require(jagsUI)

library(lavaan)

library(rube)

library(mvtnorm)

library(loo)

Sys.setenv(BUGSDIR="c:\\users\\p congdon\\documents\\WINBUGS14")

**#**

**# Simulate Data for Exploratory Factor Analysis**

**#**

set.seed(1.234)

n = 200 # number of observations

p = 6 # number of indicators

q = 2 # number of factors

# Loading matrix

Lambda = matrix(c(1.1,-0.1,

-0.2,1.2,

0.8,0,

-0.9,0.6,

0.3,0.7,

-0.8,0.9),

nrow=p, ncol=q, byrow=T)

DF=5

S=matrix(c(1,0,0,1), nrow=q, ncol=q, byrow=T)

mean=c(0, 0)

F = rmvt(n, sigma=S\*(DF-2)/DF, df=DF) + mean # MVT Factor Scores

e <- rmvnorm(n, rep(0, p), diag(p)) # N(0,1) errors

y <- F %\*% t(Lambda) + e # indicator matrix

y <- scale(y)

**# max lkd EFA (lavaan)**

factanal(~y,factors=2)

Phi.Sc=diag(q)

zero=rep(0,q)

D = list(y=y,Phi.Sc=Phi.Sc,p=p,n=n,zero=zero)

**#**

**# MVN Factors**

**#**

cat("model{ for (i in 1:n){ Fstar[i,1:2]~dmnorm(zero[1:2],Phi.inv[1:2,1:2])

for (j in 1:p) {y[i,j]~dnorm(Fstar[i,1]\*lambda1star[j] + Fstar[i,2]\*lambda2star[j], tau[j])

LL[i,j] <- -0.92+0.5\*log(tau[j])-0.5\*tau[j]\*(y[i,j]-

Fstar[i,1]\*lambda1star[j] - Fstar[i,2]\*lambda2star[j])}}

# prior for non-identified loadings

for (j in 1:p){ lambda1star[j]~dnorm(0,omega[1]) }

lambda2star[1] ~ dnorm(0,100)

for (j in 2:p){ lambda2star[j] ~dnorm(0,omega[2]) }

# other priors

for (j in 1:2) {omega[j] ~ dexp(1)}

for (j in 1:p){ tau[j]~dgamma(0.1,0.1) }

# Covariance between factors

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

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

# transform to identified parameters

for (j in 1:p){ lambda1[j]<-ifelse(lambda1star[1]<0,-1,1)\*lambda1star[j]\*sqrt(CovF[1,1])

lambda2[j]<-ifelse(lambda2star[2]<0,-1,1)\* lambda2star[j]\*sqrt(CovF[2,2]) }

phi <-ifelse(lambda1star[1]\*lambda2star[2]<0,-1,1)\* CovF[1,2]/sqrt(CovF[1,1]\*CovF[2,2])}

", file="model1.jag")

**# Initial Values and Estimation**

F0 = matrix(0,n,2)

inits1 = list(lambda1star=rep(0,p), lambda2star=rep(0,p),tau=rep(0.2,p),Fstar=F0)

inits2 = list(lambda1star=rep(0.1,p), lambda2star=rep(0.1,p),tau=rep(0.4,p),Fstar=F0)

inits=list(inits1,inits2)

pars=c("lambda1","lambda2","phi","tau","LL")

R1 = autojags(D, inits, pars,model.file="model1.jag",2,iter.increment=5000, n.burnin=500,Rhat.limit=1.1, max.iter=100000, seed=1234,codaOnly=c("LL"))

R1$summary

**# Overall Fit**

LOO1=loo(matrix(as.array(R1$sims.list$LL),10000,6\*200))

**# Fit for individual observations**

loocase= LOO1$pointwise[,3]

subject = rep(seq(1:200),6)

indicator = c(rep(1,200),rep(2,200),rep(3,200),rep(4,200),rep(5,200),rep(6,200))

# extreme individual loo

list.loo = data.frame(loocase,subject,indicator)

list.loo=list.loo[order(-list.loo$loocase),]

head(list.loo,10)

**#**

**# MVT Factors**

**#**

model2= " model { for (i in 1:n){ Fstar[i,1:2]~dmnorm(zero[1:2],Phi[i,1:2,1:2])

# Student t scale mixture terms

scale[i] ~ dgamma(nu.2,nu.2)

for (j in 1:2) { for (k in 1:2) { Phi[i,j,k] <- Phi.inv[j,k]\*scale[i]}}

for (j in 1:p) {y[i,j]~dnorm(mu[i,j], tau[j])

mu[i,j] <- Fstar[i,1]\*lambda1star[j] + Fstar[i,2]\*lambda2star[j]

LL[i,j] <- -0.92+0.5\*log(tau[j])-0.5\*tau[j]\*(y[i,j]-

Fstar[i,1]\*lambda1star[j] - Fstar[i,2]\*lambda2star[j])}}

# prior for non-identified loadings

for (j in 1:p){ lambda1star[j]~dnorm(0,omega[1]) }

lambda2star[1] ~ dnorm(0,100)

for (j in 2:p){ lambda2star[j] ~dnorm(0,omega[2]) }

# Prior on student df

nu ~ dgamma(2,0.1)

nu.2 <- nu/2

# other priors

for (j in 1:2) {omega[j] ~ dexp(1)}

for (j in 1:p){ tau[j]~dgamma(0.1,0.1) }

# Covariance between factors

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

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

# transform to identified parameters

ifelse1 <- -1\*step(-lambda1star[1])+step(lambda1star[1])

ifelse2 <- -1\*step(-lambda2star[2])+step(lambda2star[2])

ifelse3 <- -1\*step(-(lambda1star[1]\*lambda2star[2]))+step(lambda1star[1]\*lambda2star[2])

for (j in 1:p){ lambda1[j]<- ifelse1\*lambda1star[j]\*sqrt(CovF[1,1])

lambda2[j]<- ifelse2\* lambda2star[j]\*sqrt(CovF[2,2]) }

phi <-ifelse3 \* CovF[1,2]/sqrt(CovF[1,1]\*CovF[2,2])}

"

**# Initial Values and Estimation**

F0 = matrix(0,n,2)

inits1 = list(lambda1star=rep(0,p), lambda2star=rep(0,p),tau=rep(0.2,p),Fstar=F0)

inits2 = list(lambda1star=rep(0.1,p), lambda2star=rep(0.1,p),tau=rep(0.4,p),Fstar=F0)

inits=list(inits1,inits2)

pars=c("lambda1","lambda2","phi","tau","nu","scale","LL")

syntaxcheck = rube(model2, D, inits)

summary(syntaxcheck)

R2 = rube(model2, D, inits, pars, n.burn=2500, n.thin=1, n.chains=2,n.iter=10000)

summary(R2,limit=20)

**# Fit**

LOO2=loo(matrix(as.array(R2$sims.list$LL),15000,6\*200))

**# scale mixture samples**

MVTscale= R2$sims.list$scale

scale.mn=apply(MVTscale,2,mean)

scale.05=apply(MVTscale,2,quantile,0.05)

scale.95=apply(MVTscale,2,quantile,0.95)

sum(scale.mn < 0.8)

sum(ifelse(scale.95<1,1,0))

**# lowest scale multipliers**

obs=rep(1:200)

list <- data.frame(scale.mn,scale.05,scale.95,obs)

tail(list[order(-list$scale.mn),],10)