Regression

Credit rating (CIBIL score):

1. Age
2. Income
3. past credit statements
4. Location
5. Assets/colla..

age = 20 income = past credit statements = good/excellent/bad/avg/poor

excellent = 5
good = 4
avg = 3
bad = 2
poor = 1

CIBIL score = w1* age + income * 0.1 + past credit history * 1 = 750

CIBIL score = age * x + x0
y = mx + c

training set -> learning algo/model -> hypothesis

x -> hypothesis -> predicted ‘y’ ~ around the actual numbers

< (actual - predicted)

regression: with one variable

hypothesis function:
h(theta) = theta0 + theta1 * x

Training (x) say age	Actuals(y) CIBIL score
20	700
22	650
25	720

h(theta) <- training set and predicts

here m = 3 since we have only 3 values on our training set

Cost function = 1/2m sumOfAllValInRange (1, m) (h(theta) - (y)^2
reduce your cost function

Training set $(x_1, x_2, x_3 \ldots x_m)$

↓

learning algorithm

↓

Test data → hypothesis $h_\theta(x)$ → prediction

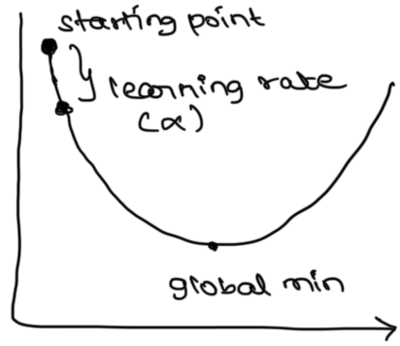
one variable hypothesis function

$$h_\theta(x) = \theta_0 + \theta_1 x$$

cost function

$$J(\theta_0, \theta_1) = 1/2m \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}]^2$$

goal: minimize $J(\theta_0, \theta_1)$



Gradient descent algorithm

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

where

α - learning rate

$j = 0, 1$

smaller 'α' will lead to smaller step size, in other words we would have to make more number of small steps to find global minimum.

larger 'α' would mean we make bigger steps and we might potentially go past global minimum.

substitute linear Regression model in Gradient descent algorithm

$\partial \quad J(\theta_0, \theta_1) \qquad m \qquad (i) \qquad (i)]^2$

$$\frac{\partial}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\left[\frac{1}{2m}\sum_{i=1}[h_\theta(x$$

sub

$h_\theta(x) = \theta_0 + \theta_1 x$ in above equation

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

at $j=0$

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

$$= 1/m \sum_{i=1}^{m} \underbrace{\theta_0 + \theta_1(x^{(i)})}_{h_\theta(x^{(i)})} - y^{(i)}$$

at $j=1$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = 1/2m \sum_{i=1}^{m} 2[\theta_0 + \theta_1(x^{(i)}) - y^{(i)}] * x^{(i)}$$

$$= \frac{1}{m}\sum_{i=1}^{m}[\underbrace{\theta_0 + \theta_1(x^{(i)})}_{h_\theta(x^{(i)})} - y^{(i)}]\ x^{(i)}$$

gradient descent algorithm for linear regression model

$$\boxed{\begin{array}{l} \text{Repeat until convergence} \\ \theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}] \\ \theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]\ x^{(i)} \end{array}}$$

multi-variable linear regression

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

e.g.)

$$h_\theta(x) = 80 + 0.1x_1 + 0.01x_2 + 3x_3 - 2x_4$$

age of

size of house   # rooms   ...   house

house price   # floors

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{(n+1)\times 1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}_{(n+1)\times 1}$$

Here
x, θ are Row vectors

where $x_0 = 1$ always.

$$\theta^T = [\theta_0 \quad \theta_1 \quad \theta_2 \quad \dots \quad \theta_n]_{1\times(n+1)}$$

column vector

$$\boxed{h_\theta(x) = \theta^T x}$$

gradient descent for multi-variable

$$J(\theta_0, \theta_1, \dots \theta_n) = 1/2m \sum_{i=1}^{m} [h_\theta(x) - y]^2$$

$$= 1/2m \sum_{i=1}^{m} [\theta^T x^{(i)} - y^{(i)}]^2$$

when $n = 1$

$$\theta_0 := \theta_0 - \alpha \underbrace{1/m \sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)}}_{\frac{\partial}{\partial\theta_0}}$$

$$\theta_1 := \theta_1 - \alpha \underbrace{1/m \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x^{(i)}}_{\frac{\partial}{\partial\theta_1}}$$

Important note:
update θ_0, θ_1 simultaneously

For $n \geq 1$

$$\dots \alpha\, 1/ \dots \sum^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_j^{(i)}$$

$$\theta_j := \theta_j - \alpha \, 1/m \sum_{i=1}$$

where $j = 0, 1, 2 \ldots n$

now $$\theta_0 := \theta_0 - \alpha \, 1/m \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] \underbrace{x_0^{(i)}}_{x_0^{(i)} = 1 \text{ initial assumption}}$$

$$\theta_1 := \theta_1 - \alpha \, 1/m \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \, 1/m \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_2^{(i)}$$

In general $$\sum_{i=1}^{m} \theta_i x_i = y_i$$

$$X\theta = Y$$

$$\boxed{\theta = X^{-1} Y}$$

$$\theta \simeq \underbrace{(X^T X)^{-1} X^T}_{\text{Pseudo inverse}} Y$$

Example: House price estimator

x0	size of house (ft) x1	#rooms(x2)	#Floors (x3)	Age of house in months(x4)	Y (House price in *1000)
1	2100	5	1	45	460
1	1400	3	2	40	232
1	1500	3	2	30	315
1	850	2	1	36	178

x = [x0 x1 x2 x3 x4]

$$X = \begin{bmatrix} 1 & 2100 & 5 & 1 & 45 \\ 1 & 1400 & 3 & 2 & 40 \\ 1 & 1500 & 3 & 2 & 30 \\ 1 & 850 & 2 & 1 & 36 \end{bmatrix}$$

$$Y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$\theta = X^{-1} Y$$

$$= (X^T X)^{-1} X^T Y$$