Iteratively Reweighted PLS Estimation for Tensor Generalized Linear Regression

Haesong Choi*
Department of Statistics, Florida State University

April 20, 2020

1 Introduction

The form of multidimensional array, known as tensor is encountered in a variety of fields such as the medical imaging field with complex structure. With this data, the problem can be formulated as regression with discrete outcome as the response on the tensor predictor. However, the high dimensionality of the data is usually challenging. To deal with those high dimension tasks, partial least squares (PLS) regression is particularly appropriate. PLS regression suggests efficient dimension reduction solution by projecting both the exploratory and response variables to the low dimensional space of latent variables. For this reason, we feel it important to extend PLS to the tensor structure motivated by Marx (1996). This paper develops PLS estimation for tensor generalized linear regression (GLR). By introducing the block relaxation algorithm in Zhou (2013), we break tensor predictor to low dimensional GLM regression model and implement PLS with the constructed new model to estimate our parameters. The rest of the article is organized as follows. Section 2 presents the estimation for generalized linear regression model and the concept of partial least squares regression. Section 3 reviews our reference paper, Marx (1996) and Section 4 establishes iteratively reweighted PLS estimation for tensor GLR.

*This is the final project for STA5934

2 Basic Model

2.1 Generalized Linear Regression

We first start with the classical generalized linear model (GLM) in McCullagh and Nelder (1983). Suppose there is a matrix of predictor, $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$ which does not contain a column vector of ones. The random response vector, $\boldsymbol{Y} \in \mathbb{R}^{n\times 1}$, has independent random variables y_i which belongs to an exponential family with probability density (mass) function

$$y_i \sim f(y_i;\theta_i,\phi) = \exp\left[\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)}\right\} + c(y_i,\phi)\right] \tag{1}$$

where $\phi > 0$ denotes a dispersion parameter, θ is referred to the natural parameter of the distribution, and $a(\cdot), b(\cdot), c(\cdot)$ are known functions. Using classical results on the exponential family, we have

$$E(y_i) = b^{'}(\theta_i) = \mu_i$$
$$\text{var}(y_i) = a(\phi)b^{''}(\theta_i) = a(\phi)V(\mu_i).$$

In the GLM setting, the mean $\mu = E(\boldsymbol{Y}|\boldsymbol{X})$ is connected to a linear predictor $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ via

$$g(\mu_i) = \eta_i = \alpha_i + \boldsymbol{\beta}^\top\boldsymbol{x}_i \tag{2}$$

where g is a strictly increasing or decreasing and twice differentiable function and it is called the canonical link function. η_i denotes the linear systematic part with intercept α and the coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$. Since g is invertible, we can rewrite it as

$$\mu_i = g^{-1}(\boldsymbol{\beta}^\top\boldsymbol{x_i}).$$

The log-likelihood for $\boldsymbol{Y}$ is

$$\ell(\alpha,\boldsymbol{\beta};\boldsymbol{X}) = \sum_{i=1}^{n}\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi}\right\} + c(y_i,\phi), \tag{3}$$

where $a(\phi) \equiv \phi$ is constant over all i. We can obtain the maximum likelihood estimates by solving the score equations of $\beta_j, j = 1, \ldots, p$

$$s(\beta_j) = \frac{\partial\ell}{\partial\beta_j} = \sum_{i=1}^{n}\frac{y_i - \mu_i}{\phi V(\mu_i)} \times \frac{x_{ij}}{g^{'}(\mu_i)} = 0$$

where $V(\mu_i) = b^{''}(\theta_i)$. But, it is clear that there does not exist simple solution for the likelihood estimator of $\boldsymbol{\beta}$. A general method of solving score equations is the iterative Fishers' Method of Scoring algorithm. In the r-th iteration, the new estimate $\beta^{(r+1)}$ is obtained from the previous estimate $\beta^{(r)}$ by

$$\boldsymbol{\beta}^{(r+1)} = \boldsymbol{\beta}^{(r)} + s(\boldsymbol{\beta}^{(r)})E(H(\boldsymbol{\beta}^{(r)}))^{-1}$$

where the gradient vector $s(\beta^{(r)})$ is

$$s(\boldsymbol{\beta}^{(r)}) = \left(\frac{\partial \ell}{\partial \beta_1}, \ldots, \frac{\partial \ell}{\partial \beta_p}\right)^{'} |_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(r)}}$$

and the Hessian matrix $H(\beta^{(r)})$ is denoted by

$$H(\boldsymbol{\beta}^{(r)})_{jk} = \frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k}|_{\beta=\beta^{(r)}}$$

for $j, k = 1, \ldots, p$. By introducing the diagonal matrix $\boldsymbol{W}^{(r)} = \text{diag}(w_i^{(r)}) \in \mathbb{R}^{n \times n}$

$$w_i^{(r)} = \frac{1}{\phi V(\mu_i^{(r)})(g'(\mu_i^{(r)}))^2},$$

it turns out that the updates can be written as

$$\beta^{(r+1)} = (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{z}^{(r)} \tag{4}$$

i.e. the score equations for a weighted least squares regression of $\boldsymbol{z}^{(r)}$ on $\boldsymbol{X}$ with weights $\boldsymbol{W}^{(r)}$, where

$$\boldsymbol{z}^{(r)} = \boldsymbol{\eta}^{(r)} + (\boldsymbol{y} - \boldsymbol{\mu}^{(r)})g^{'}(\boldsymbol{\mu}^{(r)}). \tag{5}$$

$w_i^{(r)}$ and $z_i^{(r)}$ is updated at each iteration step until convergence. Consequently we can obtain $\boldsymbol{\eta}^{(r+1)}$ by plugging in (2) via

$$\boldsymbol{\eta}^{(r+1)} = \boldsymbol{\alpha}^{(r)}\mathbf{1} + \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{z}^{(r)}. \tag{6}$$

where the update for scalar $\boldsymbol{\alpha}^{(r)}$ is the weighted mean of the adjusted dependent vector $\boldsymbol{z}^{(r)}$ using the weights in $\boldsymbol{W}^{(r)}$. The proof of (4) and (5) is collected in the Supplemental Materials.

2.2 Partial Least Squares Regression

The Partial least squares (PLS) regression is one of the iterative procedures for estimating the linear relationship between response variables and predictor variables. Given the observation matrices $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{Y} \in \mathbb{R}^{n\times r}$, this model assumes that the observations are iteratively projected into the low dimensional spaces of latent variables (Wold (1966)):

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\top} + \boldsymbol{E}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\top} + \boldsymbol{F}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_d] \in \mathbb{R}^{n\times d}$ is the matrix of the latent variables, $\boldsymbol{P} \in \mathbb{R}^{p\times d}$ and $\boldsymbol{Q} \in \mathbb{R}^{r\times d}$ are the matrices of the loading vectors. $\boldsymbol{E}$ and $\boldsymbol{F}$ are residual matrices, and d is the number of iterations. Some PLS algorithms are appropriate for the case where $\boldsymbol{Y}$ is a column vector. The PLS procedure is particularly well suited for high dimensional regression because of its efficient dimensional reduction solution. With the projection to latent structures, PLS reduces the predictor $\boldsymbol{X}$ to a sufficiently lower dimensional latent matrix $\boldsymbol{T}$.

3 Iteratively Reweighted PLS Estimation for Generalized Linear Regression

Before developing a PLS for tensor predictors, we quickly review of Iteratively reweighted partial least squares (IRPLS) estimation in Marx (1996). This paper extend the concept of PLS into the framework of generalized linear models. They introduced the following two latent factor models of $\boldsymbol{X}$ and $\boldsymbol{z}$ in (5). With $d \ll p$,

$$\boldsymbol{X} = \sum_{j=1}^{d} \boldsymbol{t}_j \boldsymbol{p}_j^{\top} + \boldsymbol{E}_d \tag{7}$$

$$\boldsymbol{z} = \sum_{j=1}^{d} \boldsymbol{q}_j \boldsymbol{t}_j + \boldsymbol{f}_d \tag{8}$$

where $\boldsymbol{t}_j \in \mathbb{R}^n$ denotes latent variables, $\boldsymbol{p}_j \in \mathbb{R}^p$ is a loading vector and $\boldsymbol{E}_k$ is a residual matrix. The $\boldsymbol{q}_j$ are scalar coefficients, and $\boldsymbol{f}_d \in \mathbb{R}^n$ is an residual vector. Now, we present the IRPLS estimation algorithm for GLM Marx (1996). The full estimation procedure is summarized in Algorithm 2. After constructing and the latent variables, they obtained estimates in (6) and performed GLR. The algorithm 2 is collected in the supplemental materials.

4 Tensor IRPLS Estimation for Generalized Linear Regression

Motivated by Marx (1996), we develop a IRPLS algorithm for tensor predictor. Zhou (2013) proposed GLM with tensor variables which admits CANDECOMP/PARAFAC (CP) decomposition. For a response variable Y, m-dimensional tensor predictor $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$, the response is assumed to belong to an exponential family where the linear systematic part is of the form,

$$g(\mu) = g(E(Y|\boldsymbol{X})) = \alpha + \langle \boldsymbol{B}, \boldsymbol{X} \rangle \tag{9}$$

where $g(\cdot)$ is a strictly increasing link function and $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \text{vec}\boldsymbol{B}, \text{vec}\boldsymbol{X} \rangle = \sum \beta_{i_1,\ldots,i_M} x_{i_1,\ldots,i_M}$. A tensor coefficient $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ captures the effects of tensor covariate $\boldsymbol{X}$ and follows a rank-R decomposition,

$$\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_M^{(r)},$$

where $\boldsymbol{\beta}_m^{(r)} \in \mathbb{R}^{p_m}$ $m = 1, \ldots, M,$ $r_m = 1, \ldots, R_m$ are all column vectors. Following Kolda (2006), the decomposition can be represented by $\boldsymbol{B} \approx [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_M]\!]$, where $\boldsymbol{B}_m = [\boldsymbol{\beta}_m^{(1)}, \ldots, \boldsymbol{\beta}_m^{(R)}] \in \mathbb{R}^{p_m \times R}, m = 1, \ldots, M$. In this case, $\boldsymbol{B}$ can be converted to a Matrix and a vector:

$$\boldsymbol{B}_{(m)} \approx \boldsymbol{B}_m(\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_{m+1} \odot \boldsymbol{B}_{m-1} \odot \cdots \odot \boldsymbol{B}_1)^\top \tag{10}$$

and

$$\text{vec}\boldsymbol{B} = (\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_1)\mathbf{1}_R$$

where $\mathbf{1}_R$ is R-column vector of ones. We estimate parameters in (9) by pursuing the log-likelihood function given i.i.d. data $\{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$:

$$\ell(\alpha, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_M; \boldsymbol{X}) = \sum_{i=1}^{n} \left\{ \frac{y_i\theta_i - b(\theta_i)}{\phi} \right\} + \sum_{i=1}^{n} c(y_i, \phi).$$

Zhou (2013) introduced block relaxation algorithm for efficiently maximizing $\ell(\alpha, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_M)$. A key point is that $g(\mu)$ is not linear in $(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_M)$ jointly, but linear in $\boldsymbol{B}_m$ individually in (9). So, this algorithm alternatively updates $\boldsymbol{B}_m, m = 1, \ldots, M$, while keeping other parameters fixed. At the mth iteration, $\langle \boldsymbol{B}, \boldsymbol{X} \rangle$ in (9) can be rewritten as

$$\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \left\langle \boldsymbol{B}_m, \boldsymbol{X}_{(m)}(\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_{m+1} \odot \boldsymbol{B}_{m-1} \odot \cdots \odot \boldsymbol{B}_1) \right\rangle$$

with parameters $\text{vec}\boldsymbol{B}_m \in \mathbb{R}^{p_m \cdot R}$. As a result, it transforms into a traditional GLM and they implement this algorithm by breaking it into low dimensional GLM regression model. Now, this paper extends Zhou (2013) to IRPLS introduced in section 3. Given i.i.d. $\{(y_i, \boldsymbol{x}_i^*), i = 1, \ldots, n\}$, we can rewrite 9 as when updating $\boldsymbol{B}_m$

$$
\begin{aligned}
g(\mu_i) &= \alpha + \left\langle \boldsymbol{B}_m, \boldsymbol{X}_{(m)}(\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_{m+1} \odot \boldsymbol{B}_{m-1} \odot \cdots \odot \boldsymbol{B}_1) \right\rangle \\
&= \alpha + \mathrm{vec}^\top \boldsymbol{B}_m \mathrm{vec}\{\boldsymbol{X}_{(m)}(\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_{m+1} \odot \boldsymbol{B}_{m-1} \odot \cdots \odot \boldsymbol{B}_1)\} \\
&= \alpha + \boldsymbol{\beta}^* \boldsymbol{x}_i^*.
\end{aligned}
$$

where $\boldsymbol{\beta}^*$ is the coefficients for linear relationship between $g(\mu_i)$ and $\boldsymbol{x}_i^*$.

Algorithm 1: IRPLS Estimation for tensor GLS

Result: $\alpha, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_M$

[0] Initialize $\alpha^{(0)} = \arg\max_\alpha \ell(\alpha, \mathbf{0}, \ldots, \mathbf{0})$, $\boldsymbol{B}_m^{(0)} \in \mathbb{R}^{p_m \times R}$ a random matrix for $m = 1, \ldots, M$.

repeat

  for $m = 1, \ldots, M$ **do**

    [1] Define $\boldsymbol{x}_i^* = \mathrm{vec}\{\boldsymbol{X}_{(m)}(\boldsymbol{B}_M \odot \cdots \odot \boldsymbol{B}_{m+1} \odot \boldsymbol{B}_{m-1} \odot \cdots \odot \boldsymbol{B}_1)\}$, $i = 1, \ldots, n$

    [2] Initialize $\boldsymbol{E}_0 = \boldsymbol{X}^* \in \mathbb{R}^{n \times p_m \cdot R}$; $\boldsymbol{f}_0 = \psi(\boldsymbol{y}) \in \mathbb{R}^n$; $\boldsymbol{W} = \{\phi \boldsymbol{V}(\boldsymbol{\mu})\}^{-1} h'[\psi(\boldsymbol{y})]^2 \in \mathbb{R}^{n \times n}$

      for $k = 1, \ldots, K$ **do**

        [3] Define $\boldsymbol{r}_k$ = the coefficient for the WLS regression of $\boldsymbol{f}_{k-1}$ on $\boldsymbol{E}_{k-1}$ with $\boldsymbol{W}$.

        [4] Define a latent variable $\boldsymbol{t}_k = \boldsymbol{E}_{k-1} \boldsymbol{r}_k \in \mathbb{R}^n$.

        [5] Scale $\boldsymbol{t}_k$ with center (=Weighted mean of $\boldsymbol{t}_k$ with $\boldsymbol{W}$) and scale (=**SS**).

        [6] Define $\boldsymbol{q}_k$ = the coefficient for the WLS regression of $\boldsymbol{f}_{k-1}$ on $\boldsymbol{t}_k$ with $\boldsymbol{W}$.

        [7] Deflate the response vector $\boldsymbol{f}_k = \boldsymbol{f}_{k-1} - \boldsymbol{t}_k \boldsymbol{q}_k$.

        [8] Define $\boldsymbol{E}_k$ = Residuals for the WLS regression of $\boldsymbol{E}_{k-1}$ on $\boldsymbol{t}_k$ with $\boldsymbol{W}$.

      end for

        [9] $\boldsymbol{\eta}$ = Weighted mean of $\boldsymbol{f}_0$ with weight $\boldsymbol{W}$ + $\sum_{k=1}^{R} \boldsymbol{q}_k \boldsymbol{t}$

        [10] $\boldsymbol{W} = \{\phi \boldsymbol{V}(\boldsymbol{\mu})\}^{-1} h'[\psi(\boldsymbol{\eta})]^2$

        until $\triangle \hat{\eta} < \epsilon$

    [11] Define $\boldsymbol{\beta}_m^*$ = the coefficient for the Glm of $\boldsymbol{y}$ on $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K) \in \mathbb{R}^{n \times K}$

    [12] Convert $\boldsymbol{\beta}_m^*$ into a matrix $\boldsymbol{B}_m^{(t+1)}$.

  end for

[13] $\alpha^{(t+1)} = \arg\max_\alpha \ell(\alpha, \boldsymbol{B}_1^{(t+1)}, \boldsymbol{B}_2^{(t+1)}, \ldots, \boldsymbol{B}_M^{(t+1)})$

Then our new log likelihood for $\boldsymbol{Y}$ is

$$\ell^*(\alpha, \boldsymbol{\beta}^*; \boldsymbol{X}^*) = \sum_{i=1}^{n} \left\{ \frac{y_i\theta_i - b(\theta_i)}{\phi} \right\} + c(y_i, \phi),$$

where $\boldsymbol{X}^* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*)^\top$ and we can apply IRPLS (Marx (1996)) to new GLM. We can easily construct two latent factor models of $\boldsymbol{X}^*$ and $\boldsymbol{z}$ in (8). The full estimation procedure for $(\alpha, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_M)$ is summarized in Algorithm 1. $h(\cdot)$ is a unique inverse of $g(\boldsymbol{\mu})$. For the example of the initialization for $\boldsymbol{B}_m$, we can use the binary matrix with the true signal region equal to one and the rest zero. Step 1 presents the definition of $\boldsymbol{X}^*$ by using block relaxation algorithm in Zhou (2013). In process 2, we initialize the residual vector and matrix in the latent factor model which is followed by Marx (1996). The initial values for residual vector $\boldsymbol{f}_0$ is transformed into $\psi(\boldsymbol{y})$. For example, Marx (1996) suggests $\psi_P(\boldsymbol{y}) = \ln(y + 0.5)$ and $\psi_B(\boldsymbol{y}) = (\boldsymbol{y} + 0.5)/2$ for Poisson and Bernoulli. In step 3 and 4, we construct the latent variables and deflate the predictor matrix in step 8 and the response vector in step 7. Then we obtain the updated $\boldsymbol{\eta}$ and $\boldsymbol{W}$ in step 9 and 10. Once $\boldsymbol{\eta}$ is converged, step 12 implements GLR of y on T which is the constructed latent matrix. Finally, step 13 and step 14 obtain the estimates for our parameters.

SUPPLEMENTAL MATERIALS

The proof of the new estimate $\beta^{(r+1)}$ is as follows. the new estimate $\beta^{(r+1)}$ is obtained from the previous estimate $\beta^{(r)}$ by

$$\boldsymbol{\beta}^{(r+1)} = \boldsymbol{\beta}^{(r)} + E\{H(\boldsymbol{\beta}^{(r)})\}^{-1} u(\boldsymbol{\beta}^{(r)})$$

where the gradient vector $u(\boldsymbol{\beta}^{(r)})$ is

$$u(\boldsymbol{\beta}^{(r)}) = \left(\frac{\partial \mathcal{L}}{\partial \beta_1}, \ldots, \frac{\partial \mathcal{L}}{\partial \beta_p} \right)' |_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(r)}}$$

and the Hessian matrix $H(\beta^{(r)})$ is denoted by

$$H(\boldsymbol{\beta}^{(r)})_{jk} = \frac{\partial^2 \mathcal{L}}{\partial \beta_j \partial \beta_k} |_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(r)}}$$

for $j, k = 1, \ldots, p$. At the rth iteration is

$$u_j^{(r)} = \frac{\partial \mathcal{L}}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \eta_j} \frac{\partial \eta_i}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \eta_i} x_{ij},$$

and

$$
\begin{aligned}
H_{jk}^{(r)} &= \frac{\partial^2 \mathcal{L}}{\partial\beta_j \partial\beta_k} = \sum_{i=1}^{n} \frac{\partial^2 \ell_i}{\partial\beta_j \partial\beta_k} = \sum_{i=1}^{n} \frac{\partial}{\partial\beta_k}\left(\frac{\partial \ell_i}{\partial\beta_j}\right) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial\beta_k}\left(\frac{\partial \ell_i}{\partial\eta_i} x_{ij}\right) = \sum_{i=1}^{n} \frac{\partial}{\partial\eta_i}\left(\frac{\partial \ell_i}{\partial\eta_i} x_{ij}\right) x_{ik} = \sum_{i=1}^{n} \frac{\partial^2 \ell_i}{\partial\eta_i^2} x_{ij} x_{ik}.
\end{aligned}
$$

We define $s^{(r)} = \frac{\partial \ell_i}{\partial \eta_i}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^r}$ and the diagonal matrix $\boldsymbol{W}^{(r)} = E\{\mathrm{diag}(-\frac{d^2\ell_i}{d\eta_i^2})|\beta^{(r)}\} = \mathrm{var}\{\mathrm{diag}(\frac{d\ell_i}{d\eta_i})|\beta^{(r)}\} = \mathrm{var}(s^{(r)}|\beta^{(r)})$. Then, we can rewrite (4) as

$$
\begin{aligned}
\boldsymbol{\beta}^{(r+1)} &= \boldsymbol{\beta}^{(r)} + (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{s}^{(r)} \\
&= (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X} \boldsymbol{\beta}^{(r)} + (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} (\boldsymbol{W}^{(r)})^{-1} \boldsymbol{s}^{(r)} \\
&= (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \{\boldsymbol{X} \boldsymbol{\beta}^{(r)} + (\boldsymbol{W}^{(r)})^{-1} \boldsymbol{s}^{(r)}\} \\
&= (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{z}^{(r)}.
\end{aligned}
$$

i.e. the score equations for a weighted least squares regression of $\boldsymbol{z}^{(r)}$ on $\boldsymbol{X}$ with weights $\boldsymbol{W}^{(r)}$. Since

$$
s = \frac{\partial \ell_i}{\partial \eta_i} = \frac{d\ell_i}{d\theta_i}\frac{d\theta_i}{d\mu_i}\frac{d\mu_i}{d\eta_i} = \frac{y_i - \mu_i}{\phi V(\mu_i) g'(\mu_i)},
$$

the diagonal matrix $\boldsymbol{W} = \mathrm{diag}(w_i)$ can be written as follows.

$$
w_i = \mathrm{var}(s_i) = \frac{\mathrm{var}(y_i)}{\{\phi V(\mu_i) g'(\mu_i)\}^2} = \frac{\phi V(\mu_i)}{\{\phi V(\mu_i) g'(\mu_i)\}^2} = \frac{1}{\phi V(\mu_i^{(r)})(g'(\mu_i^{(r)}))^2}.
$$

Plugging $\boldsymbol{W}$ and s in $\boldsymbol{z}$,

$$
\boldsymbol{z}^{(r)} = \boldsymbol{\eta}^{(r)} + (\boldsymbol{y} - \boldsymbol{\mu}^{(r)}) g'(\boldsymbol{\mu}^{(r)}). \tag{11}
$$

Consequently we can obtain $\boldsymbol{\eta}^{(r+1)}$ by plugging in (2) via

$$
\boldsymbol{\eta}^{(r+1)} = \boldsymbol{\alpha}^{(r)} \mathbf{1} + \boldsymbol{X} (\boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(r)} \boldsymbol{z}^{(r)}. \tag{12}
$$

Algorithm 2: IRPLS Estimation for GLS in Marx (1996)

[0] Initialize $\boldsymbol{E}_0 = \boldsymbol{X} \in \mathbb{R}^{n \times p}$; $\boldsymbol{f}_0 = \psi(\boldsymbol{y}) \in \mathbb{R}^n$; $\boldsymbol{W} = \left[\phi \boldsymbol{V}(\boldsymbol{\mu})\{g^{'}[\psi(\boldsymbol{y})]\}^2\right]^{-1} \in \mathbb{R}^{n \times n}$

for $\boldsymbol{k} = 1, \ldots, R$ **do**

[1] Define a loading vector $\boldsymbol{r}_k = (\boldsymbol{f}_{k-1}^{\top} \boldsymbol{W} \boldsymbol{E}_{k-1} \boldsymbol{E}_{k-1}^{\top} \boldsymbol{W} \boldsymbol{f}_{k-1})^{0.5} \boldsymbol{E}_{k-1}^{\top} \boldsymbol{W} \boldsymbol{f}_{k-1} \in \mathbb{R}^p$

[2] Define a latent variable $\boldsymbol{t}_k = \boldsymbol{E}_{k-1} \boldsymbol{r}_k \in \mathbb{R}^n$.

[3] Scale $\boldsymbol{t}_k$ with center (= Weighted mean of $\boldsymbol{t}_k$ with $\boldsymbol{W}$) and scale (=SS($\boldsymbol{X}$)).

[4] Define $\boldsymbol{q}_k$ = the coefficient for the WLS regression of $\boldsymbol{f}_{k-1}$ on $\boldsymbol{t}_k$ with weight $\boldsymbol{W}$.

[5] Deflate the response vector $\boldsymbol{f}_k = \boldsymbol{f}_{k-1} - \boldsymbol{t}_k \boldsymbol{q}_k$.

[6] Define $\boldsymbol{E}_k$ = Residuals for the WLS regression of $\boldsymbol{E}_{k-1}$ on $\boldsymbol{t}_k$ with weight $\boldsymbol{W}$.

end for

[7] $\boldsymbol{\eta}$ = Weighted mean of f_0 with weight $\boldsymbol{W}$ + $\sum_{k=1}^{R} \boldsymbol{q}_k \boldsymbol{t}$

[8] $\boldsymbol{W} = \left[\phi \boldsymbol{V}(\boldsymbol{\mu})\{g^{'}(\boldsymbol{\mu})\}^2\right]^{-1} \in \mathbb{R}^{n \times n}$

[9] $\boldsymbol{f}_0 = \boldsymbol{\eta} + \text{diag}\{g(\boldsymbol{\eta})\}\{\boldsymbol{y} - \frac{1}{g(\boldsymbol{\eta})}\}$

[10] Scale $\boldsymbol{E}_0$ with center (= Weighted mean($\boldsymbol{X}$) with weight $\boldsymbol{W}$ and scale (=SS($\boldsymbol{X}$)).

until $\triangle \hat{\eta} < \epsilon$

[11] Choose $s \ni \parallel \boldsymbol{f}_{s+1} \parallel$ small, $s \leq R$.

[12] Glm $\boldsymbol{y}$ on $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_s) \in \mathbb{R}^{n \times s}$.

References

Kolda, T., G. (2006), "Multilinear Operators for Higher-Order Decompositions," *Report SAND2006-2081*.

Marx, B, D. (1996), "Iteratively Reweighted Partial Least Squares Estimation for Generalized Linear Regression," *Technometrics 38*, 374–381.

McCullagh, P. and Nelder, J. A. (1983), *Generalized Linear Models (Monographs on Statistics and Applied Probability)*, London: Chapman and Hall.

Wold, S. (1966), *Estimation of principle components and related models by iterative models iterative least squares*, NY: Academic Press.

Zhou, Hua., L. L. Z. H. (2013), "Tensor Regression with Applications in Neuroimaging Data Analysis," *Journal of the American Statistical Association 108*, 540–552.