

Homework 3

Xiaocheng Zhou (1155184323)

April 10, 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 $\theta = [\mu, b, \lambda, \gamma, \Phi, \psi_\varepsilon, \psi_\delta]$ 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
$\mu_1 = 0$	0.036	0.120	$\lambda_{6,2} = 0.9$	-0.397	0.397
$\mu_2 = 0$	-0.015	0.125	$\lambda_{7,2} = 0.7$	-0.194	0.194
$\mu_3 = 0$	-0.019	0.122	$\lambda_{9,3} = 0.9$	0.240	0.313
$\mu_4 = 0$	-0.007	0.015	$\lambda_{10,3} = 0.8$	0.175	0.220
$\mu_5 = 0$	-0.002	0.013	$\gamma_1 = 0.4$	0.106	0.119
$\mu_6 = 0$	0.003	0.010	$\gamma_2 = 0.5$	0.038	0.110
$\mu_7 = 0$	-0.002	0.010	$\phi_{1,1} = 1$	-0.738	0.744
$\mu_8 = 0$	-0.052	0.111	$\phi_{1,2/2,1} = 0.2$	0.047	0.098
$\mu_9 = 0$	-0.006	0.083	$\phi_{2,2} = 0.81$	-0.248	0.293
$\mu_{10} = 0$	-0.047	0.092	$\psi_{\varepsilon 4} = 0.3$	0.075	0.075
$b = 0.3$	-0.029	0.111	$\psi_{\varepsilon 5} = 0.3$	0.073	0.073
$\lambda_{2,1} = 0.8$	0.076	0.167	$\psi_{\varepsilon 6} = 0.25$	0.126	0.126
$\lambda_{3,1} = 0.8$	0.141	0.187	$\psi_{\varepsilon 7} = 0.25$	0.125	0.125
$\lambda_{5,2} = 0.7$	-0.208	0.209	$\psi_{\delta} = 0.36$	0.027	0.059

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

¹I found it is hard to let the model converge well by checking that plots from different initial values do not meet together even though the Rhat values ~ 1 , so I try to use the a set of initial values that are close to the true values to conduct the simulation study. and

Let $g = 1$ be the index representing public high school, $g = 2$ private one, and let $i = 1, \dots, N^{(g)}$ 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

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

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

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

where

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

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

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

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

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

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

$$z_{ik}^{(g)} = m \text{ if } \alpha_{k,m} \leq y_{ik}^{*(g)} < \alpha_{k,m+1}, k = 1, \dots, 5, m = 0, 1, \dots, b_k$$

$$p(y_{ik}^{(g)} | \boldsymbol{\omega}_i^{(g)}) = \exp \left\{ \left[y_{ik}^{(g)} \vartheta_{ik}^{(g)} - b(\vartheta_{ik}^{(g)}) \right] / \psi_{\varepsilon,k+3}^{(g)} + c_k(y_{ik}^{(g)}, \psi_{\varepsilon,k+3}^{(g)}) \right\}, k = 4, \dots, 7.$$

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 $y_{ik}^{*(g)}$ 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 $y_{ik}^{*(g)}$ 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 $\boldsymbol{\alpha}_k^{(g)} = \boldsymbol{\alpha}_k^{(1)} =: \boldsymbol{\alpha}_k, k = 1, \dots, 5, g = 1, 2$.
- b. **Describe the major difference in the posterior inference of SEM with multisample data.** The major difference exists in the estimation of $[\boldsymbol{\theta} | \boldsymbol{\alpha}, \mathbf{Y}, \boldsymbol{\Omega}, \mathbf{X}, \mathbf{Z}]$ 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 $[\boldsymbol{\Lambda}_k^T | \psi_{\varepsilon k}^{(g)}]$ 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 $[\boldsymbol{\Lambda}_k^T, \psi_{\varepsilon k}^{(g)}]$ *jointly*, since it is difficult to select a $\boldsymbol{\Lambda}_k^T$ 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 constraint for **factor loadings** across subpopulations, we specify two models, \mathcal{M}_1 and \mathcal{M}_2

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

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

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

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.

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

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

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

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

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

$$= \sum_{g=1}^2 \sum_{i=1}^{N^{(g)}} \left\{ \begin{aligned} &\left[y_{i2}^{(g)} - \left(\mu_2^{(g)} + (1-t)\lambda_{2,1}\eta_i^{(g)} + t\lambda_{2,1}\eta_i^{(g)} \right) \right] \left(-\lambda_{2,1}\eta_i^{(g)} + \lambda_{2,1}\eta_i^{(g)} \right) / \psi_{\varepsilon 2}^{(g)} + \\ &\left[y_{i3}^{(g)} - \left(\mu_3^{(g)} + (1-t)\lambda_{3,1}\eta_i^{(g)} + t\lambda_{3,1}\eta_i^{(g)} \right) \right] \left(-\lambda_{3,1}\eta_i^{(g)} + \lambda_{3,1}\eta_i^{(g)} \right) / \psi_{\varepsilon 3}^{(g)} + \\ &\left[y_{i5}^{(g)} - \left(\mu_5^{(g)} + (1-t)\lambda_{5,2}\xi_{i1}^{(g)} + t\lambda_{5,2}\xi_{i1}^{(g)} \right) \right] \left(-\lambda_{5,2}\xi_{i1}^{(g)} + \lambda_{5,2}\xi_{i1}^{(g)} \right) / \psi_{\varepsilon 5}^{(g)} + \\ &\left[y_{i6}^{(g)} - \left(\mu_6^{(g)} + (1-t)\lambda_{6,2}\xi_{i1}^{(g)} + t\lambda_{6,2}\xi_{i1}^{(g)} \right) \right] \left(-\lambda_{6,2}\xi_{i1}^{(g)} + \lambda_{6,2}\xi_{i1}^{(g)} \right) / \psi_{\varepsilon 6}^{(g)} + \\ &\left[y_{i8}^{(g)} - \left(\mu_8^{(g)} + (1-t)\lambda_{8,3}\xi_{i2}^{(g)} + t\lambda_{8,3}\xi_{i2}^{(g)} \right) \right] \left(-\lambda_{8,3}\xi_{i2}^{(g)} + \lambda_{8,3}\xi_{i2}^{(g)} \right) / \psi_{\varepsilon 8}^{(g)} \end{aligned} \right\}$$

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

- (3) Calculate the estimated log-BF by dividing $[0, 1]$ into $S + 1$ segments such that $0 = t_{(0)} < t_{(1)} < \dots < t_{(S)} < t_{(S+1)} = 1$ 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

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

```
1 library(mvtnorm)
2
3 # set data repository
4 datapath = paste0(getwd(), '/data')
5 dir.create(datapath, showWarnings = FALSE, recursive = TRUE)
6
7 # data size
8 iter = 10
9 N = 500
10 P = 10
11 Neta = 1 # not used here
12 Nxi = 2 #
13 Ngam = 2 #
14
15 # set the true values of parameters
16 uby <- rep(0, P)
17 lam <- c(0.8, 0.8, 0.7, 0.9, 0.7, 0.9, 0.8)
18 sgm <- c(1, 1, 1, 0.3, 0.3, 0.25, 0.25)
19 ubeta <- 0.3
20 gam <- c(0.4, 0.5)
21 phx <- matrix(data = c(1, 0.2, 0.2, 0.81), ncol = 2)
22 sgd <- 0.36
23
24 # set important prior param R_0
25 # R0 <- matrix(c(7.0, 2.1, 2.1, 7.0), nrow = 2)
26
27
28 # containers for generated data
29 Y <- matrix(data = NA, nrow = N, ncol = P)
30 D <- numeric(N)
31 p <- numeric(P)
32 v <- numeric(P)
33
34
35 # generate data
36 for (t in 1:iter) {
37   for (i in 1:N) {
38     # BD[i] = rt(1, 5)
39     # BC[i] = rt(1, 5)
40
41     # generate the fixed covariates in SE (from Bernoulli(0.7))
42     d <- rbinom(1, 1, 0.7)
43     D[i] <- d
44
45     # generate xi
46     xi <- rmvnorm(1, c(0, 0), phx)
47
48     # generate error term in SE
49     del <- rnorm(1, 0, sqrt(sgd))
50
51     # generate eta according to the SE
52     eta <- ubeta * d + gam[1] * xi[1] + gam[2] * xi[2] + del
53
54     # generate error term in ME
55     eps <- numeric(7)
56     for (k in 1:7) { eps[k] <- rnorm(1, 0, sgm[k]) }
57
58     # generate theta in ME
59     v[1] <- uby[1] + eta + eps[1]
60     v[2] <- uby[2] + lam[1] * eta + eps[2]
61     v[3] <- uby[3] + lam[2] * eta + eps[3]
62     Y[i, 4] <- uby[4] + xi[1] + eps[4]
63     Y[i, 5] <- uby[5] + lam[3] * xi[1] + eps[5]
64     Y[i, 6] <- uby[4] + lam[4] * xi[1] + eps[6]
65     Y[i, 7] <- uby[4] + lam[5] * xi[1] + eps[7]
66     v[8] <- uby[4] + xi[1]
```

```

67 v[9] <- uby[4] + lam[6] * xi[1]
68 v[10] <- uby[4] + lam[7] * xi[1]
69
70 # transform theta to ordinal variables
71 for (j in 1:3) {
72   if (v[j] > 0) Y[i, j] <- 1
73   else Y[i, j] <- 0
74 }
75
76 # transform theta to binary variables
77 for (j in 8:10) {
78   p[j] <- exp(v[j]) / (1 + exp(v[j]))
79   Y[i, j] <- rbinom(1, 1, p[j])
80 }
81 }
82
83 # save data matrix
84 write.table(Y, paste(datapath, "/Y-", t, ".txt", sep = ""))
85 write.table(D, paste(datapath, "/D-", t, ".txt", sep = ""))
86 }
87
88 true_params = list(
89   lam = lam,
90   uby = uby,
91   sgm = sgm,
92   ubeta = ubeta,
93   gam = gam,
94   phx = phx,
95   sgd = sgd
96 )
97
98 save(true_params, file = paste0(datapath, "/trueparams.RData"))

```

codes/generateData.R

1.2 True model in BUGS language

```

1 model{
2   for (i in 1:N) {
3     # measurement equation model
4     for (j in 1:3) {
5       y[i, j] ~ dnorm(mu[i, j], 1)I(low[z[i, j] + 1], high[z[i, j] + 1])
6     }
7     ## winbugs cannot handle operation in index variables, like k = j - 2 !!!
8     y[i, 4] ~ dnorm(mu[i, 4], psi[1])
9     y[i, 5] ~ dnorm(mu[i, 5], psi[2])
10    y[i, 6] ~ dnorm(mu[i, 6], psi[3])
11    y[i, 7] ~ dnorm(mu[i, 7], psi[4])
12    for (j in 8:P) {
13      z[i, j] ~ dbin(pb[i, j], 1)
14      pb[i, j] <- exp(mu[i, j]) / (1 + exp(mu[i, j]))
15    }
16
17    mu[i, 1] <- uby[1] + eta[i]
18    mu[i, 2] <- uby[2] + lam[1] * eta[i]
19    mu[i, 3] <- uby[3] + lam[2] * eta[i]
20    mu[i, 4] <- uby[4] + xi[i, 1]
21    mu[i, 5] <- uby[5] + lam[3] * xi[i, 1]
22    mu[i, 6] <- uby[6] + lam[4] * xi[i, 1]
23    mu[i, 7] <- uby[7] + lam[5] * xi[i, 1]
24    mu[i, 8] <- uby[8] + xi[i, 2]
25    mu[i, 9] <- uby[9] + lam[6] * xi[i, 2]
26    mu[i, 10] <- uby[10] + lam[7] * xi[i, 2]
27
28
29    # structural equation model
30    xi[i, 1:2] ~ dmnorm(zero2[1:2], phi[1:2, 1:2])
31    eta[i] ~ dnorm(etamu[i], psd)
32

```

```

33   etamu[i] <- ubeta * d[i] + gam[1] * xi[i, 1] + gam[2] * xi[i, 2]
34 } # End for i
35
36 for (k in 1:2) { zero2[k] <- 0 }
37
38
39 # priors inputs for loadings and coefficients
40 for (j in 1:P) { uby[j] ~ dnorm(0.0, 4.0) }
41
42 for (j in 1:2) { lam[j] ~ dnorm(0.5, 4.0) }
43
44 # pslam <- 4.0 * psi
45 lam[3] ~ dnorm(0.5, psi[2])
46 lam[4] ~ dnorm(0.5, psi[3])
47 lam[5] ~ dnorm(0.5, psi[4])
48
49 for (j in 6:7) { lam[j] ~ dnorm(0.5, 4.0) }
50
51 ubeta ~ dnorm (0.5, psd)
52
53 psgam <- 4.0 * psd
54 for (k in 1:2) {
55   gam[k] ~ dnorm(0.5, psgam)
56 }
57
58
59 # priors inputs for precisions
60 for (j in 1:4) {
61   psi[j] ~ dgamma(9, 3)
62   sgm[j] <- 1 / psi[j]
63 }
64
65 psd ~ dgamma(9, 3)
66 sgd <- 1 / psd
67
68 phi[1:2, 1:2] ~ dwish(R[1:2, 1:2], 10)
69 phx[1:2, 1:2] <- inverse(phi[1:2, 1:2])
70 } # End of model

```

codes/model1.txt

1.3 Simulation process

```

1 library(R2WinBUGS)
2
3 # set experiment date
4 timestamp = strftime(Sys.time(), "%Y%m%d-%H")
5 winBUGS.path = "D:/pkgs/WinBUGS14/"
6 datapath = paste0(getwd(), '/data')
7
8
9 # data size
10 iter = 10
11 N = 500
12 P = 10
13 Nlam = 7
14 Neta = 1
15 Nxi = 2
16 Ngam = 2
17
18 # containers for Bayesian estimates and standard errors
19 uby.E <- matrix(data = NA, nrow = iter, ncol = P)
20 uby.SE <- matrix(data = NA, nrow = iter, ncol = P)
21 lam.E <- matrix(data = NA, nrow = iter, ncol = Nlam)
22 lam.SE <- matrix(data = NA, nrow = iter, ncol = Nlam)
23 gam.E <- matrix(data = NA, nrow = iter, ncol = Ngam)
24 gam.SE <- matrix(data = NA, nrow = iter, ncol = Ngam)
25 phx.E <- matrix(data = NA, nrow = iter, ncol = Nxi^2)
26 phx.SE <- matrix(data = NA, nrow = iter, ncol = Nxi^2)

```

```

27 ubeta.E <- numeric(iter)
28 ubeta.SE <- numeric(iter)
29 # only continuous part of y's variances need to estimate
30 sgm.E <- matrix(data = NA, nrow = iter, ncol = 4)
31 sgm.SE <- matrix(data = NA, nrow = iter, ncol = 4)
32 sgd.E <- numeric(iter)
33 sgd.SE <- numeric(iter)
34
35 # containers for HPD (Highest Probability Density) intervals
36 uby.hpd <- array(NA, c(iter, P, 2))
37 ubeta.hpd <- array(NA, c(iter, 2))
38 lam.hpd <- array(NA, c(iter, Nlam, 2))
39 gam.hpd <- array(NA, c(iter, Ngam, 2))
40 phx.hpd <- array(NA, c(iter, Nxi^2, 2))
41 sgm.hpd <- array(NA, c(iter, 4, 2))
42 sgd.hpd <- array(NA, c(iter, 2))
43
44 # container for DIC values
45 DIC = numeric(iter)
46
47 # parameters to be estimated
48 parameters <- c("uby", "ubeta", "lam", "gam", "phx", "sgm", "sgd")
49
50 # set important prior param R_0
51 R0 = matrix(c(1, 0.5, 0.5, 1), nrow = 2)
52
53 # initial values for MCMC in WinBUGS
54 init1 <- list(
55   uby = rep(0.5, P),
56   ubeta = 0.5,
57   lam = rep(0.5, Nlam),
58   gam = rep(0.5, Ngam),
59   phi = matrix(c(1, 0.5, 0.5, 1), nrow = 2),
60   psi = rep(1, 4),
61   psd = 1
62 )
63
64 init2 <- list(
65   uby = rep(0, P),
66   ubeta = 0,
67   lam = rep(0, Nlam),
68   gam = rep(0, Ngam),
69   phi = matrix(c(2, 0, 0, 2), nrow = 2),
70   psi = rep(2, 4),
71   psd = 2
72 )
73
74 init3 <- list(
75   uby = rep(0, P),
76   ubeta = 0.3,
77   lam = rep(0.8, Nlam),
78   gam = rep(0.45, Ngam),
79   phi = matrix(c(1, 0.2, 0.2, 1), nrow = 2),
80   psi = rep(0.3, 4),
81   psd = 0.4
82 )
83
84 inits <- list(init3)
85
86
87 # Do simulations based on 10 replications
88 for (t in 1:iter) {
89   iterpath = paste0(getwd(), "/Q1.1_", t)
90   dir.create(iterpath, showWarnings = FALSE, recursive = TRUE)
91
92   Y <- as.matrix(read.table(paste(datapath, "/Y-", t, ".txt", sep = "")))
93   D <- read.table(paste(datapath, "/D-", t, ".txt", sep = ""))$x
94
95   data <- list(N = N, P = P, R = R0,
96               d = D, z = Y,
97               low = c(-2000, 0),

```



```

98         high = c(0, 2000))
99
100 model = bugs(data, inits, parameters,
101             model.file = paste0(getwd(), "../model1.txt"),
102             n.chains = 1,
103             n.iter = 3000,
104             n.burnin = 1500,
105             n.thin = 1,
106             DIC = TRUE,
107             bugs.directory = winBUGS.path,
108             working.directory = iterpath,
109             debug = FALSE)
110
111 # save estimates and standard errors
112 uby.E[t, ] = model$mean$uby
113 uby.SE[t, ] = model$sd$uby
114 ubeta.E[t] = model$mean$ubeta
115 ubeta.SE[t] = model$sd$ubeta
116 lam.E[t, ] = model$mean$lam
117 lam.SE[t, ] = model$sd$lam
118 gam.E[t, ] = model$mean$gam
119 gam.SE[t, ] = model$sd$gam
120 phx.E[t, ] = c(model$mean$phx)
121 phx.SE[t, ] = c(model$sd$phx)
122 sgm.E[t, ] = model$mean$sgm
123 sgm.SE[t, ] = model$sd$sgm
124 sgd.E[t] = model$mean$sgd
125 sgd.SE[t] = model$sd$sgd
126
127 # save HPD intervals
128 for (k in 1:P) {
129     temp = model$sims.array[ , 1, k]
130     uby.hpd[t, k, ] = boa.hpd(temp, 0.05)
131 }
132 temp = model$sims.array[ , 1, P + 1]
133 ubeta.hpd[t, ] = boa.hpd(temp, 0.05)
134 for (k in 1:Nlam) {
135     temp = model$sims.array[ , 1, P + 1 + k]
136     lam.hpd[t, k, ] = boa.hpd(temp, 0.05)
137 }
138 for (k in 1:Ngam) {
139     temp = model$sims.array[ , 1, P + 1 + Nlam + k]
140     gam.hpd[t, k, ] = boa.hpd(temp, 0.05)
141 }
142 for (k in 1:Nxi^2) {
143     temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + k]
144     phx.hpd[t, k, ] = boa.hpd(temp, 0.05)
145 }
146 for (k in 1:4) {
147     temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + Nxi^2 + k]
148     sgm.hpd[t, k, ] = boa.hpd(temp, 0.05)
149 }
150 temp = model$sims.array[ , 1, P + 1 + Nlam + Ngam + Nxi^2 + 4 + 1]
151 sgd.hpd[t, ] = boa.hpd(temp, 0.05)
152
153 # save DIC values
154 DIC[t] = model$DIC
155
156 summary(model)
157 }
158
159 metr_params = list(
160     E = list(
161         uby = uby.E, ubeta = ubeta.E, lam = lam.E, gam = gam.E,
162         phx = phx.E, sgm = sgm.E, sgd = sgd.E
163     ),
164     SE = list(
165         uby = uby.SE, ubeta = ubeta.SE, lam = lam.SE, gam = gam.SE,
166         phx = phx.SE, sgm = sgm.SE, sgd = sgd.SE
167     ),
168     HPD = list(

```

```
169     uby = uby.hpd, ubeta = ubeta.hpd, lam = lam.hpd, gam = gam.hpd,  
170     phx = phx.hpd, sgm = sgm.hpd, sgd = sgd.hpd  
171 )  
172 )  
173  
174 save(metr_params, file = paste0(getwd(), "/Q1.1_metrparams.RData"))
```

codes/simulation1.R