QUESTION 4

ABHAY SHANKAR K: CS21BTECH11001 & KARTHEEK TAMMANA: CS21BTECH11028

(I) Question: Provide the expressions of the gradient, Hessian, and update equations for the Newton-Raphson optimization technique used to obtain the parameters in the logistic regression model. Provide an algorithm describing the methodology.

Solution: Knowing the maximum likelihood function,

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{t_n}$$

we can obtain the cross-entropy error function by taking negative logarithm:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} (t_n \ln y_n + (1 - t_n) \ln(1 - y_n)) \tag{1}$$

Note that $y_n = \sigma(\mathbf{w}^{\mathbf{T}} \boldsymbol{\phi}_n)$ where σ is the sigmoid function and $\boldsymbol{\phi}_n$ are the features. Furthermore, we have the design matrix $\boldsymbol{\Phi}$ with $\boldsymbol{\phi}_n^T$ as its n'th row.

For both instances of differentiation, we use the fact

$$\frac{d\sigma(a)}{da} = \sigma(a)(1 - \sigma(a))$$

Taking gradients of (1) with respect to $\mathbf{w}$, we have

- Gradient of the error:

$$\begin{aligned}
\nabla E(\mathbf{w}) &= -\sum_{n=1}^{N} t_n \cdot \frac{1}{y_n} y_n(1 - y_n)\phi_n - (1 - t_n) \cdot \frac{1}{1 - y_n} y_n(1 - y_n)\phi_n \\
&= \sum_{n=1}^{N} (y_n - t_n)\phi_n \\
&= \boldsymbol{\Phi}^{\mathbf{T}}(\mathbf{y} - \mathbf{t})
\end{aligned} \tag{2}$$

- The Hessian:

$$\nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^T = \boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\boldsymbol{\Phi}$$

  where R is the diagonal matrix given by $R_{nn} = y_n(1 - y_n)$

- Update function:

$$\mathbf{w}^{(new)} = \mathbf{w}^{(old)} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) =$$

$$\begin{aligned}
\mathbf{w}^{(new)} &= \mathbf{w}^{(old)} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) \\
&= \mathbf{w}^{(old)} - (\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathbf{T}}(\mathbf{y} - \mathbf{t}) \\
&= (\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\left(\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\boldsymbol{\Phi}\mathbf{w}^{(old)} - \boldsymbol{\Phi}^{\mathbf{T}}(\mathbf{y} - \mathbf{t})\right) \\
&= (\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{R}\mathbf{z}
\end{aligned} \tag{3}$$

  with

$$\mathbf{z} = \boldsymbol{\Phi}\mathbf{w}^{(old)} - \mathbf{R}^{\mathbf{-1}}(\mathbf{y} - \mathbf{t})$$

The algorithm for update, implemented in python:

```python
import numpy as np
def update(w, Phi, t):
    y =  mp.array([sigmoid(p @ w) for p in Phi])
    R = np.diag(y * (1 - y))
    z = Phi @ w - np.linalg.inv(R) @ (y - t)
    return np.linalg.inv(Phi.T @ R @ Phi) @ Phi.T @ R @ z
```

(II) Modifying (3), we get

$$(\mathbf{\Phi^T R \Phi})\mathbf{w} = \mathbf{\Phi^T R z}$$

which is the normal equation for weighted least squares. Thus, the new weight vector $\mathbf{w}^{(new)}$ is the solution to the weighted least squares problem with $\mathbf{z}$ and the weighing matrix $\mathbf{R}$. However, the weighing matrix R is not constant, but depends on the parameter vector $\mathbf{w}$.

Thus, we must apply the normal equations iteratively, each time using the new weight vector $\mathbf{w}$ to compute a revised weighing matrix R. So, the algorithm is known as iterative reweighted least squares (IRLS).

(III) With the Hessian:

$$\mathbf{H} = \mathbf{\Phi^T R \Phi}$$

We know that a function is convex if its Hessian is positive definite. Thus, it is sufficient to prove positive-definiteness of the Hessian.

Expanding $\mathbf{u^T H u}$, we can prove positive-definiteness

$$\begin{aligned}
\mathbf{u^T H u} &= \mathbf{u^T \Phi^T R \Phi u} \\
&= (\mathbf{u^T \Phi^T})\mathbf{R}(\mathbf{\Phi u}) \\
&= \sum_{n=1}^{N} u_n {\phi_\mathbf{n}}^T R_{nn} \phi_\mathbf{n} u \\
&= \sum_{n=1}^{N} R_{nn} \left\|\phi_n\right\|^2 (u_n)^2 \\
&> 0
\end{aligned} \tag{4}$$

We have used the fact that $R_{nn} > 0$ and $\mathbf{u} \neq 0$. Thus, the Hessian is positive-definite, and the function is convex with unique minimum.