Homework 3

Xiaocheng Zhou (1155184323)

April 12, 2023

Answers

1. Please conduct a simulation study for model stated in the question 1. Use bias and RMS to summarize the result of Bayesian analysis based on 10 replications.
   A simulation study was conducted to generate 500 samples for 10 replications, respectively, based on the SEM and true parameters (Appendix 1.1). Then each dataset was applied to estimate the parameters $\boldsymbol{\theta} = [\boldsymbol{\mu}, b, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\Phi}, \boldsymbol{\psi}_\varepsilon, \psi_\delta]$ using the code in Appendix 1.3 based on the true model specified in Appendix 1.2.[1] The bias and RMS of estimates are summarized as Table 1.

$$\text{Bias}(\hat{\theta}) = \frac{1}{R}\sum_{r=1}^{R}\hat{\theta}_r - \theta$$

$$\text{RMS}(\hat{\theta}) = \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\hat{\theta}_r - \theta)^2}$$

Table 1: Bias and RMSE over the 10 replicated estimates

θ = true value	Bias	RMS	θ = true value	Bias	RMS
$\mu_1 = 0$	-0.017	0.144	$\lambda_{6,2} = 0.9$	-0.159	0.197
$\mu_2 = 0$	0.000	0.084	$\lambda_{7,2} = 0.7$	-0.100	0.104
$\mu_3 = 0$	-0.022	0.094	$\lambda_{9,3} = 0.9$	-0.110	0.157
$\mu_4 = 0$	0.001	0.010	$\lambda_{10,3} = 0.8$	0.015	0.103
$\mu_5 = 0$	0.000	0.008	$\gamma_1 = 0.4$	-0.016	0.113
$\mu_6 = 0$	0.003	0.011	$\gamma_2 = 0.5$	-0.055	0.100
$\mu_7 = 0$	-0.000	0.012	$\phi_{1,1} = 1$	0.107	0.114
$\mu_8 = 0$	-0.050	0.132	$\phi_{1,2/2,1} = 0.2$	0.046	0.067
$\mu_9 = 0$	0.005	0.087	$\phi_{2,2} = 0.81$	0.101	0.125
$\mu_{10} = 0$	-0.036	0.081	$\psi_{\varepsilon 4} = 0.3$	0.075	0.075
$b = 0.3$	0.033	0.133	$\psi_{\varepsilon 5} = 0.3$	0.076	0.076
$\lambda_{2,1} = 0.8$	-0.000	0.169	$\psi_{\varepsilon 6} = 0.25$	0.126	0.126
$\lambda_{3,1} = 0.8$	-0.020	0.173	$\psi_{\varepsilon 7} = 0.25$	0.125	0.125
$\lambda_{5,2} = 0.7$	-0.103	0.104	$\psi_\delta = 0.36$	0.007	0.041

2. a. Specify a SEM for this multisample problem, write your model in a matrix form, and state the conditions needed for model identification.

[1] I found it is hard to let the model converge well by checking that plots from different initial values did not meet together even though the Rhat values ~ 1 after 3,5000 iterations, so I try to use the two sets of initial values that are close to the true values to conduct the simulation study, on which the reported results are based.

Let $g = 1$ be the index representing public high school, $g = 2$ private one, and let $i = 1, \cdots, N^{(g)}$ be the index representing the samples collected from group g, i.e. $N^{(1)} = 3074$ and $N^{(2)} = 2909$ then we can specify the SEM as follow. **Measurement equations** are

$$\boldsymbol{v}_i^{(g)} = \boldsymbol{\mu}^{(g)} + \boldsymbol{\Lambda}^{(g)}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\varepsilon}_i^{(g)},$$

and the **Structure equations** can be specified linearly as

$$\eta_i^{(g)} = \boldsymbol{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)},$$

where

$$\boldsymbol{v}_i^{(g)} = \begin{bmatrix} y_{i1}^{(g)} & y_{i2}^{(g)} & y_{i3}^{(g)} & y^{*(g)}_{i1} & y^{*(g)}_{i2} & y^{*(g)}_{i3} & y^{*(g)}_{i4} & y^{*(g)}_{i5} & \vartheta_{i4}^{(g)} & \vartheta_{i5}^{(g)} & \vartheta_{i6}^{(g)} & \vartheta_{i7}^{(g)} \end{bmatrix}^\intercal,$$

$$\boldsymbol{\mu}^{(g)} = \begin{bmatrix} \mu_1^{(g)} & \mu_2^{(g)} & \mu_3^{(g)} & \mu_4^{(g)} & \mu_5^{(g)} & \mu_6^{(g)} & \mu_7^{(g)} & \mu_8^{(g)} & \mu_9^{(g)} & \mu_{10}^{(g)} \end{bmatrix},$$

$$\boldsymbol{\Lambda}^{(g)} = \begin{bmatrix} 1 & \lambda_{2,1}^{(g)} & \lambda_{3,1}^{(g)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \lambda_{5,2}^{(g)} & \lambda_{6,2}^{(g)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \lambda_{8,3}^{(g)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \lambda_{10,4}^{(g)} & \lambda_{11,4}^{(g)} & \lambda_{12,4}^{(g)} \end{bmatrix}^\intercal,$$

$$\boldsymbol{\omega}_i^{(g)} = \begin{bmatrix} \eta_i^{(g)} & \xi_{i1}^{(g)} & \xi_{i2}^{(g)} & \xi_{i3}^{(g)} \end{bmatrix}^\intercal, \ \boldsymbol{\Gamma}^{(g)} = \begin{bmatrix} \gamma_1^{(g)} & \gamma_2^{(g)} & \gamma_3^{(g)} \end{bmatrix}, \boldsymbol{\xi}_i^{g} = \begin{bmatrix} \xi_{i1}^{(g)} & \xi_{i2}^{(g)} & \xi_{i3}^{(g)} \end{bmatrix}^\intercal,$$

$$\text{and } \boldsymbol{\varepsilon}_i^{(g)} = \begin{bmatrix} \varepsilon_{i1}^{(g)} & \varepsilon_{i2}^{(g)} & \varepsilon_{i3}^{(g)} & \varepsilon_{i4}^{(g)} & \varepsilon_{i5}^{(g)} & \varepsilon_{i6}^{(g)} & \varepsilon_{i7}^{(g)} & \varepsilon_{i8}^{(g)} & 0 & 0 & 0 & 0 \end{bmatrix}, \ \varepsilon_{ik}^{(g)} \sim \mathcal{N}(0, \psi_{\varepsilon,k}^{(g)}), \ k = 1, \cdots, 8.$$

in which $\boldsymbol{y}^{*(g)}_{i,1:5}$ are the latent continuous measurement for ordered categorical variables $\boldsymbol{z}_{i,1:5}^{(g)}$ and $\boldsymbol{\vartheta}_{i,4:7}^{(g)}$ are the canonical parameters for $\boldsymbol{y}_{i,4:7}^{(g)}$ that are from EFDs. Given shared threshold parameters $\boldsymbol{\alpha}_{k,1:b_k}$ for each $y^{*(g)}_{ik}$, the relationship between latent continuous values and observed values can be specified as

$$z_{ik}^{(g)} = m \text{ if } \alpha_{k,m} \leq y^{*(g)}_{ik} < \alpha_{k,m+1}, \ k = 1, \cdots, 5, m = 0, 1, \cdots, b_k$$
$$p\left(y_{ik}^{(g)} | \boldsymbol{\omega}_i^{(g)}\right) = \exp\left\{\left[y_{ik}^{(g)}\vartheta_{ik}^{(g)} - b(\vartheta_{ik}^{(g)})\right]/\psi_{\varepsilon,k+3}^{(g)} + c_k\left(y_{ik}^{(g)}, \psi_{\varepsilon,k+3}^{(g)}\right)\right\}, \ k = 4, \cdots, 7.$$

To identify this SEM with multisample data, we impose following identification conditions

i) to identify each measurement equation, fix $\lambda_{1,1} = \lambda_{4,2} = \lambda_{7,3} = \lambda_{9,4} = 1$ and others, like in $\boldsymbol{\Gamma}^{(g)}$, equal 0;
ii) to identify the issue induced by ordered categorical variables, unify the normal distribution of latent $y^{*(g)}_{ik}$ implicitly by fix $\alpha_{k,1}^{(1)} = \Phi^{-1}(f_{k,1}^{(1)})$ and $\alpha_{k,b_k}^{(1)} = \Phi^{-1}(f_{k,b_k}^{(1)})$;
iii) specially, if the ordered categorical variables is dichotomous, then unify the normal distribution of latent $y^{*(g)}_{ik}$ directly by fixing the $\psi_{\varepsilon K} = 1$;
iv) to let the latent continuous variables share scale among groups, e.g. select the first group as the reference and impose $\boldsymbol{\alpha}_k^{(g)} = \boldsymbol{\alpha}_k^{(1)} =: \boldsymbol{\alpha}_k, \ k = 1, \cdots, 5, \ g = 1, 2.$

b. Describe the major difference in the posterior inference of SEM with multisample data.

The major difference exists in the estimation of $[\boldsymbol{\theta}|\boldsymbol{\alpha}, \mathbf{Y}, \boldsymbol{\Omega}, \mathbf{X}, \mathbf{X}, \mathbf{Z}]$ the specification of their prior distributions:

i) For **nonconstrained** parameters, their priors in different groups are naturally assumed to be independent, so in estimating the unconstrained parameters, the *prior distributions specified for each group* and *the group-corresponding data* are used.
ii) For **constrained** parameters, *only one prior distribution* for these constrained parameters is needed, and *all the data over all groups* should be combined in the estimation.
iii) Moreover, the original dependent parameters are assumed to have *independent priors*, i.e. $p(\boldsymbol{\Lambda}^{(g)}, \boldsymbol{\Psi}_\varepsilon^{(g)}) = p(\boldsymbol{\Lambda}^{(g)})p(\boldsymbol{\Psi}_\varepsilon^{(g)})$ for *both variant or invariant case.* Because in the estimation of some dependent parameters, such as $\left[\boldsymbol{\Lambda}_k^\intercal | \psi_{\varepsilon k}^{(g)}\right]$ in $\mathcal{M}_1 : \boldsymbol{\Lambda}^{(1)} = \boldsymbol{\Lambda}^{(2)} = \boldsymbol{\Lambda}, \boldsymbol{\Psi}_\varepsilon^{(1)} \neq \boldsymbol{\Psi}_\varepsilon^{(2)}$, it is *not suitable* to estimate $\left[\boldsymbol{\Lambda}_k^\intercal, \psi_{\varepsilon k}^{(g)}\right]$ *jointly*, since it is difficult to select a $\boldsymbol{\Lambda}_k^\intercal$ with a set different $\psi_{\varepsilon k}^{(g)}$ for different groups. And for convenience, we assume independence uniformly.

c. Briefly describe how to test the invariant constraint for factor loadings across the subpopulations using Bayes factor and DIC. [Hint: the major steps of BF/DIC calculation across iterations]
To test the invariant constrant for **factor loadings** across subpopulations, we specify two models, $\mathcal{M}_1$ and $\mathcal{M}_2$

$$\mathcal{M}_1 : \begin{cases} \boldsymbol{v}_i^{(g)} = \boldsymbol{\mu}^{(g)} + \boldsymbol{\Lambda}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\varepsilon}_i^{(g)} \\ \eta_i^{(g)} = \boldsymbol{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)} \end{cases} \tag{1}$$

$$\mathcal{M}_2 : \begin{cases} \boldsymbol{v}_i^{(g)} = \boldsymbol{\mu}^{(g)} + \boldsymbol{\Lambda}^{(g)}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\varepsilon}_i^{(g)} \\ \eta_i^{(g)} = \boldsymbol{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)} \end{cases} \tag{2}$$

$$i = 1, \cdots, N^{(g)},\ \ g = 1, 2$$

Bayes Factor

(1) Find a link model $\mathcal{M}_t$ with path $t \in [0,1]$ to link $\mathcal{M}_1$ (when $t = 0$) and $\mathcal{M}_2$ (when $t = 1$) directly.

$$\mathcal{M}_t : \begin{cases} \boldsymbol{v}_i^{(g)} = \boldsymbol{\mu}^{(g)} + (1-t)\boldsymbol{\Lambda}\boldsymbol{\omega}_i^{(g)} + t\boldsymbol{\Lambda}^{(g)}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\varepsilon}_i^{(g)} \\ \eta_i^{(g)} = \boldsymbol{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)} \end{cases} \tag{3}$$

$$i = 1, \cdots, N^{(g)},\ \ g = 1, 2$$

(2) Differentiate the complete-data log-likelihood. Since $\boldsymbol{\varepsilon}_{i,-9:12}^{(g)} = \boldsymbol{v}_{i,-9:12}^{(g)} - [\boldsymbol{\mu}_{-9:12}^{(g)} + (1-t)\boldsymbol{\Lambda}_{-4}\boldsymbol{\omega}_{i,-4}^{(g)} + t\boldsymbol{\Lambda}_{-4}^{(g)}\boldsymbol{\omega}_{i,-4}^{(g)}] \sim \mathcal{N}(0, \text{diag}(\boldsymbol{\psi}_{\varepsilon,-9:12}))$, then

$$\begin{aligned}
y_{i1}^{(g)} &= \mu_1^{(g)} + (1-t)\eta_i^{(g)} + t\eta_i^{(g)} + \varepsilon_{i1}^{(g)} \\
y_{i2}^{(g)} &= \mu_2^{(g)} + (1-t)\lambda_{2,1}\eta_i^{(g)} + t\lambda_{2,1}^{(g)}\eta_i^{(g)} + \varepsilon_{i2}^{(g)} \\
y_{i3}^{(g)} &= \mu_3^{(g)} + (1-t)\lambda_{3,1}\eta_i^{(g)} + t\lambda_{3,1}^{(g)}\eta_i^{(g)} + \varepsilon_{i3}^{(g)} \\
y^{*(g)}_{i1} &= \mu_4^{(g)} + (1-t)\xi_{i1}^{(g)} + t\xi_{i1}^{(g)} + \varepsilon_{i4}^{(g)} \\
y^{*(g)}_{i2} &= \mu_5^{(g)} + (1-t)\lambda_{5,2}\xi_{i1}^{(g)} + t\lambda_{5,2}^{(g)}\xi_{i1}^{(g)} + \varepsilon_{i5}^{(g)} \\
y^{*(g)}_{i3} &= \mu_6^{(g)} + (1-t)\lambda_{6,2}\xi_{i1}^{(g)} + t\lambda_{6,2}^{(g)}\xi_{i1}^{(g)} + \varepsilon_{i6}^{(g)} \\
y^{*(g)}_{i4} &= \mu_7^{(g)} + (1-t)\xi_{i2}^{(g)} + t\xi_{i2}^{(g)} + \varepsilon_{i7}^{(g)} \\
y^{*(g)}_{i5} &= \mu_8^{(g)} + (1-t)\lambda_{8,3}\xi_{i2}^{(g)} + t\lambda_{8,3}^{(g)}\xi_{i2}^{(g)} + \varepsilon_{i8}^{(g)}
\end{aligned}$$

$$U(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{Y}, \mathbf{V}, \boldsymbol{\Omega}, \mathbf{X}, \mathbf{Z}, t) = \frac{d}{dt}\log p(\mathbf{Y}, \mathbf{V}, \boldsymbol{\Omega}, \mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\alpha}, t)$$

$$= \sum_{g=1}^{2}\sum_{i=1}^{N^{(g)}} \left\{ \begin{array}{l}
\left[y_{i2}^{(g)} - \left(\mu_2^{(g)} + (1-t)\lambda_{2,1}\eta_i^{(g)} + t\lambda_{2,1}^{(g)}\eta_i^{(g)}\right)\right]\left(-\lambda_{2,1}\eta_i^{(g)} + \lambda_{2,1}^{(g)}\eta_i^{(g)}\right)/\psi_{\varepsilon 2}^{(g)} + \\
\left[y_{i3}^{(g)} - \left(\mu_3^{(g)} + (1-t)\lambda_{3,1}\eta_i^{(g)} + t\lambda_{3,1}^{(g)}\eta_i^{(g)}\right)\right]\left(-\lambda_{3,1}\eta_i^{(g)} + \lambda_{3,1}^{(g)}\eta_i^{(g)}\right)/\psi_{\varepsilon 3}^{(g)} + \\
\left[y^{*(g)}_{i2} - \left(\mu_5^{(g)} + (1-t)\lambda_{5,2}\xi_{i1}^{(g)} + t\lambda_{5,2}^{(g)}\xi_{i1}^{(g)}\right)\right]\left(-\lambda_{5,2}\xi_{i1}^{(g)} + \lambda_{5,2}^{(g)}\xi_{i1}^{(g)}\right)/\psi_{\varepsilon 5}^{(g)} + \\
\left[y^{*(g)}_{i3} - \left(\mu_6^{(g)} + (1-t)\lambda_{6,2}\xi_{i1}^{(g)} + t\lambda_{6,2}^{(g)}\xi_{i1}^{(g)}\right)\right]\left(-\lambda_{6,2}\xi_{i1}^{(g)} + \lambda_{6,2}^{(g)}\xi_{i1}^{(g)}\right)/\psi_{\varepsilon 6}^{(g)} + \\
\left[y^{*(g)}_{i5} - \left(\mu_8^{(g)} + (1-t)\lambda_{8,3}\xi_{i2}^{(g)} + t\lambda_{8,3}^{(g)}\xi_{i2}^{(g)}\right)\right]\left(-\lambda_{8,3}\xi_{i2}^{(g)} + \lambda_{8,3}^{(g)}\xi_{i2}^{(g)}\right)/\psi_{\varepsilon 8}^{(g)}
\end{array} \right\}$$

where $\mathbf{Y} = \left[\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}\right]$, latent continuous measurements, contains all $\boldsymbol{y}^{*(g)}_{i,1:5}$ and $\boldsymbol{\vartheta}_{i,4:7}^{(g)}$, $\mathbf{V} = \left[\mathbf{V}^{(1)}, \mathbf{V}^{(2)}\right]$, observed variables from EFDs, contains all $\boldsymbol{y}_{i,4:7}^{(g)}$ $\mathbf{X} = \left[\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right]$, observed continuous data, contains all $\boldsymbol{y}_{i,1:3}^{(g)}$, $\mathbf{Z} = \left[\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\right]$, observed ordered categorical data, contains all $\boldsymbol{z}_{i,1:5}^{(g)}$, $\boldsymbol{\Omega} = \left[\boldsymbol{\Omega}^{(1)}, \boldsymbol{\Omega}^{(2)}\right]$, latent variables, contains all $\eta_i^{(g)}$ and $\boldsymbol{\xi}_{i,1:3}^{(g)}$, and $\boldsymbol{\theta}$ contains all the parameters to be estimated, $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}$, $\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}$, $\boldsymbol{\Psi}_\varepsilon^{(1)}, \boldsymbol{\Psi}_\varepsilon^{(2)}$, $\boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(2)}$, $\boldsymbol{\Gamma}^{(1)}, \boldsymbol{\Gamma}^{(2)}$, $\boldsymbol{\Psi}_\delta^{(1)}, \boldsymbol{\Psi}_\delta^{(2)}$

(3) Calculate the estimated log-BF by dividing $[0,1]$ into $S+1$ segments such that $0 = t_{(0)} < t_{(1)} < \cdots < t_{(S)} < t_{(S+1)} = 1$ and sampling J observations simulated from the joint posterior

distribution $[\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{Y}, \mathbf{V}, \boldsymbol{\Omega}|\mathbf{X}, \mathbf{Z}, t_{(s)}]$.

$$\widehat{\log B_{21}} = \frac{1}{2}\sum_{s=0}^{S}(t_{(s+1)} - t_{(s)})(\bar{U}_{(s+1)} + \bar{U}_s)$$
$$\bar{U}_{(s)} = \frac{1}{J}\sum_{j=1}^{J} U(\boldsymbol{\theta}^{(j)}, \boldsymbol{\alpha}^{(j)}, \mathbf{Y}^{(j)}, \mathbf{V}^{(j)}, \boldsymbol{\Omega}^{(j)}, \mathbf{X}, \mathbf{Z}, t_{(s)})$$

DIC

$$\mathrm{DIC}_k = \bar{D}(\boldsymbol{\theta}_k) + d_k$$
$$\text{(goodness-of-fit) } \bar{D}(\boldsymbol{\theta}_k) = \mathbb{E}_{\boldsymbol{\theta}_k}\left[-2\log p(\mathbf{V}, \mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}_k, \mathcal{M}_k)|\mathbf{V}, \mathbf{X}, \mathbf{Z}\right]$$
$$\text{(effective \#params) } d_k = \mathbb{E}_{\boldsymbol{\theta}_k}\left[-2\log p(\mathbf{V}, \mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}_k, \mathcal{M}_k)|\mathbf{V}, \mathbf{X}, \mathbf{Z}\right] + 2\log p(\mathbf{V}, \mathbf{X}, \mathbf{Z}|\tilde{\boldsymbol{\theta}}_k)$$
$$\mathbb{E}_{\boldsymbol{\theta}_k}\left[-2\log p(\mathbf{V}, \mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}_k, \mathcal{M}_k)|\mathbf{V}, \mathbf{X}, \mathbf{Z}\right] \approx -\frac{2}{J}\sum_{j=1}^{J}\log p(\mathbf{V}, \mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}_k^{(j)}, \mathcal{M}_k),$$

where $\tilde{\boldsymbol{\theta}}_k$ is the Bayesian estimate of $\boldsymbol{\theta}_k$ and $\{\boldsymbol{\theta}_k^{(j)}, j = 1, \cdots, J\}$ are a set of observations simulated from the posterior distribution. The DIC values can be calculated automatically when using `WinBUGS` to fit $\mathcal{M}_1$ and $\mathcal{M}_2$.

1 Appendix

1.1 Simulation data generation

```
library(mvtnorm)

# set data repository
datapath = paste0(getwd(),'/data')
dir.create(datapath, showWarnings = FALSE, recursive = TRUE)

# data size
iter = 10
N = 500
P = 10
Neta = 1 # not used here
Nxi = 2 #
Ngam = 2 #

# set the true values of parameters
uby <- rep(0, P)
lam <- c(0.8, 0.8, 0.7, 0.9, 0.7, 0.9, 0.8)
sgm <- c(1, 1, 1, 0.3, 0.3, 0.25, 0.25)
ubeta <- 0.3
gam <- c(0.4, 0.5)
phx <- matrix(data = c(1, 0.2, 0.2, 0.81), ncol = 2)
sgd <- 0.36

# set important prior param R_0
# R0 <- matrix(c(7.0, 2.1, 2.1, 7.0), nrow = 2)


# containers for generated data
Y <- matrix(data = NA, nrow = N, ncol = P)
D <- numeric(N)
p <- numeric(P)
v <- numeric(P)


# generate data
for (t in 1:iter) {
  for (i in 1:N) {
    # BD[i] = rt(1, 5)
    # BC[i] = rt(1, 5)

    # generate the fixed covariates in SE (from Bernoulli(0.7))
    d <- rbinom(1, 1, 0.7)
    D[i] <- d

    # generate xi
    xi <- rmvnorm(1, c(0, 0), phx)

    # generate error term in SE
    del <- rnorm(1, 0, sqrt(sgd))

    # generate eta according to the SE
    eta <- ubeta * d + gam[1] * xi[1] + gam[2] * xi[2] + del

    # generate error term in ME
    eps <- numeric(7)
    for (k in 1:7) { eps[k] <- rnorm(1, 0, sgm[k]) }

    # generate theta in ME
    v[1] <- uby[1] + eta + eps[1]
    v[2] <- uby[2] + lam[1] * eta + eps[2]
    v[3] <- uby[3] + lam[2] * eta + eps[3]
    Y[i, 4] <- uby[4] + xi[1] + eps[4]
    Y[i, 5] <- uby[5] + lam[3] * xi[1] + eps[5]
    Y[i, 6] <- uby[6] + lam[4] * xi[1] + eps[6]
    Y[i, 7] <- uby[7] + lam[5] * xi[1] + eps[7]
    v[8] <- uby[8] + xi[2]
```

```
    v[9] <- uby[9] + lam[6] * xi[2]
    v[10] <- uby[10] + lam[7] * xi[2]

    # transform theta to ordinal variables
    for (j in 1:3) {
      if (v[j] > 0) Y[i, j] <- 1
      else Y[i, j] <- 0
    }

    # transform theta to binary variables
    for (j in 8:10) {
      p[j] <- exp(v[j]) / (1 + exp(v[j]))
      Y[i, j] <- rbinom(1, 1, p[j])
    }
  }

  # save data matrix
  write.table(Y, paste(datapath, "/Y-", t, ".txt", sep = ""))
  write.table(D, paste(datapath, "/D-", t, ".txt", sep = ""))
}

true_params = list(
  lam = lam,
  uby = uby,
  sgm = sgm,
  ubeta = ubeta,
  gam = gam,
  phx = phx,
  sgd = sgd
)

save(true_params, file = paste0(datapath, "/trueparams.RData"))
```

codes/generateData.R

1.2 True model in BUGS language

```
model{
  for (i in 1:N) {
    # measurement equation model
    for (j in 1:3) {
      y[i, j] ~ dnorm(mu[i, j], 1)I(low[z[i, j] + 1], high[z[i, j] + 1])
    }
    ## winbugs cannot handle operation, in index variables, like k = j - 2 !!!
    y[i, 4] ~ dnorm(mu[i, 4], psi[1])
    y[i, 5] ~ dnorm(mu[i, 5], psi[2])
    y[i, 6] ~ dnorm(mu[i, 6], psi[3])
    y[i, 7] ~ dnorm(mu[i, 7], psi[4])
    for (j in 8:P) {
      z[i, j] ~ dbin(pb[i, j], 1)
      pb[i, j] <- exp(mu[i, j]) / (1 + exp(mu[i, j]))
    }

    mu[i, 1] <- uby[1] + eta[i]
    mu[i, 2] <- uby[2] + lam[1] * eta[i]
    mu[i, 3] <- uby[3] + lam[2] * eta[i]
    mu[i, 4] <- uby[4] + xi[i, 1]
    mu[i, 5] <- uby[5] + lam[3] * xi[i, 1]
    mu[i, 6] <- uby[6] + lam[4] * xi[i, 1]
    mu[i, 7] <- uby[7] + lam[5] * xi[i, 1]
    mu[i, 8] <- uby[8] + xi[i, 2]
    mu[i, 9] <- uby[9] + lam[6] * xi[i, 2]
    mu[i, 10] <- uby[10] + lam[7] * xi[i, 2]


    # structural equation model
    xi[i, 1:2] ~ dmnorm(zero2[1:2], phi[1:2, 1:2])
    eta[i] ~ dnorm(etamu[i], psd)

```

```
    etamu[i] <- ubeta * d[i] + gam[1] * xi[i, 1] + gam[2] * xi[i, 2]
  } # End for i

  for (k in 1:2) { zero2[k] <- 0 }


  # priors inputs for loadings and coefficients
  for (j in 1:P) { uby[j] ~ dnorm(0.0, 4.0) }

  for (j in 1:2) { lam[j] ~ dnorm(0.5, 4.0) }

  pslam[1] <- 4.0 * psi[2]
  lam[3] ~ dnorm(0.5, pslam[1])
  pslam[2] <- 4.0 * psi[3]
  lam[4] ~ dnorm(0.5, pslam[2])
  pslam[3] <- 4.0 * psi[4]
  lam[5] ~ dnorm(0.5, pslam[3])

  for (j in 6:7) { lam[j] ~ dnorm(0.5, 4.0) }

  ubeta ~ dnorm (0.5, psd)

  psgam <- 4.0 * psd
  for (k in 1:2) {
    gam[k] ~ dnorm(0.5, psgam)
  }


  # priors inputs for precisions
  for (j in 1:4) {
    psi[j] ~ dgamma(9, 3)
    sgm[j] <- 1 / psi[j]
  }

  psd ~ dgamma(9, 3)
  sgd <- 1 / psd

  phi[1:2, 1:2] ~ dwish(R[1:2, 1:2], 4)
  phx[1:2, 1:2] <- inverse(phi[1:2, 1:2])
} # End of model
```

codes/model1.txt

1.3 Simulation process

```
library(R2WinBUGS)

# set experiment date
timestamp = strftime(Sys.time(), "%Y%m%d-%H")
winBUGS.path = "D:/pkgs/WinBUGS14/"
datapath = paste0(getwd(),'/data')
print(datapath)


# data size
iter = 10
N = 500
P = 10
Nlam = 7
Neta = 1
Nxi = 2
Ngam = 2

# containers for Bayesian estimates and standard errors
uby.E <- matrix(data = NA, nrow = iter, ncol = P)
uby.SE <- matrix(data = NA, nrow = iter, ncol = P)
lam.E <- matrix(data = NA, nrow = iter, ncol = Nlam)
lam.SE <- matrix(data = NA, nrow = iter, ncol = Nlam)
gam.E <- matrix(data = NA, nrow = iter, ncol = Ngam)
```

```
gam.SE <- matrix(data = NA, nrow = iter, ncol = Ngam)
phx.E <- matrix(data = NA, nrow = iter, ncol = Nxi^2)
phx.SE <- matrix(data = NA, nrow = iter, ncol = Nxi^2)
ubeta.E <- numeric(iter)
ubeta.SE <- numeric(iter)
# only continuous part of y's variances need to estimate
sgm.E <- matrix(data = NA, nrow = iter, ncol = 4)
sgm.SE <- matrix(data = NA, nrow = iter, ncol = 4)
sgd.E <- numeric(iter)
sgd.SE <- numeric(iter)

# containers for HPD (Highest Probability Density) intervals
uby.hpd <- array(NA, c(iter, P, 2))
ubeta.hpd <- array(NA, c(iter, 2))
lam.hpd <- array(NA, c(iter, Nlam, 2))
gam.hpd <- array(NA, c(iter, Ngam, 2))
phx.hpd <- array(NA, c(iter, Nxi^2, 2))
sgm.hpd <- array(NA, c(iter, 4, 2))
sgd.hpd <- array(NA, c(iter, 2))

# container for DIC values
DIC = numeric(iter)

# parameters to be estimated
parameters <- c("uby", "ubeta", "lam", "gam", "phx", "sgm", "sgd")

# set important prior param R_0
R0 = matrix(c(7.0, 2.1, 2.1, 7.0), nrow = 2)

# initial values for MCMC in WinBUGS
init1 <- list(
  uby = rep(0.5, P),
  ubeta = 0.5,
  lam = rep(0.5, Nlam),
  gam = rep(0.5, Ngam),
  phi = matrix(c(1, 0.5, 0.5, 1), nrow = Nxi),
  psi = rep(1, 4),
  psd = 1,
  xi = matrix(data = rep(0.3, N * Nxi), ncol = Nxi)
)

init2 <- list(
  uby = rep(0, P),
  ubeta = 0,
  lam = rep(0, Nlam),
  gam = rep(0, Ngam),
  phi = matrix(c(2, 0, 0, 2), nrow = Nxi),
  psi = rep(2, 4),
  psd = 2,
  xi = matrix(data = rep(0.3, N * Nxi), ncol = Nxi)
)

init3 <- list(
  uby = rep(0, P),
  ubeta = 0.3,
  lam = rep(0.8, Nlam),
  gam = rep(0.45, Ngam),
  phi = matrix(c(1, 0.2, 0.2, 1), nrow = Nxi),
  psi = rep(0.3, 4),
  psd = 0.4,
  xi = matrix(data = rep(0.3, N * Nxi), ncol = Nxi)
)

init4 <- list(
  uby = rep(0, P),
  ubeta = 0.4,
  lam = rep(0.9, Nlam),
  gam = rep(0.5, Ngam),
  phi = matrix(c(1, 0.3, 0.3, 1), nrow = Nxi),
  psi = rep(0.4, 4),
  psd = 0.3,
```

```
  xi = matrix(data = rep(0.3, N * Nxi), ncol = Nxi)
)

inits <- list(init3, init4)


# Do simulations based on 10 replications
for (t in 1:iter) {
  iterpath = paste0(getwd(),"/Q1.2_", t)
  dir.create(iterpath, showWarnings = FALSE, recursive = TRUE)

  Y <- as.matrix(read.table(paste(datapath, "/Y-", t, ".txt", sep = "")))
  D <- read.table(paste(datapath, "/D-", t, ".txt", sep = ""))$x

  data <- list(N = N, P = P, R = R0,
               d = D, z = Y,
               low = c(-2000, 0),
               high = c(0, 2000))

  model = bugs(data, inits, parameters,
                model.file = paste0(getwd(),"/../model1.txt"),
                n.chains = 2,
                n.iter = 5000,
                n.burnin = 3000,
                n.thin = 1,
                DIC = TRUE,
                bugs.directory = winBUGS.path,
                working.directory = iterpath,
                debug = FALSE)

  # save estimates and standard errors
  uby.E[t, ] = model$mean$uby
  uby.SE[t, ] = model$sd$uby
  ubeta.E[t] = model$mean$ubeta
  ubeta.SE[t] = model$sd$ubeta
  lam.E[t, ] = model$mean$lam
  lam.SE[t, ] = model$sd$lam
  gam.E[t, ] = model$mean$gam
  gam.SE[t, ] = model$sd$gam
  phx.E[t, ] = c(model$mean$phx)
  phx.SE[t, ] = c(model$sd$phx)
  sgm.E[t, ] = model$mean$sgm
  sgm.SE[t, ] = model$sd$sgm
  sgd.E[t] = model$mean$sgd
  sgd.SE[t] = model$sd$sgd

  # save HPD intervals
  for (k in 1:P) {
    temp = model$sims.array[ , 1, k]
    uby.hpd[t, k, ] = boa.hpd(temp, 0.05)
  }
  temp = model$sims.array[ , 1, P + 1]
  ubeta.hpd[t, ] = boa.hpd(temp, 0.05)
  for (k in 1:Nlam) {
    temp = model$sims.array[ , 1, P + 1 + k]
    lam.hpd[t, k, ] = boa.hpd(temp, 0.05)
  }
  for (k in 1:Ngam) {
    temp = model$sims.array[ , 1, P + 1 + Nlam + k]
    gam.hpd[t, k, ] = boa.hpd(temp, 0.05)
  }
  for (k in 1:Nxi^2) {
    temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + k]
    phx.hpd[t, k, ] = boa.hpd(temp, 0.05)
  }
  for (k in 1:4) {
    temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + Nxi^2 + k]
    sgm.hpd[t, k, ] = boa.hpd(temp, 0.05)
  }
  temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + Nxi^2 + 4 + 1]
  sgd.hpd[t, ] = boa.hpd(temp, 0.05)
```

```r

  # save DIC values
  DIC[t] = model$DIC

  print(model$summary)
}

metr_params = list(
  E = list(
    uby = uby.E, ubeta = ubeta.E, lam = lam.E, gam = gam.E,
    phx = phx.E, sgm = sgm.E, sgd = sgd.E
  ),
  SE = list(
    uby = uby.SE, ubeta = ubeta.SE, lam = lam.SE, gam = gam.SE,
    phx = phx.SE, sgm = sgm.SE, sgd = sgd.SE
  ),
  HPD = list(
    uby = uby.hpd, ubeta = ubeta.hpd, lam = lam.hpd, gam = gam.hpd,
    phx = phx.hpd, sgm = sgm.hpd, sgd = sgd.hpd
  )
)

save(metr_params, file = paste0(getwd(), "/Q1.2_metrparams.RData"))
```

codes/simulation1.R