require(rstan)

require(rjags)

require(loo)

setwd("C:/R files BHMRA")

attach("DS\_9\_1.Rdata")

#

# EXACT CONFIRMATORY

#

cfaexact.stan <- "

data {

// n: number of subjects

int n ;

// p: number of outcomes

int p ;

// number of subjects\*outcomes

int np ;

// y: matrix of outcomes

matrix[n,p] y ;

// q: number of latent factors

int q ;

// assoc: specify factor associated with outcome

int<lower=1,upper=q> assoc[p] ;}

parameters{

matrix[q, n] Z;

// Cholesky factor of correlation matrix

cholesky\_factor\_corr[q] F\_cor\_chol ;

vector[p] alpha ;

vector<lower=0>[p] sd\_y ;

vector<lower=0>[p] lambda ;}

transformed parameters{

matrix[n,q] F ;

// factor scores

F = transpose(F\_cor\_chol \* Z) ;}

model { to\_vector(Z) ~ normal(0,1) ;

F\_cor\_chol ~ lkj\_corr\_cholesky(1) ;

alpha ~ normal(0,1) ;

sd\_y ~ uniform(0,10) ;

lambda ~ normal(0,1) ;

for(i in 1:p){y[,i] ~ normal(alpha[i] + F[,assoc[i]] \* lambda[i], sd\_y[i]) ;}}

generated quantities{

vector[np] loglik;

corr\_matrix[q] F\_cor ;

F\_cor = multiply\_lower\_tri\_self\_transpose(F\_cor\_chol) ;

for (i in 1:n){for (j in 1:p){

loglik[i+(j-1)\*n] = normal\_lpdf(y[i,j]|alpha[j]+F[i,assoc[j]] \* lambda[j], sd\_y[j]);}}}"

# Estimation

sm <- stan\_model(model\_code=cfaexact.stan)

post <- sampling(sm, data =DS\_9\_1, iter = 2000,warmup=500,chains = 2,seed=1234)

# lambda loadings

summary(post, pars = c("lambda"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# factor correlations

summary(post, pars = c("F\_cor"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# WAIC

waic1=waic(as.matrix(post,pars="loglik"))

#

# CFA with INFORMATIVE PRIORS ON CROSS LOADINGS

#

bsem.stan <- "data {// n: number of subjects

int n ;

// p: number of outcomes

int p ;

// number of subjects\*outcomes

int np ;

// y: matrix of outcomes

matrix[n,p] y ;

// q: number of latent factors

int q ;

// assoc: which factor associated with which outcome

int<lower=1,upper=q> assoc[p] ;

// cross: which factor not associated with which outcome

int<lower=1,upper=q> cross[p] ;}

parameters{

matrix[q, n] Z;

// Cholesky factor of correlation matrix

cholesky\_factor\_corr[q] F\_cor\_chol ;

vector[p] alpha ;

vector<lower=0>[p] sd\_y ;

vector<lower=0>[p] lambda\_assoc ;

vector<lower=0>[p] lambda\_cross ;}

transformed parameters{

matrix[n,q] F ;

// factor scores

F = transpose(F\_cor\_chol \* Z) ;}

model { to\_vector(Z) ~ normal(0,1) ;

F\_cor\_chol ~ lkj\_corr\_cholesky(1) ;

alpha ~ normal(0,1) ;

sd\_y ~ uniform(0,10) ;

lambda\_assoc ~ normal(0,1) ;

lambda\_cross ~ normal(0,0.1) ;

for(i in 1:p){y[,i] ~ normal(alpha[i] + F[,assoc[i]] \* lambda\_assoc[i]

+ F[,cross[i]] \* lambda\_cross[i], sd\_y[i]) ; }}

generated quantities{

vector[np] loglik;

corr\_matrix[q] F\_cor ;

F\_cor = multiply\_lower\_tri\_self\_transpose(F\_cor\_chol) ;

for(i in 1:n) {for (j in 1:p){

loglik[i+(j-1)\*n] = normal\_lpdf(y[i,j]|alpha[j]+

F[i,assoc[j]] \* lambda\_assoc[j] + F[i,cross[j]] \* lambda\_cross[j], sd\_y[j]);}}}"

sm <- stan\_model(model\_code=bsem.stan)

post <- sampling(sm, data =DS\_9\_1, iter = 2000,warmup=500,chains = 2,seed=1234)

# main CFA loadings

summary(post, pars = c("lambda\_assoc"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# cross CFA loadings

summary(post, pars = c("lambda\_cross"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# factor correlations

summary(post, pars = c("F\_cor"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# WAIC

waic2=waic(as.matrix(post,pars="loglik"))

#

# SELECT ITEMS and ITEM-FACTOR COMBINATIONS

#

model= "model {for (j in 1:p) {s[j]~dunif(0,100)

a[j] ~ dnorm(0,0.001)}

# priors for loadings on factor 1

# priors on main loadings

for (j in 1:6){lam1[j] ~ dnorm(0,1) T(0,)}

# informative priors for cross loadings concentrated at zero

for (j in 7:p){lam1[j] ~ dnorm(0,100)}

# priors for loadings on factor 2

# informative priors for cross loadings concentrated at zero

for (j in 1:6){lam2[j] ~ dnorm(0,100)}

# priors on main loadings

for (j in 7:p) {lam2[j] ~ dnorm(0,1) T(0,)}

# realised loadings

for (j in 1:p){lam1.r[j] <- lam1[j]\*gamma[j,1]

lam2.r[j] <- lam2[j]\*gamma[j,2] }

# selection indicators

for (j in 1:p) {pi.gamma[j] ~ dbeta(1,1)}

for (k in 1:q) {for (j in 1:p) { gamma[j,k] ~ dbern(pi.gamma[j])}}

# prior for correlated factors

for (j in 1:q) { B[j] <- 0

for (k in 1:q) {Q[j,k] <- equals(j,k)}}

inv.Phi[1:2,1:2] ~ dwish(Q[,],2)

Phi[1:2,1:2] <- inverse(inv.Phi[,])

cor.F <- Phi[1,2]/sqrt(Phi[1,1]\*Phi[2,2])

for (i in 1:n) { F[i,1:2] ~ dmnorm(B[],inv.Phi[,])

# Likelihood

for (j in 1:p){ y[i,j] ~ dnorm(mu[i,j], 1/(s[j]\*s[j]));

mu[i,j] <- a[j]+gamma[j,1]\*lam1[j]\*F[i,1] + gamma[j,2]\*lam2[j]\*F[i,2]}}} "

# Initial Values and Estimation

F0 <- matrix(0,175,2)

inits1 <- list(s=rep(1,11),lam1=rep(1,11), lam2=rep(1,11),a=rep(11,0),F=F0)

inits2 <- list(s=rep(2,11),lam1=rep(0.8,11), lam2=rep(0.8,11),a=rep(11,0),F=F0)

inits=list(inits1,inits2)

mod=jags.model(textConnection(model),data=DS\_9\_1,n.chains=2,inits=inits)

update(mod,500)

pars=c("a","lam1.r","lam2.r","gamma","pi.gamma","cor.F")

mod\_sim=coda.samples(model=mod,variable.names=pars,n.iter=10000,seed=1234)

summary(mod\_sim)