# Supplementary Materials for "Reduction of bias due to misclassified exposures using instrumental variables"

Christopher Manuel, Samiran Sinha[†] and Suojin Wang

Department of Statistics, Texas A&M University, College Station, Texas 77843-3143
[†] sinha@stat.tamu.edu

Table 1: Results of the simulation study **using MCMC method** for three categories, two strongly associated instruments (scenario Ia). MT: method, B: median bias ×100, SD: median of posterior standard deviation ×100,CI: 95% credible interval, CI-W: Median width of 95% credible interval, RMSE: Root mean squared error×100, $\text{pr}(Y = 1|X, Z) = H\{\beta_0 + \beta_{x,2}(X = 2) + \beta_{x,3}I(X = 3) + \beta_z Z\}$, $\beta_0 = -2, \beta_{x,2} = 1, \beta_{x,3} = 0.7, \beta_z = 0.5$, S1: $\sigma = 2$, S2: $\sigma = 5$.200 replications were used.

| MT | | M1 | | | | M2 | | | | M3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ |
| $n = 5,000$ | | | | | | | | | | | | | |
| | B | 0.21 | 0.71 | −0.03 | 0.01 | 46.94 | −51.88 | −46.09 | 0.54 | −1.30 | 3.63 | 2.61 | −0.65 |
| | SD | 12.59 | 13.66 | 13.46 | 6.09 | 7.36 | 9.14 | 9.28 | 6.09 | 22.97 | 24.27 | 24.52 | 6.05 |
| S1 | CI | (−2.25, −1.75) | (0.69, 1.26) | (0.44, 0.95) | (0.37, 0.60) | (−1.66, −1.37) | (0.28, 0.64) | (0.06, 0.39) | (0.38, 0.61) | (−2.55, −1.66) | (0.67, 1.59) | (0.34, 1.24) | (0.37, 0.60) |
| | CI-W | 0.49 | 0.54 | 0.53 | 0.24 | 0.29 | 0.36 | 0.36 | 0.24 | 0.90 | 0.95 | 0.96 | 0.24 |
| | RMSE | 12.66 | 14.17 | 13.70 | 5.85 | 48.05 | 53.33 | 47.59 | 5.81 | 23.14 | 25.47 | 24.72 | 5.99 |
| | B | −0.41 | 0.40 | 0.42 | 0.06 | 46.88 | −51.89 | −45.95 | 0.52 | −2.54 | 4.89 | 5.78 | −0.52 |
| | SD | 12.78 | 13.86 | 13.65 | 6.10 | 7.37 | 9.15 | 9.28 | 6.09 | 26.70 | 27.81 | 27.81 | 6.02 |
| S2 | CI | (−2.27, −1.75) | (0.70, 1.28) | (0.44, 0.96) | (0.37, 0.61) | (−1.66, −1.37) | (0.28, 0.64) | (0.06, 0.40) | (0.38, 0.61) | (−2.78, −1.64) | (0.62, 1.87) | (0.35, 1.53) | (0.37, 0.60) |
| | CI-W | 0.50 | 0.54 | 0.53 | 0.24 | 0.29 | 0.36 | 0.36 | 0.24 | 1.05 | 1.09 | 1.09 | 0.24 |
| | RMSE | 12.77 | 14.27 | 13.79 | 5.85 | 48.06 | 53.34 | 47.59 | 5.81 | 33.19 | 34.81 | 34.15 | 5.93 |
| $n = 10,000$ | | | | | | | | | | | | | |
| | B | −1.30 | 0.98 | 0.95 | −0.49 | 47.39 | −52.81 | −46.14 | −0.11 | 1.69 | 0.15 | −1.01 | −0.63 |
| | SD | 9.02 | 9.76 | 9.63 | 4.30 | 5.19 | 6.44 | 6.51 | 4.29 | 16.32 | 17.20 | 17.29 | 4.26 |
| S1 | CI | (−2.19, −1.85) | (0.85, 1.20) | (0.54, 0.90) | (0.43, 0.57) | (−1.64, −1.44) | (0.37, 0.61) | (0.13, 0.37) | (0.43, 0.58) | (−2.36, −1.72) | (0.71, 1.38) | (0.41, 1.10) | (0.42, 0.57) |
| | CI-W | 0.35 | 0.38 | 0.38 | 0.17 | 0.20 | 0.25 | 0.26 | 0.17 | 0.64 | 0.67 | 0.68 | 0.17 |
| | RMSE | 8.95 | 9.41 | 9.82 | 4.00 | 47.40 | 52.77 | 46.24 | 3.98 | 16.23 | 17.49 | 17.46 | 4.10 |
| | B | −1.29 | 1.50 | 1.25 | −0.49 | 47.24 | −52.79 | −46.07 | −0.14 | −0.63 | 0.68 | 1.25 | −0.44 |
| | SD | 9.09 | 9.87 | 9.71 | 4.29 | 5.19 | 6.45 | 6.53 | 4.29 | 18.32 | 19.26 | 18.98 | 4.26 |
| S2 | CI | (−2.19, −1.85) | (0.85, 1.21) | (0.55, 0.91) | (0.43, 0.57) | (−1.64, −1.44) | (0.37, 0.61) | (0.13, 0.37) | (0.43, 0.58) | (−2.41, −1.71) | (0.68, 1.45) | (0.40, 1.15) | (0.42, 0.57) |
| | CI-W | 0.36 | 0.39 | 0.38 | 0.17 | 0.20 | 0.25 | 0.26 | 0.17 | 0.72 | 0.76 | 0.74 | 0.17 |
| | RMSE | 9.00 | 9.51 | 9.88 | 4.00 | 47.42 | 52.79 | 46.27 | 3.98 | 18.92 | 20.18 | 19.78 | 4.06 |
| $n = 20,000$ | | | | | | | | | | | | | |
| | B | −0.33 | −0.11 | −0.31 | 0.47 | 47.65 | −52.37 | −46.52 | 1.13 | 1.91 | −1.21 | −1.63 | 0.53 |
| | SD | 6.19 | 6.73 | 6.63 | 3.05 | 3.68 | 4.56 | 4.62 | 3.04 | 11.78 | 12.51 | 12.35 | 2.99 |
| S1 | CI | (−2.13, −1.89) | (0.88, 1.16) | (0.58, 0.82) | (0.45, 0.56) | (−1.60, −1.45) | (0.38, 0.56) | (0.15, 0.31) | (0.45, 0.57) | (−2.23, −1.77) | (0.79, 1.24) | (0.45, 0.94) | (0.44, 0.56) |
| | CI-W | 0.24 | 0.26 | 0.26 | 0.12 | 0.14 | 0.18 | 0.18 | 0.12 | 0.46 | 0.49 | 0.48 | 0.12 |
| | RMSE | 6.52 | 7.30 | 6.64 | 2.94 | 47.62 | 52.52 | 46.75 | 3.06 | 11.86 | 12.14 | 11.83 | 2.98 |
| | B | −0.52 | 0.16 | −0.31 | 0.44 | 47.52 | −52.37 | −46.54 | 1.14 | 1.27 | −1.03 | −0.81 | 0.61 |
| | SD | 6.23 | 6.77 | 6.66 | 3.06 | 3.70 | 4.57 | 4.64 | 3.04 | 12.59 | 13.20 | 13.02 | 2.9 |
| S2 | CI | (−2.13, −1.89) | (0.88, 1.16) | (0.59, 0.82) | (0.45, 0.56) | (−1.60, −1.44) | (0.38, 0.56) | (0.15, 0.31) | (0.45, 0.57) | (−2.26, −1.73) | (0.74, 1.29) | (0.43, 0.97) | (0.45, 0.57) |
| | CI-W | 0.24 | 0.27 | 0.26 | 0.12 | 0.15 | 0.18 | 0.18 | 0.12 | 0.49 | 0.52 | 0.51 | 0.12 |
| | RMSE | 6.50 | 7.30 | 6.60 | 2.94 | 47.61 | 52.51 | 46.75 | 3.07 | 13.12 | 13.53 | 12.97 | 2.98 |

Table 2: Results of the simulation study **using MCMC method** for three categories, two moderately associated instruments (scenario Ib). MT: method, B: median bias $\times 100$, SD: median of posterior standard deviation $\times 100$,CI: 95% credible interval, CI-W: Median width of 95% credible interval, RMSE: Root mean squared error$\times 100$, $\mathrm{pr}(Y = 1|X, Z) = H\{\beta_0 + \beta_{x,2}(X = 2) + \beta_{x,3}I(X = 3) + \beta_z Z\}$, $\beta_0 = -2, \beta_{x,2} = 1, \beta_{x,3} = 0.7, \beta_z = 0.5$, $S1 : \sigma = 2$, $S2 : \sigma = 5$. 200 replications were used.

| MT | | M1 | | | | M2 | | | | M3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_z$ |
| $n = 5,000$ | | | | | | | | | | | | | |
| | B | −0.49 | −0.02 | −0.07 | 0.72 | 40.86 | −45.69 | −40.38 | 0.88 | −19.58 | 21.06 | 23.87 | −0.07 |
| | SD | 11.02 | 12.23 | 12.19 | 6.14 | 7.06 | 8.88 | 9.19 | 6.14 | 33.74 | 34.38 | 34.87 | 6.24 |
| S1 | CI | (−2.22, −1.84) | (0.82, 1.21) | (0.52, 0.95) | (0.39, 0.63) | (−1.74, −1.44) | (0.34, 0.74) | (0.11, 0.50) | (0.39, 0.63) | (−2.89, −1.67) | (0.66, 2.03) | (0.38, 1.65) | (0.37, 0.61) |
| | CI-W | 0.43 | 0.48 | 0.48 | 0.24 | 0.28 | 0.35 | 0.36 | 0.24 | 1.32 | 1.35 | 1.37 | 0.24 |
| | RMSE | 9.9 | 10.75 | 11.31 | 5.8 | 41.02 | 46.68 | 40.71 | 5.92 | 39.19 | 41.48 | 42.89 | 6.04 |
| | B | −1.23 | 0.14 | 0.13 | 0.72 | 40.77 | −45.82 | −40.35 | 0.81 | −18.35 | 23.25 | 25.92 | −0.02 |
| | SD | 11.01 | 12.26 | 12.18 | 6.16 | 7.06 | 8.9 | 9.2 | 6.13 | 40 | 40 | 41.05 | 6.2 |
| S2 | CI | (−2.22, −1.85) | (0.82, 1.22) | (0.52, 0.95) | (0.39, 0.63) | (−1.73, −1.44) | (0.34, 0.74) | (0.12, 0.50) | (0.39, 0.63) | (−3.51, −1.58) | (0.58, 2.51) | (0.29, 2.23) | (0.37, 0.61) |
| | CI-W | 0.43 | 0.48 | 0.48 | 0.24 | 0.28 | 0.35 | 0.36 | 0.24 | 1.57 | 1.57 | 1.61 | 0.24 |
| | RMSE | 10.04 | 10.85 | 11.46 | 5.8 | 41.03 | 46.69 | 40.71 | 5.92 | 59.74 | 59.77 | 61.42 | 6.01 |
| $n = 10,000$ | | | | | | | | | | | | | |
| | B | −1.28 | 0.86 | 1.17 | 0.22 | 41.11 | −46.1 | −40.13 | 0.63 | −14.87 | 19.71 | 17.32 | −0.07 |
| | SD | 7.65 | 8.51 | 8.45 | 4.33 | 4.94 | 6.24 | 6.44 | 4.32 | 25.59 | 25.86 | 26.19 | 4.38 |
| S1 | CI | (−2.17, −1.87) | (0.83, 1.18) | (0.55, 0.87) | (0.43, 0.59) | (−1.69, −1.49) | (0.40, 0.68) | (0.16, 0.43) | (0.43, 0.59) | (−2.65, −1.81) | (0.78, 1.68) | (0.49, 1.40) | (0.42, 0.60) |
| | CI-W | 0.3 | 0.33 | 0.33 | 0.17 | 0.19 | 0.24 | 0.25 | 0.17 | 1.00 | 1.01 | 1.03 | 0.17 |
| | RMSE | 7.76 | 8.7 | 8.45 | 4.3 | 41.24 | 46.61 | 40.47 | 4.3 | 29.61 | 30.81 | 31.32 | 4.39 |
| | B | −1.36 | 0.72 | 1.52 | 0.21 | 41.18 | −45.9 | −40.07 | 0.65 | −12.96 | 16.61 | 14.18 | 0.02 |
| | SD | 7.65 | 8.5 | 8.45 | 4.33 | 4.95 | 6.24 | 6.43 | 4.32 | 29.09 | 29.42 | 29.48 | 4.35 |
| S2 | CI | (−2.17, −1.87) | (0.83, 1.18) | (0.55, 0.88) | (0.43, 0.59) | (−1.69, −1.49) | (0.40, 0.68) | (0.17, 0.43) | (0.43, 0.59) | (−2.91, −1.69) | (0.68, 1.86) | (0.37, 1.64) | (0.42, 0.60) |
| | CI-W | 0.3 | 0.33 | 0.33 | 0.17 | 0.19 | 0.24 | 0.25 | 0.17 | 1.14 | 1.15 | 1.16 | 0.17 |
| | RMSE | 7.83 | 8.75 | 8.53 | 4.31 | 41.23 | 46.59 | 40.45 | 4.3 | 37.36 | 37.02 | 38.08 | 4.34 |
| $n = 20,000$ | | | | | | | | | | | | | |
| | B | −0.68 | 0.53 | 0.53 | −0.22 | 40.53 | −45.49 | −39.96 | 0.07 | −4.84 | 6.73 | 7.11 | −0.23 |
| | SD | 5.37 | 5.99 | 5.92 | 3.08 | 3.55 | 4.45 | 4.59 | 3.06 | 16.8 | 17.1 | 17.37 | 3.05 |
| S1 | CI | (−2.12, −1.90) | (0.89, 1.12) | (0.59, 0.81) | (0.44, 0.56) | (−1.66, −1.52) | (0.47, 0.64) | (0.22, 0.39) | (0.45, 0.57) | (−2.41, −1.81) | (0.83, 1.43) | (0.51, 1.13) | (0.44, 0.56) |
| | CI-W | 0.21 | 0.23 | 0.23 | 0.12 | 0.14 | 0.17 | 0.18 | 0.12 | 0.66 | 0.67 | 0.68 | 0.12 |
| | RMSE | 5.18 | 5.82 | 5.7 | 3.14 | 40.76 | 45.5 | 39.99 | 3.15 | 16.66 | 17.76 | 17.41 | 3.16 |
| | B | −0.55 | 0.65 | 0.61 | −0.22 | 40.7 | −45.41 | −39.89 | 0.09 | 1.37 | −0.94 | 0.37 | −0.25 |
| | SD | 5.37 | 6 | 5.93 | 3.07 | 3.56 | 4.46 | 4.6 | 3.06 | 18.34 | 18.81 | 18.8 | 3.03 |
| S2 | CI | (−2.11, −1.90) | (0.89, 1.11) | (0.58, 0.81) | (0.44, 0.57) | (−1.66, −1.52) | (0.47, 0.64) | (0.22, 0.39) | (0.45, 0.57) | (−2.43, −1.71) | (0.70, 1.45) | (0.41, 1.16) | (0.44, 0.56) |
| | CI-W | 0.21 | 0.24 | 0.23 | 0.12 | 0.14 | 0.17 | 0.18 | 0.12 | 0.72 | 0.74 | 0.74 | 0.12 |
| | RMSE | 5.15 | 5.78 | 5.67 | 3.14 | 40.76 | 45.49 | 39.98 | 3.15 | 20.26 | 20.34 | 20.29 | 3.16 |

Table 3: Results of the simulation study mimicking the real dataset (scenario II) **using MCMC method**. MT: method, B: median bias $\times 100$, SD: median of posterior standard deviation $\times 100$, CI: 95% credible interval, CI-W: Median width of 95% credible interval, RMSE: Root mean squared error$\times 100$, $\text{pr}(Y=1|X,Z) = H\{\beta_0 + \beta_{x,2} I(X=2) + \beta_{x,3} I(X=3) + \beta_{z_1} Z_1 + \beta_{z_2} Z_2\}$, $\beta_0 = 1.13, \beta_{x,2} = 0, \beta_{x,3} = 1.98, \beta_{z_1} = -0.5, \beta_{z_2} = -2$, $n = 10,000$. 200 replications were used.

| MT | | $\beta_0$ | $\beta_{x,2}$ | $\beta_{x,3}$ | $\beta_{z_1}$ | $\beta_{z_2}$ |
|---|---|---|---|---|---|---|
| | | | | S1 | | |
| M1 | B | 0.11 | 0.2 | $-0.23$ | $-0.46$ | 8.12 |
| | SD | 5.09 | 5.91 | 7.37 | 5.6 | 6.54 |
| | CI | $(1.04, 1.24)$ | $(-0.10, 0.10)$ | $(1.83, 2.12)$ | $(-0.61, -0.40)$ | $(-2.13, -1.88)$ |
| | CI-W | 0.2 | 0.23 | 0.29 | 0.22 | 0.26 |
| | RMSE | 4.9 | 5.56 | 6.87 | 5.81 | 9.85 |
| M2 | B | 9.08 | 0.21 | $-74.84$ | 1.39 | 69.21 |
| | SD | 4.57 | 5.51 | 6.59 | 5.35 | 5.27 |
| | CI | $(1.14, 1.30)$ | $(-0.10, 0.10)$ | $(1.09, 1.36)$ | $(-0.59, -0.39)$ | $(-1.48, -1.28)$ |
| | CI-W | 0.18 | 0.22 | 0.26 | 0.21 | 0.21 |
| | RMSE | 9.84 | 5.2 | 75.47 | 5.69 | 69.31 |
| M3 | B | $-1.08$ | 0.83 | $-0.16$ | 1.68 | 2.83 |
| | SD | 6.88 | 9.77 | 12.53 | 6.72 | 11.02 |
| | CI | $(0.98, 1.25)$ | $(-0.16, 0.22)$ | $(1.78, 2.17)$ | $(-0.64, -0.37)$ | $(-2.26, -1.87)$ |
| | CI-W | 0.27 | 0.38 | 0.49 | 0.26 | 0.43 |
| | RMSE | 7.06 | 9.73 | 10.72 | 6.8 | 9.96 |
| | | | | S2 | | |
| M1 | B | 0.2 | 0.18 | $-0.06$ | $-0.64$ | 7.76 |
| | SD | 5.15 | 6 | 7.4 | 5.6 | 6.54 |
| | CI | $(1.04, 1.23)$ | $(-0.10, 0.09)$ | $(1.83, 2.12)$ | $(-0.61, -0.41)$ | $(-2.13, -1.88)$ |
| | CI-W | 0.2 | 0.24 | 0.29 | 0.22 | 0.26 |
| | RMSE | 4.91 | 5.53 | 6.9 | 5.83 | 9.61 |
| M2 | B | 9.18 | 0.11 | $-74.49$ | 1.24 | 68.86 |
| | SD | 4.71 | 5.56 | 6.43 | 5.24 | 5.39 |
| | CI | $(1.14, 1.30)$ | $(-0.10, 0.10)$ | $(1.09, 1.36)$ | $(-0.59, -0.39)$ | $(-1.48, -1.28)$ |
| | CI-W | 0.18 | 0.22 | 0.25 | 0.21 | 0.21 |
| | RMSE | 9.94 | 5.19 | 75.3 | 5.71 | 69.06 |
| M3 | B | $-0.84$ | 1.37 | $-2.45$ | 1.36 | 4.6 |
| | SD | 6.85 | 9.68 | 12.61 | 6.72 | 11.06 |
| | CI | $(0.98, 1.25)$ | $(-0.15, 0.23)$ | $(1.75, 2.17)$ | $(-0.65, -0.37)$ | $(-2.23, -1.85)$ |
| | CI-W | 0.27 | 0.38 | 0.49 | 0.26 | 0.43 |
| | RMSE | 7.01 | 10.02 | 11.45 | 6.93 | 10.95 |

## How to perform the ADVI analysis using the R package `rstan`

Step 0. Here we present the code for a simulation study.
Step 1. First, data, parameter and likelihood must be written and saved in a file with extension .stan.
Step 2. The above .stan function (file) will be used by the R functions saved in .r file. Simulation results can be obtained when this .r file is run (r code is run).
Step 3. Now, we show the content of the .stan file.

```
/start of the rstan file

data {
  //these were the data that was passed from R
  int<lower=1> N;           // Sample size
  vector[N] ms1;
  vector[N] ms2;
  vector[N] w1;
  vector[N] w2;
  vector[N] w3;
  vector[N] z;
  vector[N] y;              // Target variable
  //estimates of parameters for use as the mean in priors for gamma and beta
  //we use real since these are
  real wm2int;
  real wm2slop1;
  real wm2z;
  real wm2slop2;
  real wm3int;
  real wm3slop1;
  real wm3z;
  real wm3slop2;


  real ywint;
  real ywbeta1;
  real ywbeta2;
  real ywbeta3;
}

parameters {
  // Here we declare which variables will have priors
  // Coefficient vector
  real beta1;
  real beta2;
```

```
  real beta3;
  real beta4;

  // Declare gammas parameter
  real gamma21;
  real gamma22;
  real gamma23;
  real gamma24;

  real gamma31;
  real gamma32;
  real gamma33;
  real gamma34;

  // Declare parameters for transformations of gamma and alpha parameters
  real eta1;
  real eta2;
  real eta3;
  real eta4;
  real eta5;
  real eta6;
}


model {
  // Here we define the model block
  //First we define misclassification probabilities in terms of the eta variables
  real alpha21=0.5/(1+exp(-eta1)+exp(eta2-eta1));
  real alpha31=0.5/(1+exp(-eta2)+exp(eta1-eta2));

  real alpha11=1-alpha21-alpha31;

  real alpha12=0.5/(1+exp(-eta3)+exp(eta4-eta3));
  real alpha32=0.5/(1+exp(-eta4)+exp(eta3-eta4));

  real alpha22=1-alpha12-alpha32;

  real alpha13=0.5/(1+exp(-eta5)+exp(eta6-eta5));
  real alpha23=0.5/(1+exp(-eta6)+exp(eta5-eta6));

  real alpha33=1-alpha13-alpha23;
```

```
 //initialize probabilities for the true x
 real lpm1;
 real lpm2;
 real lpm3;
 real pm1;
 real pm2;
 real pm3;

 //initialize probabilities for y
 real py11;
 real py12;
 real py13;
 real py01;
 real py02;
 real py03;

 //initialize probabilities for y conditional on misclassified w
 real py1w1;
 real py1w2;
 real py1w3;

 real py0w1;
 real py0w2;
 real py0w3;

 real y1w;
 real y0w;

// generate priors for the random variables
//gamma
 gamma21 ~ normal(wm2int,2);
 gamma22 ~ normal(wm2slop1,2);
 gamma23 ~ normal(wm2z,2);
 gamma24 ~ normal(wm2slop2,2);

 gamma31 ~ normal(wm3int,2);
 gamma32 ~ normal(wm3slop1,2);
 gamma33 ~ normal(wm3z,2);
 gamma34 ~ normal(wm3slop2,2);

 eta1 ~ normal(0,2);
```

```
eta2 ~ normal(0,2);
eta3 ~ normal(0,2);
eta4 ~ normal(0,2);
eta5 ~ normal(0,2);
eta6 ~ normal(0,2);

beta1 ~ normal(ywint,2);
beta2 ~ normal(ywbeta1,2);
beta3 ~ normal(ywbeta2,2);
beta4 ~ normal(ywbeta3,2);

 //likelihood
 for (i in 1:num_elements(ms1)){

  //generate probabilities for m=1 and m=0, we include z in the model for m but
  //not in data generation process
lpm2= gamma21+ms1[i]*gamma22+z[i]*gamma23+ms2[i]*gamma24;
lpm3= gamma31+ms1[i]*gamma32+z[i]*gamma33+ms2[i]*gamma34;

pm2 = 1/(1+exp(-lpm2)+exp(lpm3-lpm2));
pm3 = 1/(1+exp(lpm2-lpm3)+exp(-lpm3));
pm1 = 1-pm2-pm3;

//generate probabilities for y=1 and y=0 conditional on m
py11 = inv_logit(beta1+beta4*z[i]);
py12 = inv_logit(beta1+beta2+beta4*z[i]);
py13 = inv_logit(beta1+beta3+beta4*z[i]);
py01=1-py11;
py02=1-py12;
py03=1-py13;

py1w1=w1[i]*log(py11*alpha11*pm1+py12*alpha12*pm2+py13*alpha13*pm3);
py1w2=w2[i]*log(py11*alpha21*pm1+py12*alpha22*pm2+py13*alpha23*pm3);
py1w3=w3[i]*log(py11*alpha31*pm1+py12*alpha32*pm2+py13*alpha33*pm3);

//y1w=y[i]*(w1[i]*log(py11)+w2[i]*log(py12)+w3[i]*log(py13));
y1w=y[i]*(py1w1+py1w2+py1w3);

py0w1=w1[i]*log(py01*alpha11*pm1+py02*alpha12*pm2+py03*alpha13*pm3);
py0w2=w2[i]*log(py01*alpha21*pm1+py02*alpha22*pm2+py03*alpha23*pm3);
py0w3=w3[i]*log(py01*alpha31*pm1+py02*alpha32*pm2+py03*alpha33*pm3);
```

```
  y0w=(1-y[i])*(py0w1+py0w2+py0w3);
  //y0w=(1-y[i])*(w1[i]*log(py01)+w2[i]*log(py02)+w3[i]*log(py03));

  //increment log likelihood for each observation
  target += y1w+y0w;
}
  }
generated quantities {  // Generated quantities block.
}
```

Step 4. Now, we show the content of the R file.

```
#Set up
#Libraries
#rstan is used for implementing the ADVI approach
#nnet is used for modeling the multinomial probabilities for X
library(rstan)
library(nnet)

store=store1=store2=storesd=NULL;

current_directory=getwd()
#specify models for use in simulation
#run outside of loop to speed up
#you must set  the location of the stan file
# Here we assume that the stan code is saved in a file
# named 3by3categoricaltest.stan
logreg_our_model=stan_model(file="3by3categoricaltest.stan")


##Data generation############
#Simulation parameters
#n represents the sample size for the simulation
n=5000

##The function myfn1 is used to run the simulation
#nt represents the iteration
myfn1=function(nt){
  print(nt)
  set.seed(nt)
```

```
#generate prognostic factor z
z=runif(n,-1,1)
#generate instrumental variables xstar (xs)
#ms1 is a bernoulli random variable where n is the sample size, and p = P(ms1=1)
p=.55
xs1=rbinom(n,1,p)
#ms2 is a continuous random variable with sample size n and bounds from (-1,1)
xs2=runif(n,-1,1)

##################
#Generating the true X
##################
#gamma parameters used to model of X
g20=0.25
g21=2
g22=0.3
g23=1
g30=0.5
g31=2.5
g32=-0.3
g33=-1
#
#g2 is the function used in estimating Pr(X=2|XS1,XS2)
# while g3 is the function used to estimate Pr(X=3|XS1,XS2)
g2=g20+g21*xs1+g22*z+g23*xs2
g3=g30+g31*xs1+g32*z+g33*xs2

#Next we set probabilities for different categories of m
#px2 represents the conditional probability Pr(X=2|XS1,XS2)
#px3 represents the conditional probability Pr(X=3|XS1,XS2)
#px1 = P(X=1|XS1,XS2)
px2=1/(1+exp(-g2)+exp(g3-g2))
px3=1/(1+exp(-g3)+exp(g2-g3))
px1=1-px2-px3

#next we combine the multinomial probabilities of Pr(X|XS1,XS2) then use that to
#generate X
px=cbind(px1,px2,px3)
x=matrix(,nrow=3,ncol=n,byrow=TRUE)
for (i in 1:n) {
x[,i]=rmultinom(1,1,as.numeric(px[i,]))
}
```

```
x1=x[1,]
x2=x[2,]
x3=x[3,]

x=colSums(x*c(1,2,3))

#######################################
#generate misclassified W
##################
#parameters for model of W - the alpha parameters
#which come from the misclassification matrix
#where a21 represents the misclassification probability Pr(W=2|X=1)
#and the diagonal element Pr(W=1|X=1)
#represents the non misclassified probability P(W=1|X=1)
#w=.,x=1
a21=.025
a31=.015
a11=1-a21-a31
#w=.,x=2
a22=.85
a32=.05
a12=1-a22-a32
#w=.,x=3
a23=.1
a33=.70
a13=1-a23-a33

#generate w
#Following Hu 2008
# this is used as a threshold to determine what the value of w will be
# based on an observations true x and the misclassification matrix
etaw=runif(n,0,1)

alpha.1=c(0,rep(a11,2),rep(a11+a21,2),1)
alpha.2=c(0,rep(a12,2),rep(a12+a22,2),1)
alpha.3=c(0,rep(a13,2),rep(a13+a23,2),1)

a.1.mat=matrix(alpha.1,nrow=3,ncol=2,byrow=TRUE)
a.2.mat=matrix(alpha.2,nrow=3,ncol=2,byrow=TRUE)
a.3.mat=matrix(alpha.3,nrow=3,ncol=2,byrow=TRUE)
```

```
#a represents the misclassification thresholds
a.mat=array(cbind(a.1.mat,a.2.mat,a.3.mat),dim = c(3,2,3))
#w is the final misclassified values for each observation
w=matrix(,nrow=n,ncol=1)


for (i in 1:n){
w[i,]=sum((etaw[i]>a.mat[,1,m[i]] & etaw[i]<=a.mat[,2,x[i]])*c(1,2,3))
}
w1=as.numeric(w==1)
w2=as.numeric(w==2)
w3=as.numeric(w==3)


#Generate values of parameters for w~ms
wx.out = multinom(w ~ xs1+z+xs2)
#Save estimates from the wx.out object to use as initial estimates in rstan model
wx2int=as.numeric(wx.out$wts[7])
wx2slop1=as.numeric(wx.out$wts[8])
wx2z=as.numeric(wx.out$wts[9])
wx2slop2=as.numeric(wx.out$wts[8])
wx3int=as.numeric(wx.out$wts[12])
wx3slop1=as.numeric(wx.out$wts[13])
wx3z=as.numeric(wx.out$wts[14])
wx3slop2=as.numeric(wx.out$wts[15])
#parameters for model of y
beta0=-2
beta1=1
beta2=0.7
beta3=0.5
py22=exp(beta0+beta1*x2+beta2*x3+beta3*z)/(1+exp(beta0+beta1*x2+beta2*x3+beta3*z))
y=rbinom(n,1,py22)
yw.out=glm(y~w2+w3+z,family=binomial)
#Save estimates to use as initial estimates in rstan model
# which are ywint, ywbeta1, ywbeta2, ywbeta3
yw.coef=yw.out$coef
ywint=as.numeric(yw.out$coef[1])
ywbeta1=as.numeric(yw.out$coef[2])
ywbeta2=as.numeric(yw.out$coef[3])
ywbeta3=as.numeric(yw.out$coef[4])
#Rstan models
#specify data for use in rstan
```

```
#by passing the starting values to a list that rstan uses
stan_data_our=list(N = n, w1=w1, w2=w2,w3=w3,ms1=xs1,ms2=xs2, y = y, z=z,
                   wm2int=wx2int,wm2slop1=wx2slop1, wm2z=wx2z, wm2slop2=wx2slop2,
                   wm3int=wx3int,wm3slop1=wx3slop1, wm3z=wx3z, wm3slop2=wx3slop2,
                   ywint=ywint,ywbeta1=ywbeta1,ywbeta2=ywbeta2,ywbeta3=ywbeta3)

########################################################
#####Computation########################################
#Run models
#generate starting values
#betas for x, w, and our method

#the initial values for eta are set using random uniform
eta_init=runif(6,-1,1)

init_fn2<- function() {
  list(beta1 = ywint,
       beta2 = ywbeta1,
       beta3 = ywbeta2,
       beta4 = ywbeta3,
       gamma21 = wx2int,
       gamma31 = wx3int,
       gamma22 = wx2slop1,
       gamma32 = wx3slop1,
       gamma23 = wx2z,
       gamma33 = wx3z,
       gamma24 = wx2slop2,
       gamma34 = wx3slop2,
       eta1 = eta_init[1] ,
       eta2 = eta_init[2],
       eta3 = eta_init[3],
       eta4 = eta_init[4],
       eta5 = eta_init[5],
       eta6 = eta_init[6]
  )
}

#Run models
#Model using our method
#vb is the function in rstan used to run advi
#the number of iterations is specified by iter option
#the initial values are set using the init option
```

```
#the stopping point based on the change of the relative norm is set using tol_rel_obj
logreg_our_out=vb(logreg_our_model, data=stan_data_our,
                  algorithm = c("fullrank"), iter=20000, seed=nt, init=init_fn2,
                  tol_rel_obj=0.00001,output_samples=10000,refresh=0)


#save posterior estimates of the model
#beta coefficients
ourcoef=c(as.numeric(summary(logreg_our_out)[[1]][1:4,1]))
oursd= c(as.numeric(summary(logreg_our_out)[[1]][1:4,2]))

#other parameters
othercoef=c(as.numeric(summary(logreg_our_out)[[1]][5:18,1]))
othersd= c(as.numeric(summary(logreg_our_out)[[1]][5:18,2]))

#Save results to a matrix
write.table(matrix(as.numeric(c(nt, ourcoef,  oursd)), nrow=1),
'estimates_beta_parameters_n5000.txt', row.names=F,  col.names=F, append=T)
write.table(matrix(as.numeric(c(nt, othercoef,  othersd)), nrow=1),
'estimates_other_paramters_n5000.txt', row.names=F,  col.names=F, append=T)
return(c(ourcoef,  oursd))}


#nrepli represents the replication number
nrepli=200
start = Sys.time()
for (it in 1:nrepli){
  store = myfn1(nt = it)
}

end = Sys.time()
```