Logistic Regression

Linear regression:-

$$\hat{y} = h_\beta(x) = \sum_{i=0}^{N} \beta_i x_i$$

→ objective :- Minimise the sum of squared error (SSE)

→ Used for regression problem

Classification:-

Linear function + Another function ⇒ classification

i.e) $\hat{y} = g(h_\beta(x))$ where) $g(z) = \frac{1}{1+e^{-z}}$

Value between 0 and 1

<0.5 ≥ 0.5

$\hat{y} = h_\beta(x) \Rightarrow g(\beta^T x)$

$\Rightarrow \frac{1}{1+e^{-\beta^T x}}$ i.e) sigmoidal function

Note:- $g(z) = 1$ if $z = \infty$

$g(z) = 0$ if $z = -\infty$

$$g'(z) = \frac{1}{1+e^{-z}} \times e^{-z}$$

$$= \frac{1}{1+e^{-z}} \times \left(1 - \frac{1}{1+e^{-z}}\right)$$

$$\left.\begin{aligned} &= g(z) * (1-g(z)) \\ &\text{(or)} \\ &h_\beta(x) - (1-h_\beta(x)) \end{aligned}\right\} \text{---} \ 1$$

* A single trial
* $P(\text{success}) = p$

$P(\text{failure}) = 1-p$.

Likelihood of a single observation for p given x and y:

$$L(p_i / y_i) = p(y_i = y_i) = p_i^{y_i} (1-p_i)^{1-y_i}$$

Given the observations are independent,

$$L(P/y) = \prod_i p(y_i = y_i) = \prod p_i^{y_i} (1-p_i)^{(1-y_i)}$$

Log likelihood turns product into sums.

$$l(P/y) = \sum_i y_i \log(p_i) + (1-y_i) \log(1-p_i)$$

$$= \sum_i y_i \log(\hat{y}) + (1-y_i) \log(1-\hat{y})$$

Take derivatives of $l(P/y)$ to maximize log likelihood

$$\text{New } \beta = \text{old (or) initial } \beta + \alpha \nabla_\beta l(P/y)$$

$$\frac{\partial l(P/y)}{\partial \beta} = \left[y \left(\frac{1}{\hat{y}}\right) + (1-y) \times \frac{-1}{1-\hat{y}} \right] \times \frac{\partial}{\partial \beta}(\hat{y})$$

$$= \left[\frac{y}{g(\beta^T x)} + (1-y) \times \frac{-1}{1-g(\beta^T x)} \right] \times \frac{\partial}{\partial \beta} g(\beta^T x)$$

$$= \left[\frac{y}{g(B^T x)} - (1-y) \times \frac{1}{1-g(B^T x)} \right] \times g(B^T x)\,(1-g(B^T x))\,\frac{\partial}{\partial B}(B^T x)$$

②

$$= \frac{\left[y\,(1-g(B^T x)) \right] - \left[(1-y) \times g(B^T x) \right]}{g(B^T x)\,(1-g(B^T x))} \times g(B^T x)\,(1-g(B^T x)) * x$$

$$= \left[y - y\,g(B^T x) - g(B^T x) + y\,g(B^T x) \right] * x$$

$$= \left[y - g(B^T x) \right] * x$$

$$= (y - h_\beta(x)) * x \longrightarrow ②$$

$$\boxed{\therefore \quad \beta_j = \beta_j + \alpha * y - h_\beta(x) * x_j} \longrightarrow ③$$

Stochastic

Gradient Descent :-

1. Initialize β

2. For randomly selected sample

$\Rightarrow h_\beta(x) \Rightarrow$ prediction for x_i using the current i/p

$\Rightarrow$ for each non zero feature of x_{ij}

$$\beta_j = \beta_j * \alpha(y^{(i)} - h_\beta(x^{(i)})) * x_j^{(i)}$$

eg:- $\beta = (0,0,0)$ $\quad \alpha = 0.1$ $\quad x_1 = ((62, 58), 1)$

$x_2 = ((52, 41), 0)$

$$h_\beta(x_1) = \frac{1}{1+e^{-0}} = 0.5$$

Use the score to fine tune the parameter

$\beta_0 = 0 + 0.1[(1 - 0.5) * 1] = 0 + 0.1 * 0.5 = 0.05$

$\beta_1 = 0 + 0.1(1 - 0.5) * 62 = 0 + 0.05 \times 62 = 3.1$

$\beta_2 = 0 + 0.1(1 - 0.5) * 58 = 0 + 0.05 \times 58 = 2.9$

$$h_\beta(x_2) = \frac{1}{1+e^{-(0.05 + 3.1 * 52 + 2.9 * 41)}}$$

$$= \frac{1}{1+e^{-280.15}} = 1$$

Expected value is zero, fine tune the parameters

$\beta_0 = 0.05 + 0.1[0 - 1] * 1 = -0.05$

$\beta_1 = 3.1 + 0.1[0 - 1] * 52 = -2.1$

$\beta_2 = 2.9 + 0.1[0 - 1] * 41 = -1.2$.

(3)

$$x_1 \Rightarrow h_\beta(x) = \frac{1}{1+e^{-(-0.05-2.1*62-1.2*58)}} = 1.6\times10^{-87}$$

$\Rightarrow$ class 0

$$x_2 = h_\beta(x) = \frac{1}{1+e^{-(0.05-2.1*52-1.2*41)}}$$

$$= 1.53\times10^{-69}$$

$= $ class 0

Accuracy $= 0.5 \Rightarrow 50\%$

(e)

$1/2$.

Gradient Descent (GD):-

$$\beta_j = \beta_j - \alpha\frac{\partial}{\partial\beta_j} l(\beta/y)$$

1. Start with random
2. Loop until convergence
   2.1 compute gradient
   2.2 Update.
3. Return.

Stochastic GD:-

1. Start with random
2. Loop until convergence
   2.1. pick the single data point i
   2.2 compute gradient over that single point
   2.3 Update
3. Return.

Gradient descent	Stochastic GD
- Small (or) medium dataset	- Large dataset
- Slow in computation.	- Fast in computation

Newton Raphson:-

$$\beta_1 = \beta_1 - \frac{\partial^2}{\partial \beta^2} l(P/y) \Rightarrow \beta_1 - \frac{\partial}{\partial \beta} l(P/y) * \left(\frac{\partial^2}{\partial \beta^2} l(P/y)\right)^{-1}$$

→ Reduces Number of iterations.

$$\frac{\partial}{\partial \beta} l(P/y) = (y - h_\beta(x)) * x. = yx - xh_\beta(x) = yx - x * \frac{1}{1+e^{-\beta^T x}}$$

$$\frac{\partial^2}{\partial \beta^2} l(P/y) = 0 - x(h_\beta(x)(1-h_\beta(x)) * x)$$

Hessian ⇐ $H = -h_\beta(x)(1-h_\beta(x)) * x^2$

$$\beta_1 = \beta_1 - \left(h_\beta(x^{(i)})(1-h_\beta(x^{(i)}))\, x^{(i)}(x^{(i)})^T\right)^{-1} \frac{\partial}{\partial \beta}(l(P/y))$$

(or)

$$\beta_1 = \beta_1 - H^{-1} \cdot \frac{\partial}{\partial \beta} l(P/y)$$