Doubly robust pseudo-likelihood for incomplete hierarchical binary data

Peer-reviewed author version

HERMANS, Lisa; IVANOVA, Anna; SOTTO, Cristina; MOLENBERGHS, Geert; VERBEKE, Geert & Kenward, Michael G. (2020) Doubly robust pseudo-likelihood for incomplete hierarchical binary data. In: Statistical modelling, 20 (1) , p. 42 -57.

DOI: 10.1177/1471082X18808611
Handle: http://hdl.handle.net/1942/34576

Doubly Robust Pseudo-likelihood for Incomplete Hierarchical Binary Data

Supplementary Materials

Lisa Hermans[1*], Anna Ivanova[2], Cristina Sotto[3], Geert Molenberghs[1,2], Geert Verbeke[2,1] and Michael G. Kenward[4]

[1] *I-BioStat, Universiteit Hasselt, B-3590 Diepenbeek, Belgium*

[2] *I-BioStat, KU Leuven, B-3000 Leuven, Belgium*

[3] *Janssen Pharmaceutica; B-2340 Beerse, Belgium*

[4] *Luton, United Kingdom*

[*] *lisa.hermans@uhasselt.be*

A General Theory and Concepts on Pseudo-likelihood

To review pseudo-likelihood, we will use the convenient general definition given by Arnold & Strauss (1991). Define S as the set of all 2^n-1 vectors of length n, consisting solely of zeros and ones, with each vector having at least one non-zero entry. Denote by $\boldsymbol{Y}_i^{(s)}$ the sub-vector of $\boldsymbol{Y}_i$ corresponding to the components of s that are non-zero. The associated joint density is $f_s(\boldsymbol{y}_i^{(s)};\boldsymbol{\theta}_i)$. To define a pseudo-likelihood function, one chooses a set $\delta = \{\delta_s | s \in S\}$ of real numbers, with at least one non-zero component. The log of the pseudo-likelihood is then defined as

$$p\ell = \sum_{i=1}^{N} \sum_{s \in S} \delta_s \ln f_s(\boldsymbol{y}_i^{(s)};\boldsymbol{\theta}_i). \tag{A.1}$$

The classical log-likelihood function is found by setting $\delta_s = 1$ if s is the vector consisting solely of ones, and 0 otherwise.

Maximization of Eq. (A.1) can be done, subject to adequate regularity conditions, by solving the pseudo-likelihood (score) equations, which can be obtained by differentiating the logarithmic pseudo-likelihood and equating the resulting derivative to zero. Suppose that $\boldsymbol{\theta}$ is the true parameter. Under suitable regularity conditions (Arnold & Strauss, 1991; Geys, Molenberghs, & Ryan, 1999; Aerts et al., 2002), it can be shown that maximizing Eq. (A.1) produces a consistent and asymptotically normal estimator $\widetilde{\boldsymbol{\theta}}$ so that $\sqrt{N}(\widetilde{\boldsymbol{\theta}}_N - \boldsymbol{\theta})$ converges in distribution to

$$N_p \left[\mathbf{0} \, , \; I_0(\boldsymbol{\theta})^{-1} I_1(\boldsymbol{\theta}) I_0(\boldsymbol{\theta})^{-1} \right] . \tag{A.2}$$

Precise statements are provided in Molenberghs et al. (2011). This result provides an easy way to consistently estimate the asymptotic covariance. The matrix I_0 arises from evaluating the second derivative of $p\ell$ in Eq. (A.1) at the PL estimate. The expectation in I_1 can be replaced by the cross-products of the observed scores. As discussed by Arnold & Strauss (1991), the Cramèr-Rao inequality implies that $I_0^{-1} I_1 I_0^{-1}$ is greater than the inverse of I (the Fisher information matrix for the maximum likelihood case), in the sense that $I_0^{-1} I_1 I_0^{-1} - I^{-1}$ is positive semi-definite. Strict inequality holds if the PL estimator fails to be a function of a minimal sufficient statistic. Geys, Molenberghs, & Ryan (1999) have shown that, in realistic clustered-data settings in toxicology experiments, efficiency loss is often negligible and is certainly justified in view of computational convenience and speed.

As stated earlier, marginal models for non-Gaussian data can become prohibitive when subjected to full maximum likelihood inference, especially with large within-unit replication. le Cessie & van Houwelingen (1991) and Geys, Molenberghs, & Lipsitz (1998) replace the true contribution of a vector of correlated binary data to the full likelihood, written as $f(y_{i1}, \ldots, y_{in_i})$, by the product of all pairwise contributions

$f(y_{ij}, y_{ik})$, $1 \leq j < k \leq n_i$, to obtain a pseudo-likelihood function. Also the term composite likelihood is encountered in this context, but we stick to 'pseudo-likelihood' throughout. Renard, Molenberghs, & Geys (2004) refer to this particular instance of pseudo-likelihood as *pairwise likelihood*. The contribution of the ith subject or cluster to the log pseudo-likelihood then specializes to

$$p\ell_i = \sum_{j<k} \ln f(y_{ij}, y_{ik}), \tag{A.3}$$

if it contains more than one observation. Otherwise, $p\ell_i = f(y_{i1})$. Extension to three-way and higher-order pseudo-likelihood is straightforward, all of which are special cases of Eq. (A.1).

1.1 Pairwise Estimating Equations Under Exchangeability

Molenberghs et al. (2011) proved that under exchangeability, meaning that the distribution of any sub-vector of $\boldsymbol{Y}_i$ is that of any other sub-vector of equal length or a permutation thereof, the estimating equations simplify considerably.

In the general formula

$$\boldsymbol{U}_{\text{CS,dr}} = \sum_{i=1}^{N} \sum_{s \in S} \left[\frac{R_{i,s}}{\pi_{i,s}} \cdot \delta_s \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)o}) + \left(1 - \frac{R_{i,s}}{\pi_{i,s}}\right) \cdot \delta_s E_{\boldsymbol{Y}_i^m | \boldsymbol{Y}_i^o} \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)}) \right], \tag{A.4}$$

the term $E_{\boldsymbol{Y}_i^m | \boldsymbol{Y}_i^o} \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)})$ equals $E_{\boldsymbol{Y}_i^m | \boldsymbol{Y}_i^o}[\boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)o}) + \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)m}) | \boldsymbol{Y}_i^{(s)o}]$. Now, the expectation over the second term can be replaced by $E_{\boldsymbol{Y}_i^{(s)m} | \boldsymbol{Y}_i^{(s)o}} \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)m}) | \boldsymbol{Y}_i^{(s)o}$, due to the full exchangeability and the fact that the score contributions arise from derivatives of sub-vectors of $\boldsymbol{Y}_i$. Under this the conditional expectations vanishes and (A.4) reduces to

$$\boldsymbol{U}_{\text{CS,dr}} = \sum_{i=1}^{N} \sum_{s \in S} \delta_s \boldsymbol{U}_s(\boldsymbol{Y}_i^{(s)o}). \tag{A.5}$$

This makes the naive available case version not only valid, but actually doubly robust. Of course, this is the case only under exchangeability.

B Details on Pseudo-likelihood for Incomplete Data

The general formulation is outlined at the beginning of Section 3 resulting in the estimating equations in Table 1. In this part of the appendix some more detailed calculations on parts of Section 3 can be found.

Let us consider precision estimation. In the singly robust case we must also take into account the uncertainty coming from the parameters of the weight model. The asymptotic variance-covariance matrix can than be estimated using the sandwich estimator in Eq. (3.14). The logistic form of the missingness model equals

$$\pi_i = \prod_{j=2}^{n_i} \left(1 + e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}}\right)^{-1}$$

and

$$\begin{aligned}
\frac{\partial \pi_i}{\boldsymbol{\psi}} &= -\sum_{k=2}^{n_i} \left(\prod_{j=2, j\neq k}^{n_i} (1 + e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}})^{-1} \right) \cdot \frac{e^{\boldsymbol{z}'_{ik}\boldsymbol{\psi}}}{\left(1 + e^{\boldsymbol{z}'_{ik}\boldsymbol{\psi}}\right)^2} \cdot \boldsymbol{z}'_{ik} \\
&= -\sum_{k=2}^{n_i} \boldsymbol{z}'_{ik} \cdot e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}} \cdot \left(1 + e^{\boldsymbol{z}'_{ik}\boldsymbol{\psi}}\right)^{-1} \cdot \prod_{j=2}^{n_i} \left(1 + e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}}\right)^{-1} \\
&= -\pi_i \cdot \left(\sum_{k=2}^{n_i} \boldsymbol{z}'_{ik} \cdot p_{ik} \right)
\end{aligned}$$

with $\boldsymbol{\psi}$ the missingness parameter and $\boldsymbol{z}_{ij}$ stacked with covariates prior to the jth moment.

The parameter $\boldsymbol{\psi}$ is estimated from the weight model, a logistic regression with

$$
\begin{aligned}
L_i &= \prod_{j=2}^{n_i} \frac{e^{R_{ij}\boldsymbol{z}'_{ij}\boldsymbol{\psi}}}{1+e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}}} \\
\ell_i &= \sum_{j=2}^{n_i} \left[R_{ij}\boldsymbol{z}'_{ij}\boldsymbol{\psi} - \ln\left(1+e^{\boldsymbol{z}'_{ij}\boldsymbol{\psi}}\right)\right] \\
\boldsymbol{W}_i &= \sum_{j=2}^{n_i} \boldsymbol{z}_{ij}(R_{ij}-p_{il}) \\
R_{ij} &= \begin{cases} 0 & if \quad j < d_i \\ 1 & if \quad j = d_i \end{cases}
\end{aligned}
$$

The first block of Eq. (3.14), $\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}}$, can be straightforwardly calculated from the Bahadur model with the $\boldsymbol{\psi}$-parameter kept fixed. The same for the weight model, $\frac{\partial \boldsymbol{W}_i}{\partial \boldsymbol{\psi}}$ can be deducted from is logistic structure. More attention and thorough calculations should go to the third part, $\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}}$. In contrast to the other two, this one is not directly computable by software and needs some extra programming. For the three cases respectively:

- CC, sr: $\boldsymbol{V}_i = \frac{\tilde{R}_i}{\pi_i}\sum_{j<k}\boldsymbol{U}_i(y_{ij},y_{ik})$ and $\boldsymbol{W}_i = \sum_{j=2}^{n_i}\boldsymbol{z}_{ij}(R_{ij}-p_{ij})$

$$
\begin{aligned}
\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} &= \frac{\tilde{R}_i}{\pi_i}\sum_{j<k}\frac{\partial \boldsymbol{U}_i(y_{ij},y_{ik})}{\partial \boldsymbol{\theta}} \\
\frac{\partial \boldsymbol{W}_i}{\partial \boldsymbol{\psi}} &= -\sum_{j=2}^{n_i}(\boldsymbol{z}_{ij}\boldsymbol{z}'_{ij})p_{ij}(1-p_{ij}) \\
\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} &= \frac{\tilde{R}_i}{\pi_i}\sum_{j<k}\boldsymbol{U}_i(y_{ij},y_{ik})\left(\sum_{l=2}^{n_i}\boldsymbol{z}'_{il}p_{il}\right)
\end{aligned}
$$

- CP, sr: $\boldsymbol{V}_i = \sum_{j<k<d_i} \frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ij}, y_{ik})$ and $\boldsymbol{W}_i = \sum_{j=2}^{n_i} \boldsymbol{z}_{ij}(R_{ij} - p_{ij})$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} = \sum_{j<k<d_i} \frac{R_{ik}}{\pi_{ik}} \frac{\partial \boldsymbol{U}_i(y_{ij}, y_{ik})}{\partial \boldsymbol{\theta}}$$

$$\frac{\partial \boldsymbol{W}_i}{\partial \boldsymbol{\psi}} = -\sum_{j=2}^{n_i} (\boldsymbol{z}_{ij}\boldsymbol{z}'_{ij}) p_{ij}(1 - p_{ij})$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} = \sum_{j<k<d_i} \frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ij}, y_{ik}) \left(\sum_{l=2}^{k} \boldsymbol{z}'_{il} p_{il} \right)$$

- AC, sr: $\boldsymbol{V}_i = \sum_{j<k} \left[\frac{R_{ij}}{\pi_{ij}} \boldsymbol{U}_i(y_{ij}) + \frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ik}|y_{ij}) \right]$ and $\boldsymbol{W}_i = \sum_{j=2}^{n_i} \boldsymbol{z}_{ij}(R_{ij} - p_{ij})$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} = \sum_{j<k} \left[\frac{R_{ij}}{\pi_{ij}} \frac{\partial \boldsymbol{U}_i(y_{ij})}{\partial \boldsymbol{\theta}} + \frac{R_{ik}}{\pi_{ik}} \frac{\partial \boldsymbol{U}_i(y_{ik}|y_{ij})}{\partial \boldsymbol{\theta}} \right]$$

$$\frac{\partial \boldsymbol{W}_i}{\partial \boldsymbol{\psi}} = -\sum_{j=2}^{n_i} (\boldsymbol{z}_{ij}\boldsymbol{z}'_{ij}) p_{ij}(1 - p_{ij})$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} = \sum_{j<k} \left[\frac{R_{ij}}{\pi_{ij}} \boldsymbol{U}_i(y_{ij}) + \frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ik}|y_{ij}) \right] \left(\sum_{l=2}^{k} \boldsymbol{z}'_{il} p_{il} \right)$$

The doubly robust version is extend with a predictive model for the unobserved responses, given the observed ones. The asymptotic variance-covariance estimator is than outlined in Eq. (3.15), given we use the full expressions in Table 1. For the predictive model a separate Bahadur model is implemented, meaning that a third score equation $T(\boldsymbol{\phi})$ representing the conditional Bahadur model joins the entire score.

Here we focus on the first row of Eq. (3.15), as $\frac{\partial \boldsymbol{W}_i}{\partial \boldsymbol{\psi}}$ is identical as in the single robust case and $\frac{\partial \boldsymbol{T}_i}{\partial \boldsymbol{\phi}}$ follows directly form the conditional Bahadur model. For the expectations in the formulas Eq. (3.11)-(3.12) are used.

Also here for the three cases respectively:

- CC, dr: $\boldsymbol{V}_i = \sum_{j<k} \left[\frac{\tilde{R}_i}{\pi_i} \boldsymbol{U}_i(y_{ij}, y_{ik}) + \left(1 - \frac{\tilde{R}_i}{\pi_i}\right) E_{\boldsymbol{Y}^m|\boldsymbol{y}^o} \boldsymbol{U}_i(y_{ij}, y_{ik}) \right]$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} = \sum_{j<k} \left[\frac{\widetilde{R}_i}{\pi_i} \frac{\partial \boldsymbol{U}_i(y_{ij}, y_{ik})}{\partial \boldsymbol{\theta}} + \left(1 - \frac{\widetilde{R}_i}{\pi_i}\right) \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \frac{\partial \boldsymbol{U}_i(y_{ij}, y_{ik})}{\partial \boldsymbol{\theta}} q(y_{ij}, y_{ik}) \right]$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} = \sum_{j<k} \left[\left(\frac{\widetilde{R}_i}{\pi_i} \boldsymbol{U}_i(y_{ij}, y_{ik}) - \frac{\widetilde{R}_i}{\pi_i} \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \boldsymbol{U}_i(y_{ij}, y_{ik}) q(y_{ij}, y_{ik}) \right) \left(\sum_{l=2}^{n_i} \boldsymbol{z}'_{il} p_{il} \right) \right]$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\phi}} = \sum_{j<k} \left[\left(1 - \frac{\widetilde{R}_i}{\pi_i}\right) \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \boldsymbol{U}_i(y_{ij}, y_{ik}) \left(\frac{\partial q(y_{ij}, y_{ik})}{\partial \boldsymbol{\phi}} \right)' \right]$$

- CP, dr: $\boldsymbol{V}_i = \sum_{j<k<n_i} \left[\frac{R_{ijk}}{\pi_{ijk}} \boldsymbol{U}_i(y_{ij}, y_{ik}) + \left(1 - \frac{R_{ijk}}{\pi_{ijk}}\right) E_{\boldsymbol{Y}^m|\boldsymbol{y}^o} \boldsymbol{U}_i(y_{ij}, y_{ik}) \right]$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} = \sum_{j<k<n_i} \left[\frac{R_{ik}}{\pi_{ik}} \frac{\partial \boldsymbol{U}_i(y_{ij}, y_{ik})}{\partial \boldsymbol{\theta}} + \left(1 - \frac{R_{ik}}{\pi_{ik}}\right) \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \frac{\partial \boldsymbol{U}_i(y_{ij}, y_{ik})}{\partial \boldsymbol{\theta}} q(y_{ij}, y_{ik}) \right]$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} = \sum_{j<k<n_i} \left[\left(\frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ij}, y_{ik}) - \frac{R_{ik}}{\pi_{ik}} \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \boldsymbol{U}_i(y_{ij}, y_{ik}) q(y_{ij}, y_{ik}) \right) \left(\sum_{l=2}^{k} \boldsymbol{z}'_{il} p_{il} \right) \right]$$

$$\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\phi}} = \sum_{j<k<n_i} \left[\left(1 - \frac{R_{ik}}{\pi_{ik}}\right) \sum_{y_{ij}=0}^{1} \sum_{y_{ik}=0}^{1} \boldsymbol{U}_i(y_{ij}, y_{ik}) \left(\frac{\partial q(y_{ij}, y_{ik})}{\partial \boldsymbol{\phi}} \right)' \right]$$

- AC, dr: $\boldsymbol{V}_i = \sum_{j<k} \frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ik}|y_{ij}) + \sum_{j=1}^{d_i-1} \frac{R_{ij}}{\pi_{ij}} \boldsymbol{U}_i(y_{ij})$
  $+ \sum_{j<k} \left(1 - \frac{R_{ik}}{\pi_{ik}}\right) E_{\boldsymbol{Y}^m|\boldsymbol{y}^o} \boldsymbol{U}_i(y_{ik}|y_{ij}) + \sum_{j=1}^{d_i-1} \left(1 - \frac{R_{ij}}{\pi_{ij}}\right) E_{\boldsymbol{Y}^m|\boldsymbol{y}^o} \boldsymbol{U}_i(y_{ij})$

$$\begin{aligned}\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} &= \sum_{j<k} \frac{R_{ik}}{\pi_{ik}} \frac{\partial \boldsymbol{U}_i(y_{ik}|y_{ij})}{\partial \boldsymbol{\theta}} + \sum_{j=1}^{d_i-1} \frac{R_{ij}}{\pi_{ij}} \frac{\partial \boldsymbol{U}_i(y_{ij})}{\partial \boldsymbol{\theta}} \\ &\quad + \sum_{j<k} \left(1 - \frac{R_{ik}}{\pi_{ik}}\right) \sum_{y_{ik}=0}^{1} \frac{\partial \boldsymbol{U}_i(y_{ik}|y_{ij})}{\partial \boldsymbol{\theta}} q(y_{ik}|y_{ij}) + \sum_{j=1}^{d_i-1} \left(1 - \frac{R_{ij}}{\pi_{ij}}\right) \sum_{y_{ij}=0}^{1} \frac{\partial \boldsymbol{U}_i(y_{ij})}{\boldsymbol{\theta}} q(y_{ij}) \\ \frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\psi}} &= \sum_{j<k} \left[\left(\frac{R_{ik}}{\pi_{ik}} \boldsymbol{U}_i(y_{ik}|y_{ij}) - \frac{R_{ik}}{\pi_{ik}} \sum_{y_{ik}=0}^{1} \boldsymbol{U}_i(y_{ik}|y_{ij}) q(y_{ik}|y_{ij}) \right) \left(\sum_{l=2}^{k} \boldsymbol{z}'_{il} p_{il} \right) \right] \\ &\quad + \sum_{j=1}^{d_i-1} \left[\left(\frac{R_{ij}}{\pi_{ij}} \boldsymbol{U}_i(y_{ij}) - \frac{R_{ij}}{\pi_{ij}} \sum_{y_{ij}=0}^{1} \boldsymbol{U}_i(y_{ij}) q(y_{ij}) \right) \left(\sum_{l=2}^{j} \boldsymbol{z}'_{il} p_{il} \right) \right] \\ \frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\phi}} &= \sum_{j<k} \left(1 - \frac{R_{ik}}{\pi_{ik}}\right) \boldsymbol{U}_i(y_{ik}|y_{ij}) \left(\frac{\partial q(y_{ik}|y_{ij})}{\partial \boldsymbol{\phi}} \right)' + \sum_{j=1}^{d_i-1} \left(1 - \frac{R_{ij}}{\pi_{ij}}\right) \boldsymbol{U}_i(y_{ij}) \left(\frac{\partial q(y_{ij})}{\partial \boldsymbol{\phi}} \right)'\end{aligned}$$

All three expressions coincide as expressed in Eq. (3.1). In this case it is not needed to explicitly model the missingness and the entire score reduces to $\boldsymbol{S}_i = (\boldsymbol{V}_i, \boldsymbol{T}_i)$. The asymptotic variance-covariance matrix can be estimated using

$$I_0 = \frac{1}{N}\sum_{i=1}^{N} \begin{pmatrix} \dfrac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\theta}} & \dfrac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{\phi}} \\ \\ \mathbf{0} & \dfrac{\partial \boldsymbol{T}_i}{\partial \boldsymbol{\phi}} \end{pmatrix} \quad \text{and} \quad I_1 = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{S}_i(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\phi}})\boldsymbol{S}_i'(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\phi}}). \qquad \text{(B.1)}$$

Notice that to avoid complex and too long formulas for the derivatives in this section of the appendix, factors representing a deviation with the amount of pairs or pairs with a single observation are omitted. Although these factors most of time cancel when solving the estimating equations for the parameters, they are necessary for the calculation of the standard errors as they get estimated to big otherwise.

C Implementation with SAS

3.1 Full Likelihood Based Bahadur Model

To apply the full likelihood Bahadur model, the original format of the GSA data was used:

PATID	gsabin1	gsabin2	gsabin3	gsabin4	pca0	intercept
1	0	.	.	.	3	1
2	1	.	.	.	5	1
3	1	1	1	0	1	1
4	1	0	.	.	4	1

```
     6          0          .          .          .     4          1
...
```

Further, the NLMIXED procedure of SAS (SAS 9.4) was applied, when all possible profiles were considered. For a theoretical background on the Bahadur model, see Section 3.2. Note, that variable *intercept* is used here only to make it possible to apply the general(.) statement of proc nlmixed. For the first-order correlations, a Toepliz structure was assumed, for the higher-order correlations a general form of the correlations.

```
proc nlmixed data=gsa;
parms beta0=0 beta1=0 beta2=0 beta3=0 rho_1 rho_2=0 rho_3=0 rho_123=0
rho_124=0 rho_134=0 rho_234=0 rho_1234=0;

eta_1 = beta0 + beta1*1 + beta2*1+ beta3*pca0 ;
eta_2 = beta0 + beta1*2 + beta2*4+ beta3*pca0 ;
eta_3 = beta0 + beta1*3 + beta2*9+ beta3*pca0 ;
eta_4 = beta0 + beta1*4 + beta2*16+ beta3*pca0 ;

nu_1=exp(eta_1)/(1+exp(eta_1));
nu_2=exp(eta_2)/(1+exp(eta_2));
nu_3=exp(eta_3)/(1+exp(eta_3));
nu_4=exp(eta_4)/(1+exp(eta_4));

e_1=(gsabin1-nu_1)/sqrt(nu_1*(1-nu_1));
e_2=(gsabin2-nu_2)/sqrt(nu_2*(1-nu_2));
e_3=(gsabin3-nu_3)/sqrt(nu_3*(1-nu_3));
e_4=(gsabin4-nu_4)/sqrt(nu_4*(1-nu_4));
```

```
if (gsabin2 =. and gsabin3 =. and gsabin4 =.)then do;
c=1;
loglik=log(c)+gsabin1*log(nu_1)+(1-gsabin1)*log(1-nu_1);
end; else
if (gsabin2 ne . and gsabin3 =. and gsabin4 =.) then do;
c=1+rho_1*e_1*e_2;
loglik=log(c)+gsabin1*log(nu_1)+(1-gsabin1)*log(1-nu_1)+
gsabin2*log(nu_2)+(1-gsabin2)*log(1-nu_2);
end; else
if (gsabin2 ne . and gsabin3 ne . and gsabin4 =.) then do;
c=1+rho_1*e_1*e_2+rho_2*e_1*e_3+rho_1*e_2*e_3+rho_123*e_1*e_2*e_3;
loglik=log(c)+gsabin1*log(nu_1)+(1-gsabin1)*log(1-nu_1)+
gsabin2*log(nu_2)+(1-gsabin2)*log(1-nu_2)+gsabin3*log(nu_3)+
(1-gsabin3)*log(1-nu_3);
end; else
if (gsabin2 ne . and gsabin3 ne . and gsabin4 ne .) then do;
c=1+rho_1*e_1*e_2+rho_2*e_1*e_3+rho_3*e_1*e_4+rho_1*e_2*e_3+rho_2*e_2*e_4+
rho_1*e_3*e_4+rho_123*e_1*e_2*e_3+rho_124*e_1*e_2*e_4+rho_134*e_1*e_3*e_4
+rho_234*e_2*e_3*e_4+rho_1234*e_1*e_2*e_3*e_4;
loglik=log(c)+gsabin1*log(nu_1)+(1-gsabin1)*log(1-nu_1)+gsabin2*log(nu_2)+
(1-gsabin2)*log(1-nu_2)+gsabin3*log(nu_3)+(1-gsabin3)*log(1-nu_3)+
gsabin4*log(nu_4)+(1-gsabin4)*log(1-nu_4);
end;
model intercept ~ general(loglik);
run;
```

3.2 Pairwise Likelihood Based Model

In this section we list some examples of data formatting and the code to fit the model to different cases of the estimating equation. In these examples, some cases are omitted due to their mutual similarity. Complete cases and complete pairs are different in their inverse probability weights: for complete cases, the weights are calculated based on the probability of a patient to be completely observed, for complete pairs the probability to be observed until a certain time-point. After the model is fitted, the principle of the sandwich-type robust variance estimation will be applied to obtain precision estimates.

3.2.1 Naive Case

For modeling using naive estimating equations, all possible pairs from the sequence of GSA measurement per patient were considered. If the response measurement of a pair was missing, the 999 code was used instead. As result, the data was re-formated as follows:

PATID	pca0	intercept	responsej	timej	timej_2	responsek	timek	timek_2
1	3	1	0	1	1	999	2	4
1	3	1	0	1	1	999	3	9
1	3	1	0	1	1	999	4	16
2	5	1	1	1	1	999	2	4
2	5	1	1	1	1	999	3	9
2	5	1	1	1	1	999	4	16
3	1	1	1	1	1	1	2	4
3	1	1	1	1	1	1	3	9
3	1	1	1	1	1	0	4	16

3	1	1	1	2	4	1	3	9
3	1	1	1	2	4	0	4	16
3	1	1	1	3	9	0	4	16
4	4	1	1	1	1	0	2	4
4	4	1	1	1	1	999	3	9
4	4	1	1	1	1	999	4	16
4	4	1	0	2	4	999	3	9
4	4	1	0	2	4	999	4	16
6	4	1	0	1	1	999	2	4
6	4	1	0	1	1	999	3	9
6	4	1	0	1	1	999	4	16

...

For the available cases the whole data was used, for the complete pairs and complete cases the corresponding subject selection. Herewith, the code for the available cases:

```
proc nlmixed data=model_gsa_ac_naive;
parms beta0=0 beta1=0 beta2=0 beta3=0 rho_1=0 rho_2=0 rho_3=0;
eta_j = beta0 + beta1*timej + beta2*timej_2+ beta3*pca0 ;
eta_k = beta0 + beta1*timek + beta2*timek_2+ beta3*pca0 ;
nu_j = exp(eta_j)/(1+exp(eta_j));
nu_k = exp(eta_k)/(1+exp(eta_k));

if (timej=1 and timek=2) or (timej=2 and timek=3) or (timej=3 and timek=4)
then mu11 = nu_j*nu_k + rho_1*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=3) or (timej=2 and timek=4)
then mu11 = nu_j*nu_k + rho_2*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=4)
then mu11 = nu_j*nu_k + rho_3*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k));
```

```
mu10 = nu_j - mu11;
mu01 = nu_k - mu11;
mu00 = 1 - (mu11+mu10+mu01);

if responsej = 1 and responsek = 1 then loglik=log(mu11); else
if responsej = 1 and responsek = 0 then loglik=log(mu10); else
if responsej = 1 and responsek = 999 then loglik=log(mu11 + mu10); else
if responsej = 0 and responsek = 1 then loglik=log(mu01); else
if responsej = 0 and responsek = 0 then loglik=log(mu00); else
if responsej = 0 and responsek = 999 then loglik=log(mu00+mu01);

model intercept ~ general(loglik);
run;
```

3.2.2 Singly Robust Case

First, a dropout model with previous measurement was considered, using the data in the following format and the corresponding SAS code:

PATID	pca0	ID_AC	time	GSAbin	dropout	prev
1	3	1	1	0	0	.
1	3	1	2	.	1	0
1	3	1	3	.	1	.
1	3	1	4	.	1	.
2	5	2	1	1	0	.
2	5	2	2	.	1	1
2	5	2	3	.	1	.
2	5	2	4	.	1	.

```
3      1       3       1       1       0       .
3      1       3       2       1       0       1
3      1       3       3       1       0       1
3      1       3       4       0       0       1
```

...

```
proc nlmixed data=model_weights;
parms gamma0=0 gamma1=0 gamma2=0;
eta = gamma0 + gamma1*prev + gamma2*pca0 ;
nu = exp(eta)/(1+exp(eta));
if dropout = 1 then loglik=log(nu); else
if dropout = 0 then loglik=log(1-nu);
model dropout ~ general(loglik);
run;
```

Based on probabilities estimated with the dropout model, the inverse probabilities weights were calculated. So, the data for the analysis of the available cases are as follows:

PATID	pca0	intercept	responsej	wij	timej	timej_2	responsek	wik	timek	timek_2
1	3	1	0	1.00000	1	1	999	1.50253	2	4
1	3	1	0	1.00000	1	1	999	.	3	9
1	3	1	0	1.00000	1	1	999	.	4	16
2	5	1	1	1.00000	1	1	999	1.25292	2	4
2	5	1	1	1.00000	1	1	999	.	3	9
2	5	1	1	1.00000	1	1	999	.	4	16
3	1	1	1	1.00000	1	1	1	1.16671	2	4
3	1	1	1	1.00000	1	1	1	1.36121	3	9
3	1	1	1	1.00000	1	1	0	1.58814	4	16

```
     3   1      1         1     1.16671   2      4         1    1.36121   3       9
     3   1      1         1     1.16671   2      4         0    1.58814   4      16
     3   1      1         1     1.36121   3      9         0    1.58814   4      16
...
```

The code for the available cases is as follows:

```
proc nlmixed data=model_gsa_wi_ac;
parms beta0=0 beta1=0 beta2=0 beta3=0 rho_1=0 rho_2=0 rho_3=0;
eta_j = beta0 + beta1*timej + beta2*timej_2+ beta3*pca0 ;
eta_k = beta0 + beta1*timek + beta2*timek_2+ beta3*pca0 ;
nu_j = exp(eta_j)/(1+exp(eta_j));
nu_k = exp(eta_k)/(1+exp(eta_k));

if (timej=1 and timek=2) or (timej=2 and timek=3) or (timej=3 and timek=4)
then mu11 = nu_j*nu_k + rho_1*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=3) or (timej=2 and timek=4)
then mu11 = nu_j*nu_k + rho_2*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=4)
then mu11 = nu_j*nu_k + rho_3*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k));

mu10 = nu_j - mu11;
mu01 = nu_k - mu11;
mu00 = 1 - (mu11+mu10+mu01);

if responsek = 1 and responsej = 1 then loglik=wik*log(mu11/(mu11 + mu10)); else
if responsek = 0  and responsej = 1 then loglik=wik*log(mu10/(mu11 + mu10)); else
if responsek = 1 and responsej = 0 then loglik=wik*log(mu01/(mu01 + mu00)); else
if responsek = 0  and responsej = 0 then loglik=wik*log(mu00/(mu01 + mu00)); else

if responsej = 1 and responsek = 999 then loglik= wij*log(mu11 + mu10); else
```

```
if responsej = 0 and responsek = 999 then loglik= wij*log(mu00 + mu01);


model intercept ~ general(loglik);
run;
```

As for naive case, the data and the code was adopted for the complete pairs and complete cases.

3.2.3 Doubly Robust Case

The inverse probability weights were calculated using the same way as for the singly robust case.

To model the expectations in doubly robust case, a "help" Bahadur model was applied. To model the marginal probabilities, the expression for the linear predictors were defined at different time-points. To model pairs, assuming that all previous measurements are available, the "history"parameters were used taking as reference for the history the response with the lower value of the time point (e.g., in pair Y2 - Y3 we used only gsabin1 as the history covariate). If some of the measurements are missing, the corresponding term in the model will be omitted.

```
proc nlmixed data=model_gsa_wi_AC;
parms kappa0=0 kappa1=0 kappa2=0 kappa3=0 omega1=0 omega2=0 omega3=0 tau_1=0 tau_2=0 tau_3=0;


if gsabin2 ne . then do;
if (timej=1 and timek=2) or (timej=1 and timek=3) or (timej=1 and timek=4) then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 ;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 ;
end; else
```

```
if timej=2 and timek=3 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega1*gsabin1;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega2*gsabin1;
end; else
if timej=2 and timek=4 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega1*gsabin1;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega3*gsabin1;
end; else
if timej=3 and timek=4 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega2*gsabin1 + omega1*gsabin2;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega3*gsabin1 + omega2*gsabin2;
end;
end;

if gsabin2 = . then do;
if (timej=1 and timek=2) or (timej=1 and timek=3) or (timej=1 and timek=4) then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 ;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 ;
end; else
if timej=2 and timek=3 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega1*gsabin1;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega2*gsabin1;
end; else
if timej=2 and timek=4 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega1*gsabin1;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega3*gsabin1;
end; else
if timej=3 and timek=4 then do;
eta_j = kappa0 + kappa1*timej + kappa2*timej_2+ kappa3*pca0 + omega2*gsabin1;
eta_k = kappa0 + kappa1*timek + kappa2*timek_2+ kappa3*pca0 + omega3*gsabin1;
```

```
end;
end;

q_j = exp(eta_j)/(1+exp(eta_j));
q_k = exp(eta_k)/(1+exp(eta_k));

if (timej=1 and timek=2) or (timej=2 and timek=3) or (timej=3 and timek=4)
then mu11 = q_j*q_k + tau_1*sqrt(q_j*(1-q_j)*q_k*(1-q_k)); else
if (timej=1 and timek=3) or (timej=2 and timek=4)
then mu11 = q_j*q_k + tau_2*sqrt(q_j*(1-q_j)*q_k*(1-q_k)); else
if (timej=1 and timek=4)
then mu11 = q_j*q_k + tau_3*sqrt(q_j*(1-q_j)*q_k*(1-q_k));

mu10 = q_j - mu11;
mu01 = q_k - mu11;
mu00 = 1 - (mu11+mu10+mu01);

if responsej = 1 and responsek = 1 then loglik=log(mu11); else
if responsej = 1 and responsek = 0 then loglik=log(mu10); else
if responsej = 0 and responsek = 1 then loglik=log(mu01); else
if responsej = 0 and responsek = 0 then loglik=log(mu00);

if responsej = 1 and responsek = 999 then loglik= log(mu11 + mu10); else
if responsej = 0 and responsek = 999 then loglik= log(mu00 + mu01);

model intercept ~ general(loglik);
run;
```

Further, the pairs are complemented with expectations. The data for the analysis looks as follows:

		i	r		r								
		n	e		e								
		t	s	t	s		t						
		e	p	i	p		i						
P		r	o	t	m	o	t	m					
A	p	c	n	i	e	n	i	e					
T	c	e	s	m	j	s	m	k	w	w	q	q	q	q
I	a	p	e	e	_	e	e	_	i	i	1	1	0	0
D	0	t	j	j	2	k	k	2	j	k	1	0	1	0
1	3	1	0	1	1	999	2	4	1.00000	1.50253	0.65036	0.16425	0.09806	0.08733
1	3	1	0	1	1	999	3	9	1.00000	1.50253	0.65232	0.16229	0.09357	0.09182
1	3	1	0	1	1	999	4	16	1.00000	1.50253	0.70389	0.11072	0.10458	0.08081
2	5	1	1	1	1	999	2	4	1.00000	1.25292	0.55410	0.19757	0.11795	0.13039
2	5	1	1	1	1	999	3	9	1.00000	1.25292	0.55665	0.19501	0.11244	0.13590
2	5	1	1	1	1	999	4	16	1.00000	1.25292	0.61516	0.13650	0.12893	0.11940
3	1	1	1	1	1	1	2	4	1.00000	1.16671	0.73422	0.13026	0.07777	0.05775
3	1	1	1	1	1	1	3	9	1.00000	1.36121	0.73566	0.12882	0.07427	0.06125
3	1	1	1	1	1	0	4	16	1.00000	1.58814	0.77841	0.08607	0.08129	0.05423
3	1	1	1	2	4	1	3	9	1.16671	1.36121	0.77933	0.07623	0.09917	0.04526
3	1	1	1	2	4	0	4	16	1.16671	1.58814	0.79464	0.06092	0.10095	0.04349
3	1	1	1	3	9	0	4	16	1.36121	1.58814	0.86708	0.04133	0.06863	0.02296

...

The code for available cases is as follows:

```
proc nlmixed data=model_gsa_wi_qs_ac;
parms beta0=0 beta1=0 beta2=0 beta3=0 rho_1=0 rho_2=0 rho_3=0;
eta_j = beta0 + beta1*timej + beta2*timej_2+ beta3*pca0 ;
eta_k = beta0 + beta1*timek + beta2*timek_2+ beta3*pca0 ;
```

```
nu_j = exp(eta_j)/(1+exp(eta_j));
nu_k = exp(eta_k)/(1+exp(eta_k));

if (timej=1 and timek=2) or (timej=2 and timek=3) or (timej=3 and timek=4)
then mu11 = nu_j*nu_k + rho_1*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=3) or (timej=2 and timek=4)
then mu11 = nu_j*nu_k + rho_2*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=4)
then mu11 = nu_j*nu_k + rho_3*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k));

mu10 = nu_j - mu11;
mu01 = nu_k - mu11;
mu00 = 1 - (mu11+mu10+mu01);

Exp_cond1=q11/(q11+q10)*log(mu11/(mu11+mu10))+q10/(q11+q10)*log(mu10/(mu11+mu10));
/*Responsej=1*/
Exp_cond0=q01/(q01+q00)*log(mu01/(mu01+mu00))+q00/(q01+q00)*log(mu00/(mu01+mu00));
/*Responsej=0*/


if responsek = 1 and responsej = 1 then  loglik=wik*log(mu11/(mu11 + mu10))
+ (1-wik)*Exp_cond1; else
if responsek = 0  and responsej = 1 then loglik=wik*log(mu10/(mu11 + mu10))
+ (1-wik)*Exp_cond1; else
if responsek = 1 and responsej = 0 then loglik=wik*log(mu01/(mu01 + mu00))
+ (1-wik)*Exp_cond0; else
if responsek = 0  and responsej = 0 then loglik=wik*log(mu00/(mu01 + mu00))
+ (1-wik)*Exp_cond0; else

if responsej = 1 and responsek = 999 then loglik= wij*log(mu11 + mu10) + (1-wik)*Exp_cond1
```

```
                        + (1-wij)*log(q11+q10); else
if responsej = 0 and responsek = 999 then loglik= wij*log(mu00 + mu01)+ (1-wik)*Exp_cond0
                        + (1-wij)*log(q01+q00);

model intercept ~ general(loglik);
run;
```

After selecting the data for the complete pairs, the model can be fitted using the following program:

```
proc nlmixed data=model_gsa_wi_qs_cp;
parms beta0=0 beta1=0 beta2=0 beta3=0 rho_1=0 rho_2=0 rho_3=0;
eta_j = beta0 + beta1*timej + beta2*timej_2+ beta3*pca0 ;
eta_k = beta0 + beta1*timek + beta2*timek_2+ beta3*pca0 ;
nu_j = exp(eta_j)/(1+exp(eta_j));
nu_k = exp(eta_k)/(1+exp(eta_k));

if (timej=1 and timek=2) or (timej=2 and timek=3) or (timej=3 and timek=4)
then mu11 = nu_j*nu_k + rho_1*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=3) or (timej=2 and timek=4)
then mu11 = nu_j*nu_k + rho_2*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k)); else
if (timej=1 and timek=4)
then mu11 = nu_j*nu_k + rho_3*sqrt(nu_j*(1-nu_j)*nu_k*(1-nu_k));

mu10 = nu_j - mu11;
mu01 = nu_k - mu11;
mu00 = 1 - (mu11+mu10+mu01);

Exp_joint=q11*log(mu11)+q10*log(mu10)+q01*log(mu01)+q00*log(mu00);
if responsej = 1 and responsek = 1 then loglik=wik*log(mu11)+(1-wik)*Exp_joint; else
if responsej = 1 and responsek = 0 then loglik=wik*log(mu10)+(1-wik)*Exp_joint; else
```

```
if responsej = 0 and responsek = 1 then loglik=wik*log(mu01)+(1-wik)*Exp_joint; else
if responsej = 0 and responsek = 0 then loglik=wik*log(mu00)+(1-wik)*Exp_joint;


model intercept ~ general(loglik);
run;
```

And for the selected complete cases data, similar code will be applied to fit the model.