05-02-2022

Saturday, February 5, 2022 2:04 PM

Inverse logit func. or Sigmoid()

$$\sigma = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}} \quad \text{or} \quad \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}}$$

$$\sigma = \frac{1}{1 + e^{-\beta^T x}} \;//$$

$$\sigma(\beta^T x) = \frac{1}{1 + e^{-\beta^T x}}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Morris notation ↓

$$\frac{\delta[\sigma(x)]}{\delta x} \quad \frac{\partial(\sigma(x))}{\partial x} = -\frac{1}{(1+e^{-x})^2} \cdot e^{-x} \cdot -1$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{e^{-x}}{1+e^{-x}} \cdot \frac{1}{1+e^{-x}}$$

$$= \frac{1+e^{-x}-1}{1+e^{-x}} \cdot \frac{1}{1+e^{-x}}$$

$$= \left[1 - \frac{1}{1+e^{-x}}\right]\left(\frac{1}{1+e^{-x}}\right)$$

$$\frac{\partial(\sigma(z))}{\partial(z)} = [1 - \sigma(z)]\,\sigma(z) \quad \text{— (A)}$$

Maximum Likelihood Estimate of β / β_{MLE}

— Best β value

minimises Prob. of seeing training data

β_{MLE} ⇒ likelihood of seeing same training data.

y	$\hat{y}$	
0	0	p_1
1	1	p_2
1	1	p_3 } Prob > 0.5 (as high as possible)
0	0	p_4 } Prob < 0.5 (as low as possible)
1	1	p_5

Joint Prob. $= (1-p_1)\, p_2 \cdot p_3 (1-p_4)\, p_5$

$$\hat{\beta}_{MLE} = \underset{\beta}{\operatorname{argmax}}\; P(y \mid x\,; \beta)$$

$$= \underset{\beta}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i / x_i\,; \beta)$$

$$L(\beta) = \max \prod_{i=1}^{n} P(y_i / x_i\,; \beta)$$

$$= \max \prod_{i=1}^{n} P(y_i = 1) \cdot \prod_{i=1}^{n} P(y_i = 0)$$

$$= \max \prod_{i=1}^{n} P(y_i = 1) \cdot \prod_{i=1}^{n} (1 - P(y_i = 1))$$

Let $\sigma(z) = \frac{1}{1+e^{-z}}$

Let $S = P(y_i = 1) = \frac{1}{1 + e^{-\beta^T x}}$

$$= \max \prod_{i=1}^{n} S \cdot \prod_{i=1}^{n} (1 - S)$$

Bernoulli : $f(x) = p^x (1-p)^{1-x}$ → explanation

$$L(\beta) = \max_{\beta} \prod_{i=1}^{n} S^{y_i} (1-S)^{(1-y_i)}$$

log Likelihood ($l\beta$)

$$\log(L(\beta)) = LL(\beta) = \log \left(\max_{\beta} \prod_{i=1}^{n} S^{y_i} (1-S)^{(1-y_i)} \right)$$

$$= \max_{\beta} \sum_{i=1}^{n} \left(\log(S)^{y_i} + \log(1-S)^{(1-y_i)} \right)$$

$$LL(\beta) = \max_{\beta} \sum_{i=1}^{n} \left(y_i \log(S) + (1 - y_i) \log(1-S) \right)$$

Cost function (or) Loss function : Both have slight diff.

Minimise cost Maximise Likelihood

$$NLL(\beta) = -\min_{\beta} \sum_{i=1}^{n} \left(y_i \log S + (1-y_i) \log(1-S) \right)$$

$$\hat{\beta}_{MLE} = \operatorname{argmax}(LL(\beta)) = \operatorname{argmin}(NLL(\beta))$$

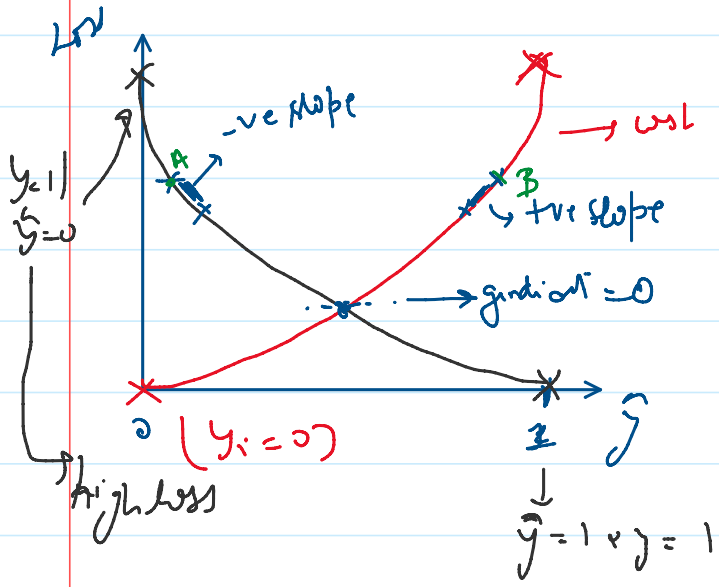
To calc β vals → we use Optimizers. — less accuracy / Jacobian Matrix

To calc β vals → we use Optimisers. // Jacobian matrix

- 1st Order -eg: Gradient Descent (Less Costly)
- 2nd Order -eg: Newton-Raphson (High Cost)
  - ↓ Better Accuracy
  - \ Hessian Matrix

Gradient Ascend - maximise Likelihood func.

// Descent - Minimise Cost / Loss func.



Gradient = Slope

$$= \frac{y_2 - y_1}{x_2 - x_1}$$

← 2:57 PM

{ GD a) in each iteration we take steps proportional to the -ve of the gradient at that pt. }

$$\beta = \beta - \alpha \frac{\partial}{\partial \beta} NLL(\beta)$$

↳ Learning Rate: How fast should model learn

$\alpha = 0.1$ to 0.3

From above graph

i) Let β be at 'A' Gradient = -ve

Loss $\propto \frac{1}{\beta}$ $\therefore \beta = \beta - \alpha(-\text{ve grad.})$

$= \beta + \alpha$ grad.

$\therefore \uparrow \beta$

β $\quad = \beta + \alpha$ grad.

$\therefore \uparrow \beta$

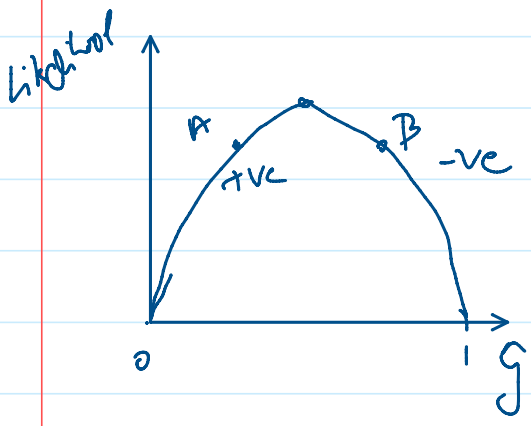
ii) Let β be at 'B' Gradient = +ve

$\beta = \beta - \alpha$ (+ve grad.)

$= \beta - \alpha$ grad

$\therefore \beta \downarrow$

Gradient Ascent



$$\beta = \beta + \alpha \frac{\partial}{\partial \beta^T} LL(\beta)$$

$\beta = \beta + \alpha$ (slope) — (A)

$\beta = \beta + \alpha$ (-ve slope) — (B)

$= \beta - \alpha$ (slope)

$$LL(\beta) = \sum_{i=1}^{n} \left[y_i \log\left(\sigma(\beta^T x_i)\right) + (1-y_i) \log\left(1 - \sigma(\beta^T x_i)\right) \right]$$

B C

$$\frac{\partial LL(\beta)}{\partial \beta^T} = \frac{\partial}{\partial \beta^T}(B) + \frac{\partial}{\partial \beta^T}(C)$$

For B : $\frac{\partial}{\partial \beta^T} \left(\sum_{i=1}^{n} y_i \log\left(\sigma(\beta^T x_i)\right) \right)$ ← Chain Rule

$$= \sum y_i \frac{\partial \left(\log(\sigma(\beta^T x_i))\right)}{\partial \left(\sigma(\beta^T x_i)\right)} \cdot \frac{\partial \left(\sigma(\beta^T x_i)\right)}{\partial \beta^T x_i} \cdot \frac{\partial \beta^T x_i}{\partial \beta^T}$$

From A ↘

From A

$$\Rightarrow \sum y_i \times \frac{1}{\sigma(\beta^T x_i)} \sigma(\beta^T x_i)(1-\sigma(\beta^T x_i)) \cdot x_i$$

$$\Rightarrow \sum y_i (1-\sigma(\beta^T x_i)) x_i$$

For C:

$$\frac{\partial}{\partial \beta^T} \left[\sum (1-y_i) \log (1-\sigma(\beta^T x_i)) \right]$$ Chain Rule

$$= \sum (1-y_i) \frac{\partial \log(1-\sigma(\beta^T x_i))}{\partial (1-\sigma(\beta^T x_i))} \cdot \frac{\partial (1-\sigma(\beta^T x_i))}{\partial (\beta^T x_i)} \cdot \frac{\partial \beta^T x_i}{\partial \beta^T}$$

From A

$$= \sum (1-y_i) \frac{1}{1-\sigma(\beta^T x_i)} \left(-\sigma(\beta^T x_i)(1-\sigma(\beta^T x_i)) \right) \cdot x_i$$

$$= - \sum (1-y_i) \sigma(\beta^T x_i) x_i$$

Combine B and C

$$\Rightarrow \sum \left[y_i (1-\sigma(\beta^T x_i)) x_i - (1-y_i) \sigma(\beta^T x_i) x_i \right]$$

$$\Rightarrow \sum \left(y_i - y_i \sigma(\beta^T x_i) - \sigma(\beta^T x_i) + y_i \sigma(\beta^T x_i) \right) x_i$$

$$\frac{\partial LL(\beta)}{\partial \beta^T x} = \sum_{i=1}^{n} \left[(y_i - \sigma(\beta^T x_i)) x_i \right]$$

Gradient Ascent

$$\left\{ \beta = \beta + \alpha \left(\sum_{i=1}^{n} (y_i - \sigma(\beta^T x_i)) x_i \right) \right\} \star$$

$$\hat{y} = \frac{1}{1 + e^{-\beta^T x}}$$