(10) The Newton - Raphson update, for minimizing a function $E(\omega)$ takes the form

$$\omega^{(new)} = \omega^{(old)} - H^{-1} \nabla E(\omega) \longrightarrow (4.0)$$

where,
$\nabla E(\omega)$ is known as gradient & H is the Hessian matrix whose elements comprise the second derivatives of $E(\omega)$ with respect to the components of ω. ($H = \nabla \nabla E(\omega)$)

In logistic regression $E(\omega)$ is given by cross entropy error function

$$E(\omega) = -\ln p(t|\omega) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

$$\nabla E(\omega) = \sum_{n=1}^{N} (y_n - t_n)\phi_n = \phi^T (y-t) \longrightarrow (4.1)$$

where, ϕ is the design matrix whose n^{th} row is given by ϕ_n^T. $\rightarrow \phi_n = \phi(x_n)$ $\{\phi_n, t_n\}$ is the data set. ~~input~~ : $\{\phi_1, t_1\} \{\phi_2, t_2\} \cdots$

$y_n = \sigma(\omega^T \phi_n) \rightarrow$ Predicted output. $y - t = \begin{bmatrix} y_1 - t_1 \\ y_2 - t_2 \\ y_n - t_n \end{bmatrix}$

$$H = \nabla \nabla E(\omega) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n \phi_n^T$$

$$= \phi^T R \phi \longrightarrow (4.2)$$

Here, we have made use of $\frac{d\sigma}{da} = \sigma(1-\sigma)$

Here, we introduced R which is an $N \times N$ diagonal matrix,

where $R_{ii} = y_i(1-y_i)$

Note that H is not constant, but depends on ω

$\omega^{(new)} =$

Putting 4.1 & 4.2 in 4.0 we get

$$w^{(new)} = w^{(old)} - \underbrace{(\phi^T R \phi)^{-1}}_{H^{-1}} \underbrace{\phi^T(y-t)}_{\nabla E(\omega)}$$

$$= (\phi^T R \phi)^{-1} \left\{ \phi^T R \phi\, \omega^{(old)} - \phi^T (y-t) \right\}$$

$$= (\phi^T R \phi)^{-1} \phi^T R z \longrightarrow 4.3$$

where z is an N - dimensional vector

$$z = \phi\, \omega^{(old)} - R^{-1}(y-t)$$

4.1 $\longrightarrow$ gradient

4.2 $\longrightarrow$ Hessian

4.3 $\longrightarrow$ update equation

If ϕ_n is M- dimensional, them ω is also m - dimensional

M- dimensional. Initialize ω to a random M- dimensional vector.

while (true) {
Evaluate $\nabla E(\omega)$
Evaluate H, since R is dependent on ω
Evaluate z
Evaluate $\omega^{(new)}$
if ($\|\omega^{(old)} - \omega^{(new)}\| < \varepsilon$) $\rightarrow$ tolerance of Newton Raphoson's method as specified by user
break;
}

(c) We know that,

$$H = \phi^T R \phi$$

where, R is a digonal matrix

$$R_{ii} = y_i(1-y_i) > 0 \quad \because 0 \le y_i < 1$$

where $y_i = \sigma(\omega^T \phi_i)$

For, any arbitary vector u of dimensions $M \times 1$,

$$u^T H u = u^T \phi^T R \phi u$$

$$= (\phi u)^T \overbrace{R}^{\text{all elements are +ve}} \phi u$$

$\phi u = N \times 1$ dimensional matrix.

Let $\phi u = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$ $\quad R = \begin{bmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & & & \\ 0 & & \ddots & & \\ \vdots & & & & \\ 0 & & & & \alpha_n \end{bmatrix}$

$$= \begin{bmatrix} \beta_1 & \cdots & \beta_n \end{bmatrix}_{1\times n} \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{bmatrix}_{n\times n} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}_{n\times 1}$$

$$= \left\{ \begin{bmatrix} \alpha_1\beta_1 & \alpha_2\beta_2 & \cdots & \alpha_n\beta_n \end{bmatrix}_{1\times n} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}_{n\times 1} \right.$$

$$= \left[\alpha_1\beta_1^2 + \alpha_2\beta_2^2 + \cdots + \alpha_n\beta_n^2 \right]$$

$$> 0$$

$u^T H u > 0 \;\; \forall u \in R^M \Rightarrow H$ is positive definite

It follows that the error function is a convex function of ω & hence has a unique minimum.

4b) The Solution of ω that minimises the error function in 3C ~~is~~ was found out to be

$$\omega = (\phi^T R \phi)^{-1} \phi^T R t$$

which is Similar to (4.3)

$$\left(\omega^{(new)} = (\phi^T R \phi)^{-1} \phi^T R z \right)$$

This similarity shows that Newton - Raphson update scheme is related to weighted least squares problem Because the weighing matrix R is not constant but depends on parameter vector ω, we must apply the normal equations iteratively, each time using the new weight vector ω to compute a revised weighting matrix R. For this reason, the algorithm is known as iterative reweighted least squares.