)} + \eta \frac{\mathrm{dg(w)}}{\mathrm{dw}}$$

  - η is the step size
  - In case $value > 0$, then we need to move the point to the right side of the graph (increase the value)
  - In case $value < 0$, then we need to move the point to the left side of the graph (decrease the value)

What is the Hill-Descent Algorithm ?

- We will check while our function is not converged

$$w^{(t+1)} = w^{(t)} - \eta \frac{\mathrm{dg(w)}}{\mathrm{dw}}$$

  - η is the step size
  - In case $value < 0$, then we need to move the point to the right side of the graph (increase the value)
  - In case $value > 0$, then we need to move the point to the left side of the graph (decrease the value)

What is the property of a Minimum and Maximum ?

- These points are always **Unique**
- **Gradient Descent** and **Gradient Ascend** will converge at these point

How can we distinguish between a Loss Function and an Objective Function ?

- A **Loss Function** or **Cost Function**: Our problem (which is an optimization problem) seeks to `minimize` a loss function.
- An **Objective Function** is either a loss function or its negative, in which case it is to be `maximized`.

What is the Cost Function of Linear Regression problem ?

- Used to evaluate our `model`
- Will return a number that emphasize the different between $y_{predict}$ and y_{actual}
    - **Residual Sum of Square**:

    $$RSS(w_0, w_1) = \sum_{i=1}^{N}(y_i - [w_0 + w_1 x_i])^2$$

    - **Mean Square Error**:

    $$J(w_0, w_1) = \frac{1}{2\text{n}} \sum_{i=1}^{n}(y_i - [w_0 + w_1 x_i])^2$$

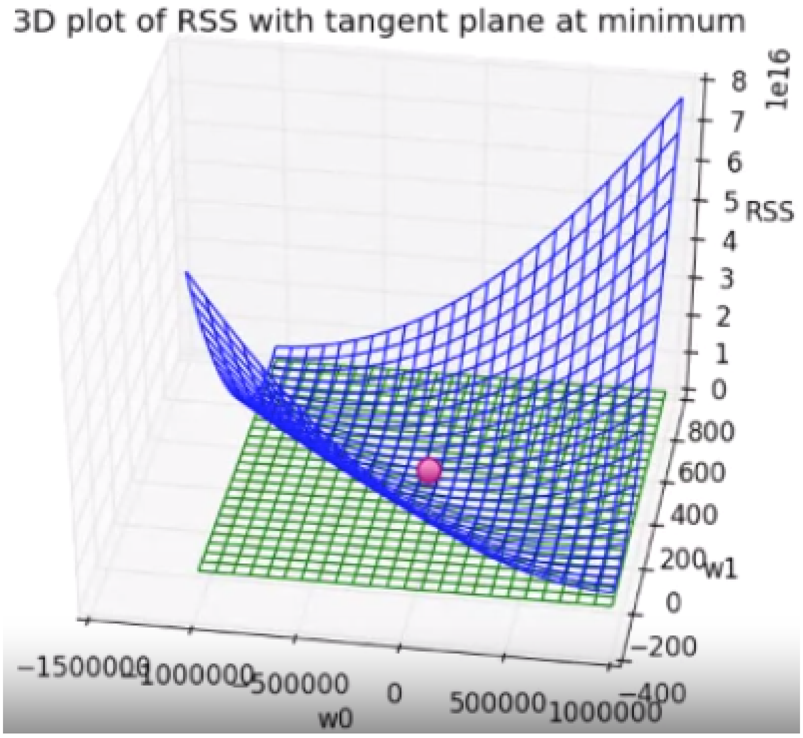
- We will want to optimize the `Cost Function`, meaning return the line with lowest error

How can we explain the Mean Square Error function ?

- Notice that, in the function, we will have two term:
    - y_i will be the actual value
    - $[w_0 + w_1 x_1]$ will be the predicted value
- So our function will compute the difference between `predicted value` and `actual value`
- We will then square these `difference` to:
    - Avoid the elimination between `positive` and `negative` value (`residual`)
    - The derivate of a quadratic function will be easier to manipulate than absolute value equation
- Sum over all of these squared values, then divide it by n to find it's `mean`. This will remove the dependence of our function on the number of example
- Finally, we divide it by 2 to make future operations simpler

How can we describe the RSS with respect to W ?

- In space, RSS will act as a plane or a hyperplane
- There will be a place where the $Gradient = 0$, and our task is to determine that place
- $W(w_0, w_1)$ helps us determine the place, means the co-ordinate of the RSS graph where the $Gradient = 0$



What is the shape of the Minimizing MSE or RSS Problem ?

- Because this problem refer to minimizing the `Cost Function`, so it will rather be a `Convex Function`
- Notice that, for **Simple Linear Regression**, it will always be **Convex** because our function is a quadratic equation, so it's derivate will be a parabolic shape with only one inflection point (`minimum`)

When should we use RSS and MSE ?

- **RSS** is the sum of all `error` on the dataset
- **MSE** is the **Squared** of `error`, show how much the `error` on a single point
- When we want to compare `train error` with `test error`, we should use **MSE** because the size of the `test data` always smaller than the size of `train data` . If we use **RSS**, then there will be a huge difference between these two.
- We can use **RSS** to compare between models of the same dataset

What is the Grad of a Function ?

- **Grad** is a **Vector** that contain partial derivative with respect to each variable. So with a **Function** with n variables, **Grad** will be a **N-Dimension Vector**
- **Gradient** has many properties, and if we draw lines or planes between them, then it will form a **Vector**
- **Grad** is a **Vector** that point to the steepest uphill slope.

What are the Derivative of the Intercept and the Slope ?

- The derivative of the cost for the `intercept` is the sum of the `errors`
- The derivative of the cost for the `slope` is the sum of the product of the `errors` and the `input`

What is the algorithm of Gradient Descent ?

- In each step of the **Gradient Descent** we will do the following:
  - Compute the predicted values given the current `slope` and `intercept`
  - Compute the prediction `errors`
  - Update the `intercept`:
    - Compute the derivative: `sum(errors)`
    - Compute the adjustment as step_size times the derivative
    - Decrease the intercept by the adjustment
  - Update the `slope`:
    - Compute the derivative: `sum(errors*input)`
    - Compute the adjustment as `step_size` times the `derivative`

- Decrease the `slope` by the adjustment
- Compute the magnitude of the `gradient`
- Check for `convergence`

- *Reference: https://www.coursera.org/learn/ml-regression/supplement/TFq2w/optional-reading-worked-out-example-for-gradient-descent*

How can we optimize the Loss Function ?

- We will use **Gradient Descent** to optimize our **RSME** as a **Convex Problem**, we will keep decreasing the `cost` until we reach the `minimum`
- So while our function is not `converged`, we will update our `parameter` base on the **Gradient** of our function. So first, we will take the **Gradient** of our **Loss Function**, which is a **Vector** that contain that `partial derivate` with respect to each `variable`
  - **RSS Approach**:

$$RSS(w_0, w_1) = \sum_{i=1}^{N}(y_i - [w_0 + w_1 x_i])^2$$

$$\bigtriangledown RSS(w_0, w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - (w_0 + w_1 x_i)] \\ -2\sum_{i=1}^{N}[y_i - (w_0 + w_1 x_i)]x_i \end{bmatrix}$$

  - **RSME Approach**:

$$J(w_0, w_1) = \frac{1}{2\text{n}}\sum_{i=1}^{n}(y_i - [w_0 + w_1 x_i])^2$$

$$\frac{\partial}{\partial w_i} J(w) = \frac{\partial}{\partial w_i}\frac{1}{2}([w_0 + w_1 x_i] - y_i)^2$$

$$= ([w_0 + w_1 x_i] - y_i)x_i$$

- Then we will use **Gradient** to adjust our `parameter`
  - **RSS Approach**:

$$\begin{bmatrix} w_0^{t+1} \\ w_1^{t+1} \end{bmatrix} = \begin{bmatrix} w_0^{t} \\ w_1^{t} \end{bmatrix} + 2\eta \begin{bmatrix} \sum_{i=1}^{N}[y_i - [w_0 + w_1 x_i] \\ \sum_{i=1}^{N}[y_i - [w_0 + w_1 x_i]x_i \end{bmatrix}$$

  - **RSME Approach**:

$$\begin{bmatrix} w_0^{t+1} \\ w_1^{t+1} \end{bmatrix} = \begin{bmatrix} w_0^{t} \\ w_1^{t} \end{bmatrix} - \frac{1}{\mathrm{n}}\eta \begin{bmatrix} \sum_{i=1}^{N}[w_0 + w_1 x_i] - y_i \\ \sum_{i=1}^{N}([w_0 + w_1 x_i] - y_i)x_i \end{bmatrix}$$

- Notice that since w_0 is just a `constant`, it's derivate is just 0, so there is no x_i term at the end of it's **Gradient**
- If we underpredicting y_{pred_i}, then $\sum_{i=1}^{N}[y_i - y_{pred_i}(w_0, w_1)]$ will be `positive`, so w_0 will be increased
- Similar to w_1, increase by multiply with x_i

How can we choose the magnitude for the Step Size or Learning Rate ?

- **Step Size** is a `hyperparameter`, which we must specify
- We can think of choosing a **fixed constant Step Size**, but this not optimized
  - If we decide to take one step at a time, we would eventually reach the bottom of the pit but this would take a longer time.
  - If we choose to take longer steps each time, we would reach sooner but, there is a chance that we could overshoot the bottom of the pit and not exactly at the bottom
- The common choices is to decrease the step size as the number of iteration increase

$$n^{(t+1)} = \frac{\alpha}{\mathrm{t}}$$

$$n^{(t+1)} = \frac{\alpha}{\sqrt{\mathrm{t}}}$$

When will the Gradient or Derivate be Converged ?

- The value of the `Derivate` or the magnitude of the `Gradient` equal 0
- We must notice that, the result will **Never** be 0, so we will accept some points that nearly ϵ, which is called `threshold`

$$\left|\frac{\mathrm{dg(w)}}{\mathrm{dw}}\right| < \epsilon \text{ or } \left\|\bigtriangledown g(x)\right\| < \epsilon$$

- In practice, ϵ will be very small, depend on the data:
  - What the form of this function is
  - What are the range of gradients we might expect
  - Is the value of the function, a plot of the value of the function over iterations? And we will tend to see that the value decrease, and it's basically not changing very much.

How can we compare between to close-form approach and Gradient Descend approach ?

- For most of the case, we will find it difficult or impossible to solve $Gradient = 0$
- **Gradient Descent** is more efficient. However, it will rely on choosing the right `stepsize` and `convergence` criteria

What are the equations to compute the Regression Line ?

- **Close-form Approach**:
  - **Sum Method**:

$$slope = \frac{\mathrm{sum(XY)} - \frac{1}{\mathrm{N}} sum(X) sum(Y)}{\mathrm{sum(X)^2} - \frac{1}{\mathrm{N}}\mathrm{sum(X)sum(X)}}$$

  - **Mean Method**:

$$slope = \frac{\mathrm{mean(XY)} - mean(X) mean(Y)}{\mathrm{mean(X)^2} - \mathrm{mean(X)mean(X)}}$$

  - **Intercept:**

$$intercept = mean(Y) - slope.mean(X)$$

How can we distinguish between Outliers and High Leverage Observations ?

- An **outlier** is a data point whose response y does not follow the general trend of the rest of the data. Or we can say that it is `extreme` with respect to the other y values, not the x values.
- A data point has high **leverage** if it has `extreme` predictor x values.
    - With a single predictor, an extreme x value is simply one that is particularly high or low.
    - With multiple predictors, extreme x values may be particularly high or low for one or more predictors, or may be `unusual` combinations of predictor values (e.g., with two predictors that are positively correlated, an unusual combination of predictor values might be a high value of one predictor paired with a low value of the other predictor).

What is the Influential Points ?

- A data point is **Influential** if it unduly influences any part of a **Regression Analysis**
    - The predicted responses y
    - The estimated `slope` coefficients
    - The `hypothesis` test results
- **Outliers** and **High Leverage** data points have the potential to be **Influential**, but we generally have to investigate further to determine whether or not they are actually **Influential**.

What are the effects of High Leverage Observations and Outliers ?

- Due to **Loss Function (RSS/RMS)**, the error rate of each training point is **Squared.**

- **Outliers** with vastly different values will pull the fitted line toward them, even though it is favoring one point over many others
- **Outliers** on the y axis will be easily controlled by other points
- However, **High Leverage Observation** on the x axis tend to be easily impact the trend of the line because it can easily pull out the line toward it

How can we determine whether a data point is Influential ?

- Find the best fitting line twice
    - Once with the suspect **Influential** data point included
    - Once with the suspect **Influential** point excluded
- Then compare the `intercept` and the `slope` of those two lines