Non-Monotone GLS Simulation

Caleb Leedy

31 January 2024

Introduction

The goal of this report is to have a good idea about the effectiveness of using GLS estimation to combine nonmonotone missing data. This document defines the GLS estimators compares them to other estimators in several simulation studies.

Non-Monotone Missingness and GLS Estimation

Non-Monotone Missingness

Let $Z = (X, Y_1, Y_2)'$. We want to estimate the parameter $\theta = E[g(Z)]$ for some integrable function g where we may not always observe Y_1 and Y_2. Define segments that contain observations in which the same variables are observed as in Table 1.

Table 1: This table identifies which variables are observed in each segment. Since X is always observed, the subscript for each segment identifies which of variables Y_1 and Y_2 are in the segment based on the position of a 1.

Segment	Variables Observed
A_{00}	X
A_{10}	X, Y_1
A_{01}	X, Y_2
A_{11}	X, Y_1, Y_2

We can also define a nested segment structure defined in Table 2.

Table 2: This table identifies which variables are observed in each segment. Since X is always observed, the subscript for each segment identifies which of variables Y_1 and Y_2 are in the segment based on the position of a 1.

Segment	Variables Observed
A_{++}	X
A_{1+}	X, Y_1
A_{+1}	X, Y_2
A_{11}	X, Y_1, Y_2

The difference between these segments is the way in which they are nested. In Table 1, $A_{00} \perp A_{10} \perp A_{01} \perp A_{11}$ because each observation can only be in one of these segments. In Table 2, we have $A_{11} \subseteq A_{1+} \subseteq A_{++}$ and $A_{11} \subseteq A_{+1} \subseteq A_{++}$. If we define the entire dataset to be A then we have $A = A_{00} \cup A_{10} \cup A_{01} \cup A_{11}$ from Table 1 and $A = A_{++}$ from Table 2.

GLS Estimation

To conduct GLS estimation we define two estimators, $\hat{\theta}_1$ and $\hat{\theta}_2$. Let δ define the inclusion indicator into a particular segment and π be the probability of observing $\delta = 1$. We assume that the selection probabilities π are known and iid for each observation. Then[1], define $\theta_0 = E[g_0(X)]$, $\theta_1 = E[g_1(X, Y_1)]$, and $\theta_1 = E[g_2(X, Y_2)]$. Let n be the number of observations and define the first GLS estimator $\hat{\theta}_1$ in Equation 1.

$$\hat{\theta}_1 = \begin{bmatrix} n^{-1}\sum_{i=1}^n g_0(x_i) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{10}+\delta_{11}}{\pi_{10}+\pi_{11}} g_0(x_i) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{10}+\delta_{11}}{\pi_{10}+\pi_{11}} g_1(x_i, y_{1i}) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{01}+\delta_{11}}{\pi_{01}+\pi_{11}} g_0(x_i) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{01}+\delta_{11}}{\pi_{01}+\pi_{11}} g_1(x_i, y_{2i}) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{11}}{\pi_{11}} g_0(x_i) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{11}}{\pi_{11}} g_1(x_i, y_{1i}) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{11}}{\pi_{11}} g_2(x_i, y_{2i}) \\ n^{-1}\sum_{i=1}^n \frac{\delta_{11}}{\pi_{11}} g(x_i, y_{1i}, y_{2i}) \end{bmatrix}. \tag{1}$$

[1]We follow the idea from Dr. Kim's note ``Efficient estimation under non-monotone missing data with planned missingness''

This makes sense as a GLS estimator because we can express this as solving Equation 2.

$$\hat{\theta}_1 = Z\tilde{\theta} + e_1 \tag{2}$$

In this case we have,

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tilde{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ E[g(Z)] \end{bmatrix} \text{ and } E[e_1] = 0.$$

To compute the GLS estimator we just need to compute $\text{Var}(e_1)$. For a proof of the specific entries of $\text{Var}(e_1) := V_1$ see Appendix A. This covariance matrix can be expressed as

$$V_1 = \begin{bmatrix} V_{1A} & V_{1B} \\ V_{1B}' & V_{1C} \end{bmatrix} - E[g]^2 1_9 1_9'.$$

where

$$V_{1A} = \begin{bmatrix} E[g_0^2] & E[g_0^2] & E[g_0^2] & E[g_0^2] & E[g_0^2] \\ E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}} E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}} E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] \\ E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}} E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}} E[g_1^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_1 g_2] \\ E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}} E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}} E[g_0^2] \\ E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_0^2] & \frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})} E[g_1 g_2] & \frac{1}{\pi_{01}+\pi_{11}} E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}} E[g_2^2] \end{bmatrix}$$

$$V_{1B} = \begin{bmatrix} E[g_0^2] & E[g_0^2] & E[g_0^2] & E[g_0^2] \\ \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] \\ \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_1^2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_1g_2] & \frac{1}{\pi_{10}+\pi_{11}}E[g_1^2] \\ \frac{1}{\pi_{01}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_0^2] \\ \frac{1}{\pi_{10}+\pi_{11}}E[g_0^2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_1g_2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_2^2] & \frac{1}{\pi_{01}+\pi_{11}}E[g_2^2] \end{bmatrix}$$

$$V_{1C} = \begin{bmatrix} \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_0^2] \\ \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_1^2] & \frac{1}{\pi_{11}}E[g_1g_2] & \frac{1}{\pi_{11}}E[g_1^2] \\ \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_1g_2] & \frac{1}{\pi_{11}}E[g_2^2] & \frac{1}{\pi_{11}}E[g_2^2] \\ \frac{1}{\pi_{11}}E[g_0^2] & \frac{1}{\pi_{11}}E[g_1^2] & \frac{1}{\pi_{11}}E[g_2^2] & \frac{1}{\pi_{11}}E[g^2] \end{bmatrix}$$

This yields an estimator $\hat{\theta} = (\hat{g}_0(X), \hat{g}_1(X, Y_1), \hat{g}_2(X, Y_2), \hat{g}(Z))$. To get the GLS estimator we use,

$$\hat{\theta}_{GLS} = (0, 0, 0, 1)'\hat{\theta}.$$

Likewise, we can construct a different GLS estimator which is a weighted average of components from Equation 3.

$$\hat{\theta}_2 = \begin{bmatrix} n^{-1}\sum_{i=1}^{n} \frac{\delta_{00}}{\pi_{00}} g_0(x_i) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{10}}{\pi_{10}} g_0(x_i) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{10}}{\pi_{10}} g_1(x_i, y_{1i}) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{01}}{\pi_{01}} g_0(x_i) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{01}}{\pi_{01}} g_1(x_i, y_{2i}) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{11}}{\pi_{11}} g_0(x_i) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{11}}{\pi_{11}} g_1(x_i, y_{1i}) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{11}}{\pi_{11}} g_2(x_i, y_{2i}) \\ n^{-1}\sum_{i=1}^{n} \frac{\delta_{11}}{\pi_{11}} g(x_i, y_{1i}, y_{2i}) \end{bmatrix}. \tag{3}$$

This is the approach that Dr. Fuller and I have been taking. Then,

$$\hat{\theta}_2 = 1_9\theta + e_2 \text{ where } E[e_2] = 0.$$

We still need to compute the covariance matrix, $\mathrm{Var}(e_2)$. For a proof of the result of each entry of the covariance matrix see Appendix A.

We can compute $V_2 = \mathrm{Var}(e_2)$ as,

$$V_2 = \begin{bmatrix} V_{2,1} & 0 & 0 & 0 \\ 0 & V_{2,2} & 0 & 0 \\ 0 & 0 & V_{2,3} & 0 \\ 0 & 0 & 0 & V_{2,4} \end{bmatrix} - E[g]^2 1_4 1_4'$$

where

$$V_{2,1} = \left(\frac{1}{\pi_{00}}\right) E[E[g \mid X]^2],$$

$$V_{2,2} = \left(\frac{1}{\pi_{10}}\right) \begin{bmatrix} E[E[g \mid X]^2] & E[E[g \mid X]^2] \\ E[E[g \mid X]^2] & E[E[g \mid X, Y_1]^2] \end{bmatrix},$$

$$V_{2,3} = \left(\frac{1}{\pi_{01}}\right) \begin{bmatrix} E[E[g \mid X]^2] & E[E[g \mid X]^2] \\ E[E[g \mid X]^2] & E[E[g \mid X, Y_2]^2] \end{bmatrix},$$

and

$$V_{2,4} = \left(\frac{1}{\pi_{11}}\right) \begin{bmatrix} E[E[g \mid X]^2] & E[E[g \mid X]^2] & E[E[g \mid X]^2] & E[E[g \mid X]^2] \\ E[E[g \mid X]^2] & E[E[g \mid X, Y_1]^2] & E[E[g \mid X, Y_1]E[g \mid X, Y_2]] & E[E[g \mid X, Y_1]^2] \\ E[E[g \mid X]^2] & E[E[g \mid X, Y_1]E[g \mid X, Y_2]] & E[E[g \mid X, Y_2]^2] & E[E[g \mid X, Y_2]^2] \\ E[E[g \mid X]^2] & E[E[g \mid X, Y_1]^2] & E[E[g \mid X, Y_2]^2] & E[g^2] \end{bmatrix}.$$

Then the GLS estimators are $\hat{\theta}_{1,GLS} = (0,0,0,1)'(Z'V_1^{-1}Z)^{-1}ZV_1^{1}\hat{\theta}_1$, and $\hat{\theta}_{2,GLS} = (1_9'V_2^{-1}1_9)^{-1}1_9'V_2^{-1}\hat{\theta}_2$.

Simulation Studies

We use the following simulation setup

$$
\begin{aligned}
\begin{bmatrix} x \\ e_1 \\ e_2 \end{bmatrix} &\overset{ind}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \rho \\ 0 & \rho & 1 \end{bmatrix} \right) \\
y_1 &= x + e_1 \\
y_2 &= \mu + x + e_2
\end{aligned}
$$

This yields outcome variables Y_1 and Y_2 that are correlated both with X and additionally with each other. To generate the missingness pattern, we draw from a categorical distribution with selection probabilities $(\pi_{00}, \pi_{10}, \pi_{01}, \pi_{11})$. Each selection probability, π_{j_1,j_2} indicates the probability that an observation is selected into a segment A_{j_1,j_2} where A_{00} indicates that only X is observed, A_{10} means that X and Y_1 are observed, A_{01} has only X and Y_2 observed, and A_{11} observes X, Y_1, and Y_2.

In addition to varying the values of the parameters μ and ρ, there are two main factors of the simulation that change: the distribution of the categories and the parameter of interest, θ. In a **balanced** distribution, we have the following selection probabilities: $\pi_{00} = 0.2$, $\pi_{10} = 0.2$, $\pi_{01} = 0.2$, and $\pi_{11} = 0.4$. In the **unbalanced** distribution we have $\pi_{00} = 0.3$, $\pi_{10} = 0.4$, $\pi_{01} = 0.1$, and $\pi_{11} = 0.2$. The two types of θ values that we consider are a linear estimate, $\theta = E[g(Z)] = E[Y_2]$ and a non-linear estimate, $\theta = E[g(Z)] = E[Y_1^2 Y_2]$.

There are several algorithms for comparison which are defined as the following:

$$
\begin{aligned}
Oracle &= n^{-1} \sum_{i=1}^{n} g(Z_i) \\
CC &= \frac{\sum_{i=1}^{n} \delta_{11} g(Z_i)}{\sum_{i=1}^{n} \delta_{11}} \\
IPW &= \sum_{i=1}^{n} \frac{\delta_{11}}{\pi_{11}} g(Z_i)
\end{aligned}
$$

Furthermore, we include four existing estimators that we have already proposed in the past: WLS, Prop, PropInd, and SemiDelta.

The estimator WLS is a a weight least square estimator derived in the following manner. We now consider a normal model:

$$\begin{pmatrix} x_i \\ e_{1i} \\ e_{2i} \end{pmatrix} \overset{ind}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma_{11} & \sigma_{12} \\ 0 & \sigma_{12} & \sigma_{22} \end{bmatrix}, \right)$$

and define $y_{1i} = x_i + e_{1i}$ and $y_{2i} = \mu + x_i e_{2i}$. Then $b_1 = b_2 = 1$. We define $\bar{z}_k^{(ij)}$ as the mean of y_k in segment A_{ij}. This means that we have means $\bar{z}_1^{(11)}$, $\bar{z}_2^{(11)}$, $\bar{z}_1^{(10)}$, and $\bar{z}_2^{(01)}$. Let $W = [\bar{z}_1^{(11)}, \bar{z}_2^{(11)}, \bar{z}_1^{(10)}, \bar{z}_2^{(01)}]'$, then for $n_{ij} = |A_{ij}|$, we have

$$Z - M\mu \sim N(\vec{0}, V)$$

where

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } V = \begin{bmatrix} \frac{\sigma_{11}}{n_{11}} & \frac{\sigma_{12}}{n_{11}} & 0 & 0 \\ \frac{\sigma_{12}}{n_{11}} & \frac{\sigma_{22}}{n_{11}} & 0 & 0 \\ 0 & 0 & \frac{\sigma_{11}}{n_{10}} & 0 \\ 0 & 0 & 0 & \frac{\sigma_{22}}{n_{01}} \end{bmatrix}.$$

Thus, the BLUE for $\mu = [\mu_1, \mu_2]'$ is

$$\hat{\mu} = (M'V^{-1}M)^{-1}M'V^{-1}W.$$

Hence, WLS is μ_2 as $g(X, Y_1, Y_2) = Y_2$.

The remaining three estimators are drived from the following expression where the values for β are provided in Table 3. These are good estimators to compare to because Prop is the original proposed estimator. PropInd has the same form as Prop except the values for β is different. PropInd shares the same values of β as all of the new models with constraints. SemiDelta is useful because it is the best estimator in general so far.

$$\hat{\theta} = \frac{\delta_{11}}{\pi_{11}} g(Z) + \beta_0(\delta, c_0)E[g(Z) \mid X] + \beta_1(\delta, c_1)E[g(Z) \mid X, Y_1] + \beta_2(\delta, c_2)E[g(Z) \mid X, Y_2].$$

Table 3: This table displays the values of β for different estimator types.

Estimator	$\beta_0(\delta, c_0)$	$\beta_1(\delta, c_1)$
Prop	$\left(1 - \frac{(\delta_{10}+\delta_{11})}{(\pi_{10}+\pi_{11})} - \frac{(\delta_{01}+\delta_{11})}{(\pi_{01}+\pi_{11})} + \frac{\delta_{11}}{\pi_{11}}\right)$	$\left(\frac{\delta_{10}+\delta_{11}}{\pi_{10}+\pi_{11}} - \frac{\delta_{11}}{\pi_{11}}\right)$
PropInd	$\left(1 - \frac{(\delta_{10})}{(\pi_{10})} - \frac{(\delta_{01})}{(\pi_{01})} + \frac{\delta_{11}}{\pi_{11}}\right)$	$\left(\frac{\delta_{10}}{\pi_{10}} - \frac{\delta_{11}}{\pi_{11}}\right)$
SemiDelta	$c_0\left(\frac{\delta_{11}}{\pi_{11}} - \frac{\delta_{00}}{\pi_{00}}\right)$	$c_1\left(\frac{\delta_{11}}{\pi_{11}} - \frac{\delta_{10}}{\pi_{10}}\right)$

The different estimators are evaluated on 1000 Monte Carlo runs. We test the estimators for bias and observe their standard deviation.

Table 4: Results from the balanced linear simulation. True values: $\theta = 5, \rho = 0.5$. The test conducted for the T-statistic and P-value is a two sample test to see if the estimator is unbiased. This simulation uses $Y_1 = x + \varepsilon_1$, $Y_2 = \mu + x + \varepsilon_2$ where $X \sim N(0,1)$ and $(\varepsilon_1, \varepsilon_2)$ come from a mean zero bivariate normal distribution with unit variance and covariance ρ. The segments are unbalanced with: $\pi_{11} = 0.4$, $\pi_{10} = 0.2$, $\pi_{01} = 0.2$, and $\pi_{00} = 0.2$.

Algorithm	Bias	SD	Tstat	Pval
Oracle	-0.002	0.045	-1.669	0.048
CC	0.000	0.058	-0.208	0.417
IPW	0.011	0.212	1.687	0.046
WLS	-0.001	0.040	-0.743	0.229
Prop	-0.002	0.052	-1.179	0.119
PropOpt	-0.002	0.052	-1.149	0.125
SemiDelta	-0.002	0.052	-1.162	0.123
GLS1	-0.018	0.343	-1.681	0.047
GLS2	-0.001	0.052	-0.848	0.198

Table 5: Results from the balanced non-linear simulation. True values: $\theta = 10, \rho = 0.5$. The test conducted for the T-statistic and P-value is a two sample test to see if the estimator is unbiased. This simulation uses $Y_1 = x + \varepsilon_1$, $Y_2 = \mu + x + \varepsilon_2$ where $X \sim N(0, 1)$ and $(\varepsilon_1, \varepsilon_2)$ come from a mean zero bivariate normal distribution with unit variance and covariance ρ. The segments are unbalanced with: $\pi_{11} = 0.4$, $\pi_{10} = 0.2$, $\pi_{01} = 0.2$, and $\pi_{00} = 0.2$.

Algorithm	Bias	SD	Tstat	Pval
Oracle	-0.037	0.528	-2.214	0.014
CC	-0.005	0.814	-0.196	0.422
IPW	0.018	0.912	0.626	0.266
Prop	-0.036	0.634	-1.808	0.035
PropOpt	-0.034	0.631	-1.704	0.044
SemiDelta	-0.037	0.637	-1.856	0.032
GLS1	-0.031	0.780	-1.273	0.102
GLS2	-0.004	0.273	-0.486	0.314

Table 6: Results from the unbalanced linear simulation. True values: $\theta = 5, \rho = 0.5$. The test conducted for the T-statistic and P-value is a two sample test to see if the estimator is unbiased. This simulation uses $Y_1 = x + \varepsilon_1$, $Y_2 = \mu + x + \varepsilon_2$ where $X \sim N(0, 1)$ and $(\varepsilon_1, \varepsilon_2)$ come from a mean zero bivariate normal distribution with unit variance and covariance ρ. The segments are unbalanced with: $\pi_{11} = 0.2$, $\pi_{10} = 0.4$, $\pi_{01} = 0.1$, and $\pi_{00} = 0.3$.

Algorithm	Bias	SD	Tstat	Pval
Oracle	-0.002	0.045	-1.669	0.048
CC	0.001	0.083	0.263	0.396
IPW	0.010	0.357	0.872	0.192
WLS	0.000	0.054	-0.065	0.474
Prop	-0.002	0.063	-0.768	0.221
PropOpt	-0.001	0.063	-0.727	0.234
SemiDelta	-0.001	0.064	-0.342	0.366
GLS1	-0.007	0.417	-0.563	0.287

Algorithm	Bias	SD	Tstat	Pval
GLS2	-0.001	0.064	-0.346	0.365

Table 7: Results from the unbalanced non-linear simulation. True values: $\theta = 10, \rho = 0.5$. The test conducted for the T-statistic and P-value is a two sample test to see if the estimator is unbiased. This simulation uses $Y_1 = x + \varepsilon_1$, $Y_2 = \mu + x + \varepsilon_2$ where $X \sim N(0, 1)$ and $(\varepsilon_1, \varepsilon_2)$ come from a mean zero bivariate normal distribution with unit variance and covariance ρ. The segments are unbalanced with: $\pi_{11} = 0.2$, $\pi_{10} = 0.4$, $\pi_{01} = 0.1$, and $\pi_{00} = 0.3$.

Algorithm	Bias	SD	Tstat	Pval
Oracle	-0.037	0.528	-2.214	0.014
CC	0.007	1.196	0.179	0.429
IPW	0.023	1.405	0.520	0.302
Prop	-0.044	0.691	-2.027	0.021
PropOpt	-0.038	0.669	-1.784	0.037
SemiDelta	-0.032	0.674	-1.521	0.064
GLS1	-0.026	1.053	-0.788	0.216
GLS2	-0.003	0.288	-0.301	0.382

Appendix A: Computing the Covariance Matrices

Understanding the covariance matrices V_1 and V_2 require quite a few individual calculations. Here we demonstrate the covariance matrix by solving a general form and demonstrating what this means for a couple examples. Consider the estimators $\hat{\theta}_a$ and $\hat{\theta}_b$ where

$$\hat{\theta}_a = n^{-1} \sum_{i=1}^{n} f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] \text{ and } \hat{\theta}_b = n^{-1} \sum_{i=1}^{n} f_b(\delta_{bi}, \pi_b) E[g \mid G_b(Z_i)]$$

such that the coefficients f_a and f_b satisfy

$$E[f_a(\delta_{ai}, \pi_a) \mid Z] = 0 \text{ and } E[f_b(\delta_{bi}, \pi_b) \mid Z] = 0.$$

Let $f_{ab} = f_a(\delta_a, \pi_a) f_b(\delta_b, \pi_b)$. We consider two cases: Case 1 is when $G_a(Z) \subseteq G_b(Z)$ and Case 2 occurs when neither $G_a(Z) \nsubseteq G_b(Z)$ nor $G_b(Z) \nsubseteq G_a(Z)$. In Case 1, we have

$$
\begin{aligned}
&\mathrm{Cov}(\widehat{\theta}_a, \widehat{\theta}_b) \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1} \mathrm{Cov}(f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)], f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]) \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \big\{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] \\
&\quad - E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)]] E[f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] \big\} \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \big\{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] \\
&\quad - E[E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] \mid Z]] E[E[f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)] \mid Z]] \big\} \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \big\{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] - E[E[g \mid G_a(Z_i)]] E[E[g \mid G_b(Z_j)]] \big\} \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \big\{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] - E[g]^2 \big\} \\
&= n^{-2} \sum_{i=1}^{n} \{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bi}, \pi_b) E[g \mid G_b(Z_i)]] \\
&\quad + \sum_{j=1, j \neq i}^{n} E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] - E[g]^2 \big\} \\
&= n^{-2} \sum_{i=1}^{n} \{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bi}, \pi_b) E[g \mid G_b(Z_i)]] \\
&\quad + \sum_{j=1, j \neq i}^{n} E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)]] E[f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)]] \big\} - E[g]^2 \\
&= n^{-2} \sum_{i=1}^{n} \{ E[f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)] f_b(\delta_{bi}, \pi_b) E[g \mid G_b(Z_i)]] + (n-1) E[g]^2 \big\} - E[g]^2 \\
&= n^{-2} \sum_{i=1}^{n} \big\{ E[f_{ab} E[g \mid G_a(Z_i)] E[g \mid G_b(Z_i)]] + (n-1) E[g]^2 \big\} - E[g]^2 \\
&= n^{-2} \sum_{i=1}^{n} \{ E[E[f_{ab} E[g \mid G_a(Z_i)] E[g \mid G_b(Z_i)] \mid Z_i]] + (n-1) E[g]^2 \big\} - E[g]^2 \\
&= n^{-2} \sum_{i=1}^{n} \{ E[f_{ab} \mid Z_i] E[E[g \mid G_a(Z_i)] E[g \mid G_b(Z_i)]] + (n-1) E[g]^2 \big\} - E[g]^2
\end{aligned}
$$

Since $G_a(Z) \subseteq G_b(Z)$,

$$
\begin{aligned}
&= n^{-2} \sum_{i=1}^{n} \left\{ E[f_{ab} \mid Z_i] E[E[E[g \mid G_a(Z_i)] E[g \mid G_b(Z_i)] \mid G_a(Z_i)]] + (n-1) E[g]^2 \right\} - E[g]^2 \\
&= n^{-2} \sum_{i=1}^{n} \left\{ E[f_{ab} \mid Z_i] E[E[g \mid G_a(Z_i)]^2] + (n-1) E[g]^2 \right\} - E[g]^2 \\
&= n^{-1} \left\{ E[f_{ab} \mid Z] E[E[g \mid G_a(Z)]^2] + (n-1) E[g]^2 \right\} - E[g]^2 \\
&= n^{-1} \left\{ E[f_{ab} \mid Z] E[E[g \mid G_a(Z)]^2] - E[g]^2 \right\}
\end{aligned}
$$

Case 2 is almost identical to Case 1, except that we do not gain anything from conditioning on the coarser variables. For this case, we have,

$$
\begin{aligned}
&\mathrm{Cov}(\hat{\theta}_a, \hat{\theta}_b) \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1} \mathrm{Cov}(f_a(\delta_{ai}, \pi_a) E[g \mid G_a(Z_i)], f_b(\delta_{bj}, \pi_b) E[g \mid G_b(Z_j)])
\end{aligned}
$$

Using the same logic as Case 1,

$$
= n^{-2} \sum_{i=1}^{n} \left\{ E[f_{ab} \mid Z_i] E[E[g \mid G_a(Z_i)] E[g \mid G_b(Z_i)]] + (n-1) E[g]^2 \right\} - E[g]^2
$$

Since $G_a(Z) \not\subseteq G_b(Z)$ and $G_b(Z) \not\subseteq G_a(Z)$,

$$
= n^{-1} \left\{ E[f_{ab} \mid Z] E[E[g \mid G_a(Z)] E[g \mid G_b(Z)]] - E[g]^2 \right\}
$$

As examples, this results in the following:

$$
\begin{aligned}
&\mathrm{Cov}\left(n^{-1} \sum_{i=1}^{n} \left(\frac{\delta_{10i} + \delta_{11i}}{\pi_{10}\pi_{11}} \right) E[g \mid x_i], \sum_{j=1}^{n} \left(\frac{\delta_{10j} + \delta_{11j}}{\pi_{10}\pi_{11}} \right) E[g \mid x_j, y_{1j}], \right) \\
&\quad = n^{-1} \left(\left(\frac{1}{\pi_{10} + \pi_{11}} \right) E[E[g \mid X]^2] - E[g]^2 \right).
\end{aligned}
$$

and

$$\mathrm{Cov}\{n^{-1}\sum_{i=1}^{n}\left(\frac{\delta_{01i}+\delta_{11i}}{\pi_{10}\pi_{11}}\right)E[g\mid x_i,y_{2i}],\sum_{j=1}^{n}\left(\frac{\delta_{10j}+\delta_{11j}}{\pi_{10}\pi_{11}}\right)E[g\mid x_j,y_{1j}],\}$$
$$= n^{-1}\left(\left(\frac{\pi_{11}}{(\pi_{10}+\pi_{11})(\pi_{01}+\pi_{11})}\right)E[E[g\mid X,Y_1]E[g\mid X,Y_2]]-E[g]^2\right).$$

Appendix B: Code for Simulations

```
gls_est1 <- function(df, gfun = "Y2", theta = NA, mean_x = 0, cov_e1e2 = 0) {

  df <- mutate(df, g_i = eval(rlang::parse_expr(gfun)))
  df_11 <- filter(df, delta_1 == 1, delta_2 == 1)
  df_10 <- filter(df, delta_1 == 1, delta_2 == 0)
  df_01 <- filter(df, delta_1 == 0, delta_2 == 1)
  df_00 <- filter(df, delta_1 == 0, delta_2 == 0)

  if (is.na(theta)) {
    theta <- opt_lin_est(df, gfun = "Y2", mean_x = mean_x, cov_y1y2 = cov_e1e2)
  }

  # We use the population functional forms for gamma_hat and v_gamma
  if (gfun == "Y2") {

    Eg <- theta
    Egx <- theta + df$X
    Egxy1 <- theta + df$X + cov_e1e2 * (df$Y1 - df$X)
    Egxy2 <- df$Y2
    EEg2 <- theta^2 + 2
    EEgx2 <- theta^2 + 1
    EEgxy12 <- theta^2 + 1 + cov_e1e2^2
    EEgxy22 <- theta^2 + 2
    EEgxy1Egxy2 <- theta^2 + 1 + cov_e1e2^2

  } else if (gfun == "Y1^2 * Y2") {

    Eg <- 2 * theta
    Egx <- df$X^3 + theta * df$X^2 + theta + 2 * df$X * cov_e1e2 + df$X
    Egxy1 <- df$Y1^2 * (theta + df$X + cov_e1e2 * (df$Y1 - df$X))
```

```
  Egxy2 <- df$Y2 * (df$X^2 + 2 * cov_e1e2 * df$X * (df$Y2 - df$X - theta) +
                    1 + cov_e1e2^2 * (df$Y2 - df$X - theta)^2 - cov_e1e2^2)
  EEg2 <- 12 * (theta^2 + 2 * cov_e1e2^2 + 4 * cov_e1e2 + 4)
  EEgx2 <- 6 * theta^2 + 4 * cov_e1e2^2 + 16 * cov_e1e2 + 22
  EEgxy12 <- 12 * (theta^2 + 3 * cov_e1e2^2 + 4 * cov_e1e2 + 3)
  EEgxy22 <-
    6 * theta^2 + 4 * theta^2 * cov_e1e2^2 + 2 * theta^2 * cov_e1e2^4 + 34 +
    32 * cov_e1e2 + 20 * cov_e1e2^2 + 16 * cov_e1e2^3 + 18 * cov_e1e2^4
  EEgxy1Egxy2 <-
    theta^2 * (6 + 4 * cov_e1e2^2 + 2 * cov_e1e2^4) + 22 + 24 * cov_e1e2 +
    26 * cov_e1e2^2 + 20 * cov_e1e2^3 + 18 * cov_e1e2^4 + 4 * cov_e1e2^5 +
    6 * cov_e1e2^6

} else {
  stop("We have only implemented two g-functions.")
}

g_11 <- mean(Egx)
g_21 <- mean((df$delta_10 + df$delta_11) / (df$prob_10 + df$prob_11) * (Egx))
g_22 <- mean((df$delta_10 + df$delta_11) / (df$prob_10 + df$prob_11) * (Egxy1))
g_31 <- mean((df$delta_01 + df$delta_11) / (df$prob_01 + df$prob_11) * (Egx))
g_33 <- mean((df$delta_01 + df$delta_11) / (df$prob_01 + df$prob_11) * (Egxy2))
g_41 <- mean(df$delta_11 / df$prob_11 * (Egx))
g_42 <- mean(df$delta_11 / df$prob_11 * (Egxy1))
g_43 <- mean(df$delta_11 / df$prob_11 * (Egxy2))
g_44 <- mean(df$delta_11 / df$prob_11 * (df$g_i))

gam_vec <- c(g_11, g_21, g_22, g_31, g_33, g_41, g_42, g_43, g_44)

v_gam <- matrix(rep(-Eg^2, 81), nrow = 9)
v_gam[1, 1] <- (EEgx2) - Eg^2
v_gam[1, 2] <- (EEgx2) - Eg^2
v_gam[1, 3] <- (EEgx2) - Eg^2
v_gam[1, 4] <- (EEgx2) - Eg^2
v_gam[1, 5] <- (EEgx2) - Eg^2
v_gam[1, 6] <- (EEgx2) - Eg^2
v_gam[1, 7] <- (EEgx2) - Eg^2
v_gam[1, 8] <- (EEgx2) - Eg^2
v_gam[1, 9] <- (EEgx2) - Eg^2

v_gam[2, 1] <- v_gam[1, 2]
```

```
v_gam[3, 1] <- v_gam[1, 3]
v_gam[4, 1] <- v_gam[1, 4]
v_gam[5, 1] <- v_gam[1, 5]
v_gam[6, 1] <- v_gam[1, 6]
v_gam[7, 1] <- v_gam[1, 7]
v_gam[8, 1] <- v_gam[1, 8]
v_gam[9, 1] <- v_gam[1, 9]

v_gam[2, 2] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[2, 3] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[3, 2] <- v_gam[2, 3]
v_gam[3, 3] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgxy12) - Eg^2

v_gam[2, 4] <- df$prob_11[1] / ((df$prob_01[1] + df$prob_11[1]) * (df$prob_10[1] + df$pr
v_gam[2, 5] <- df$prob_11[1] / ((df$prob_01[1] + df$prob_11[1]) * (df$prob_10[1] + df$pr
v_gam[2, 6] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[2, 7] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[2, 8] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[2, 9] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2

v_gam[4, 2] <- v_gam[2, 4]
v_gam[5, 2] <- v_gam[2, 5]
v_gam[6, 2] <- v_gam[2, 6]
v_gam[7, 2] <- v_gam[2, 7]
v_gam[8, 2] <- v_gam[2, 8]
v_gam[9, 2] <- v_gam[2, 9]

v_gam[3, 4] <- df$prob_11[1] / ((df$prob_01[1] + df$prob_11[1]) * (df$prob_10[1] + df$pr
v_gam[3, 5] <- df$prob_11[1] / ((df$prob_01[1] + df$prob_11[1]) * (df$prob_10[1] + df$pr
v_gam[3, 6] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[3, 7] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgxy12) - Eg^2
v_gam[3, 8] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgxy1Egxy2) - Eg^2
v_gam[3, 9] <- 1 / (df$prob_10[1] + df$prob_11[1]) * (EEgxy12) - Eg^2

v_gam[4, 3] <- v_gam[3, 4]
v_gam[5, 3] <- v_gam[3, 5]
v_gam[6, 3] <- v_gam[3, 6]
v_gam[7, 3] <- v_gam[3, 7]
v_gam[8, 3] <- v_gam[3, 8]
v_gam[9, 3] <- v_gam[3, 9]
```

```r
v_gam[4, 4] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[4, 5] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[5, 4] <- v_gam[4, 5]
v_gam[5, 5] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgxy22) - Eg^2

v_gam[4, 6] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[4, 7] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[4, 8] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[4, 9] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2

v_gam[6, 4] <- v_gam[4, 6]
v_gam[7, 4] <- v_gam[4, 7]
v_gam[8, 4] <- v_gam[4, 8]
v_gam[9, 4] <- v_gam[4, 9]

v_gam[5, 6] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgx2) - Eg^2
v_gam[5, 7] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgxy1Egxy2) - Eg^2
v_gam[5, 8] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgxy22) - Eg^2
v_gam[5, 9] <- 1 / (df$prob_01[1] + df$prob_11[1]) * (EEgxy22) - Eg^2

v_gam[6, 6] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
v_gam[6, 7] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
v_gam[7, 6] <- v_gam[6, 7]
v_gam[6, 8] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
v_gam[8, 6] <- v_gam[6, 8]
v_gam[6, 9] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
v_gam[9, 6] <- v_gam[6, 9]
v_gam[7, 7] <- 1 / df$prob_11[1] * (EEgxy12) - Eg^2
v_gam[7, 8] <- 1 / df$prob_11[1] * (EEgxy1Egxy2) - Eg^2
v_gam[8, 7] <- v_gam[7, 8]
v_gam[7, 9] <- 1 / df$prob_11[1] * (EEgxy12) - Eg^2
v_gam[9, 7] <- v_gam[7, 9]
v_gam[8, 8] <- 1 / df$prob_11[1] * (EEgxy22) - Eg^2
v_gam[8, 9] <- 1 / df$prob_11[1] * (EEgxy22) - Eg^2
v_gam[9, 8] <- v_gam[8, 9]
v_gam[9, 9] <- 1 / df$prob_11[1] * (EEg2) - Eg^2

v_gam <- v_gam / nrow(df)

z_mat <- matrix(c(1, 0, 0, 0,
                  1, 0, 0, 0,
```

```r
                    0, 1, 0, 0,
                    1, 0, 0, 0,
                    0, 0, 1, 0,
                    1, 0, 0, 0,
                    0, 1, 0, 0,
                    0, 0, 1, 0,
                    0, 0, 0, 1), ncol = 4, byrow = TRUE)

  one_mat <- t(c(0, 0, 0, 1))

  dmat <- (t(z_mat) %*% hw_inv(v_gam) %*% z_mat)
  nmat <- t(z_mat) %*% hw_inv(v_gam) %*% gam_vec
  ret <- as.numeric(one_mat %*% hw_inv(dmat) %*% nmat)
  return(ret)

}

gls_est2 <- function(df, gfun = "Y2", theta = NA, mean_x = 0, cov_e1e2 = 0) {

  df <- mutate(df, g_i = eval(rlang::parse_expr(gfun)))
  df_11 <- filter(df, delta_1 == 1, delta_2 == 1)
  df_10 <- filter(df, delta_1 == 1, delta_2 == 0)
  df_01 <- filter(df, delta_1 == 0, delta_2 == 1)
  df_00 <- filter(df, delta_1 == 0, delta_2 == 0)

  if (is.na(theta)) {
    theta <- opt_lin_est(df, gfun = "Y2", mean_x = mean_x, cov_y1y2 = cov_e1e2)
  }

  # We use the population functional forms for gamma_hat and v_gamma
  if (gfun == "Y2") {

    Eg <- theta
    Egx <- theta + df$X
    Egxy1 <- theta + df$X + cov_e1e2 * (df$Y1 - df$X)
    Egxy2 <- df$Y2
    EEg2 <- theta^2 + 2
    EEgx2 <- theta^2 + 1
    EEgxy12 <- theta^2 + 1 + cov_e1e2^2
    EEgxy22 <- theta^2 + 2
    EEgxy1Egxy2 <- theta^2 + 1 + cov_e1e2^2
```

```
} else if (gfun == "Y1^2 * Y2") {

  Eg <- 2 * theta
  Egx <- df$X^3 + theta * df$X^2 + theta + 2 * df$X * cov_e1e2 + df$X
  Egxy1 <- df$Y1^2 * (theta + df$X + cov_e1e2 * (df$Y1 - df$X))
  Egxy2 <- df$Y2 * (df$X^2 + 2 * cov_e1e2 * df$X * (df$Y2 - df$X - theta) +
                    1 + cov_e1e2^2 * (df$Y2 - df$X - theta)^2 - cov_e1e2^2)
  EEg2 <- 12 * (theta^2 + 2 * cov_e1e2^2 + 4 * cov_e1e2 + 4)
  EEgx2 <- 6 * theta^2 + 4 * cov_e1e2^2 + 16 * cov_e1e2 + 22
  EEgxy12 <- 12 * (theta^2 + 3 * cov_e1e2^2 + 4 * cov_e1e2 + 3)
  EEgxy22 <-
    6 * theta^2 + 4 * theta^2 * cov_e1e2^2 + 2 * theta^2 * cov_e1e2^4 + 34 +
    32 * cov_e1e2 + 20 * cov_e1e2^2 + 16 * cov_e1e2^3 + 18 * cov_e1e2^4
  EEgxy1Egxy2 <-
    theta^2 * (6 + 4 * cov_e1e2^2 + 2 * cov_e1e2^4) + 22 + 24 * cov_e1e2 +
    26 * cov_e1e2^2 + 20 * cov_e1e2^3 + 18 * cov_e1e2^4 + 4 * cov_e1e2^5 +
    6 * cov_e1e2^6

} else {
  stop("We have only implemented two g-functions.")
}

g_11 <- mean(df$delta_00 / df$prob_00 * (Egx))
g_21 <- mean(df$delta_10 / df$prob_10 * (Egx))
g_22 <- mean(df$delta_10 / df$prob_10 * (Egxy1))
g_31 <- mean(df$delta_01 / df$prob_01 * (Egx))
g_33 <- mean(df$delta_01 / df$prob_01 * (Egxy2))
g_41 <- mean(df$delta_11 / df$prob_11 * (Egx))
g_42 <- mean(df$delta_11 / df$prob_11 * (Egxy1))
g_43 <- mean(df$delta_11 / df$prob_11 * (Egxy2))
g_44 <- mean(df$delta_11 / df$prob_11 * (df$g_i))

gam_vec <- c(g_11, g_21, g_22, g_31, g_33, g_41, g_42, g_43, g_44)

v_gam <- matrix(rep(-Eg^2, 81), nrow = 9)
v_gam[1, 1] <- 1 / df$prob_00[1] * (EEgx2) - Eg^2

v_gam[2, 2] <- 1 / df$prob_10[1] * (EEgx2) - Eg^2
v_gam[2, 3] <- 1 / df$prob_10[1] * (EEgx2) - Eg^2
v_gam[3, 2] <- v_gam[2, 3]
v_gam[3, 3] <- 1 / df$prob_10[1] * (EEgxy12) - Eg^2
```

```r
  v_gam[4, 4] <- 1 / df$prob_01[1] * (EEgx2) - Eg^2
  v_gam[4, 5] <- 1 / df$prob_01[1] * (EEgx2) - Eg^2
  v_gam[5, 4] <- v_gam[4, 5]
  v_gam[5, 5] <- 1 / df$prob_01[1] * (EEgxy22) - Eg^2

  v_gam[6, 6] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
  v_gam[6, 7] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
  v_gam[7, 6] <- v_gam[6, 7]
  v_gam[6, 8] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
  v_gam[8, 6] <- v_gam[6, 8]
  v_gam[6, 9] <- 1 / df$prob_11[1] * (EEgx2) - Eg^2
  v_gam[9, 6] <- v_gam[6, 9]
  v_gam[7, 7] <- 1 / df$prob_11[1] * (EEgxy12) - Eg^2
  v_gam[7, 8] <- 1 / df$prob_11[1] * (EEgxy1Egxy2) - Eg^2
  v_gam[8, 7] <- v_gam[7, 8]
  v_gam[7, 9] <- 1 / df$prob_11[1] * (EEgxy12) - Eg^2
  v_gam[9, 7] <- v_gam[7, 9]
  v_gam[8, 8] <- 1 / df$prob_11[1] * (EEgxy22) - Eg^2
  v_gam[8, 9] <- 1 / df$prob_11[1] * (EEgxy22) - Eg^2
  v_gam[9, 8] <- v_gam[8, 9]
  v_gam[9, 9] <- 1 / df$prob_11[1] * (EEg2) - Eg^2

  v_gam <- v_gam / nrow(df)

  one_mat <- matrix(rep(1, length(gam_vec)), ncol = 1)

  dmat <- (t(one_mat) %*% hw_inv(v_gam) %*% one_mat)
  nmat <- t(one_mat) %*% hw_inv(v_gam) %*% gam_vec
  return(as.numeric(nmat / dmat))

}
```