convergence algorithm:

$$\underline{J(\theta_0, \theta_1)} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)} \right)^2$$

cost function

mean square Error

$h_\theta(x)^{(i)}$ → predicted

$y^{(i)}$ → actual or truth point

m = no. of datapoint.

difference between cost function and loss function.

cost function: we find error for all the points and take average of it.

Loss function: we find error for observed points and if we find error for all the points then it is loss function.

$$\text{loss function} := \left(\underline{h_\theta(x)^{(i)}} - (y)^{i} \right)^2$$

$h_\theta(x)^{(i)}$ → predicted value

$(y)^{i}$ → actual value

$$= (\hat{y}_j - y_j)^2$$

* for every single point we need to find loss function.

To achieve global minima,

derivative w.r.t ~~to~~ θ_0, $j=0$

$$\frac{\partial}{\partial\theta_0} J(\theta_0,\theta_1) = \frac{\partial}{\partial\theta_0}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2\right\}$$

$$\cancel{= \frac{1}{2m}} \qquad h_\theta(x) = \theta_0+\theta_1 x$$

$$= \frac{\partial}{\partial\theta_0}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left[(\theta_0+\theta_1 x)^i - y^i\right]^2\right\}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left((\theta_0+\theta_1 x)^i - y^i\right)\times 1$$

$$\frac{\partial}{\partial x}\left((1+x)-y\right)^2$$

$$= \underline{2[(1+x)-y]\times 1}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left((\theta_0+\theta_1 x)^i - y^i\right)$$

derivative w.r.t. θ_1, $j=1$

$$\frac{\partial}{\partial\theta_1} J(\theta_0,\theta_1) = \frac{\partial}{\partial\theta_1}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2\right\}$$

$$= \frac{\partial}{\partial\theta_1}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left((\theta_0+\theta_1 x)^i - y^i\right)^2\right\}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left((\theta_0+\theta_1 x)^i - y^i\right)x.$$

Repeat untill convergence

{

$$\theta_0 := \theta_0 - \alpha \cdot \frac{1}{m} \cdot \sum_{i=1}^{m} \left(h_\theta(x)^i - y^i \right)^2$$

$$\theta_1 := \theta_1 - \frac{\alpha}{m} \sum_{i=1}^{m} \left(h_\theta(x)^i - y^i \right) x^{(i)}$$

}

* learning rate: speed of convergence

Types of cost function:

1. MSE: mean square error
2. MAE: mean absolute error
3. RMSE: root mean square error.

Mean Square Error

actual.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y - \hat{y} \right)^2 \qquad \therefore \hat{y} = \theta_0 + \theta_1 x$$

predicted.

Root mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2}$$

adv:
- It is differentiable.
- unit remain same.

disadv:
- Not robust to outliers.

Huber Loss Function

The Huber loss offers the best of both worlds by balancing the MSE and MAE together.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y-f(x))^2 & \text{for } |y-f(x)| \le \delta, \\ \delta|y-f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

It says: for loss values less than delta, use the MSE, for loss values greater than delta, use the MAE.

using the MAE for larger loss values mitigates the weight that we put on outliers so that we still get a well-rounded model.

At the same time, we use the MSE for the smaller loss values to maintain a quadratic function near the centre.

This has the effect of magnifying the loss values as long as they are greater than 1.

Once the loss for those data points dips below 1, the quadratic function down-weights them to focus the training on the higher-error data points.

Note: use the Huber loss any time you feel that you need a balance between giving outliers some weight, but not too much.

How can we check if a model is good or not?
→ using performance metrics.

Performance metrics

1. R-squared
2. Adjusted-R squared

R-squared : measures the performance of the model.