library(R2OpenBUGS)

library(coda)

library(MCMCvis)

library(loo)

library(rstan)

setwd("C:/R files BHMRA")

attach("DS\_9\_11.Rdata")

attach(DS\_9\_11)

#

# Trends and Cycles

#

model1 <- function() { for ( t in 1:T){ for (p in 1:P) {

beta.var[p,t] <- beta[t,p]

# trends by outcome

theta.var[p,t] <- theta[t,p]

psi.var[p,t] <- psi1[t,p]

u[t,p] <- y[t,p]-nu[t,p]

ePrec[t,p] <- inprod(u[t,],Precy[,p])}

# log-likelihood

LL[t] <- -P\*log(2\*pi)/2 + logdet(Precy[,])/2 - inprod(ePrec[t,],u[t,])/2

y[t,1:P] ~ dmnorm(nu[t,1:P],Precy[,])

for (p in 1:P) {# trend plus cycle

nu[t,p] <- theta[t,p]+psi1[t,p]}}

# model for t>1

for (t in 2:T) {theta[t,1:P] ~ dmnorm(M.theta[t,1:P],Prectheta[,])

beta[t,1:P] ~ dmnorm(beta[t-1,1:P],Precbeta[,])

psi1[t,1:P] ~ dmnorm(M.psi1[t,1:P],Precps1[,])

psi2[t,1:P] ~ dmnorm(M.psi2[t,1:P],Precps2[,])

for (p in 1:P) {M.theta[t,p] <- theta[t-1,p]+beta[t,p]

M.psi1[t,p] <- rho\*cos(lambda)\*psi1[t-1,p]+rho\*sin(lambda)\*psi2[t-1,p]

M.psi2[t,p] <- -rho\*sin(lambda)\*psi1[t-1,p]+rho\*cos(lambda)\*psi2[t-1,p]}}

# Initial Conditions (priors on parameters for t=1)

for (p in 1:P){ beta1[p] ~ dnorm(0,0.01)

theta1[p] ~ dnorm(0,0.01)

psi11[p] ~ dnorm(0,0.01);

psi21[p] ~ dnorm(0,0.01)

psi1[1,p] <- psi11[p];

psi2[1,p] <- psi21[p]

beta[1,p] <- beta1[p];

theta[1,p] <- theta1[p]}

# Other priors

rho ~ dunif(0,1)

lambda ~ dunif(0.3,1.5);

Period <- 2\*pi/lambda

for (a in 1:4) {q[a] ~ dgamma(1,1)}

# prior precision matrices

Precy[1:P,1:P] ~ dwish(Scy[1:P,1:P],nu.y);

for (p1 in 1:P){ for (p2 in 1:P) {

Scy[p1,p2] <- Sigy[p1,p2]\*nu.y

Prectheta[p1,p2] <- Precy[p1,p2]/q[1]

Precbeta[p1,p2] <- Precy[p1,p2]/q[2]

Precps1[p1,p2] <- Precy[p1,p2]/q[3]

Precps2[p1,p2] <- Precy[p1,p2]/q[4]}}}

**# Initial Values and Estimation**

init1 = list(Precy=diag(1,2),q=rep(0.2,4),rho=0.5,lambda=1,beta1 = c(0,0),psi11 = c(0,0),psi21 = c(0,0),beta = matrix(0,64,2),theta=matrix(0,64,2),theta1 = c(10,12))

init2 = list(Precy=diag(1.5,2),q=rep(0.1,4),rho=0.75,lambda=0.75,beta1 = c(0,0),psi11 = c(0,0),psi21 = c(0,0),beta = matrix(0,64,2),theta=matrix(0,64,2),theta1 = c(10,12))

inits=list(init1,init2)

pars=c("q","Period","lambda","theta.var","LL")

R1= bugs(DS\_9\_11, inits, pars,100000,model1,n.chains = 2,n.burnin =25000,codaPkg=F,

debug=T,bugs.seed=1)

R1$summary

**# Fit**

LOO1=loo(R1$sims.list$LL)

loo.pw=LOO1$pointwise[,3]

years=1847+seq(1:DS\_9\_11$T)

list = data.frame(loo.pw,years)

head(list[order(-list$loo.pw),],10)

#

**# VAR(1) Model**

#

model2 <- function() { # P=no of joint dep vars, R= lag

for (p in 1:P) { gamma[p] ~ dnorm(0,0.001)

# prior on own first lag

for (lag in 1:R) {for (q in 1:P) {alpha[p,q,lag] ~dnorm(0,0.001)}}}

# prior precision

invV[1:P,1:P] ~ dwish(Sc[1:P,1:P],P);

for (p1 in 1:P){ for (p2 in 1:P) {Sc[p1,p2] <- equals(p1,p2)}}

# effect of lagged variables

for (t in R+1:T) {for (p1 in 1:P) { for (p2 in 1:P) {

for (lag in 1:R) { Elag[t,p1,p2,lag] <- alpha[p1,p2,lag]\*y[t-lag,p2]}}}

for (p in 1:P){ theta[t,p] <- gamma[p]+ sum(Elag[t,p,1:P,1:R])

u[t,p] <- y[t,p] - theta[t,p]

L1[p,t] <- inprod(invV[p,], u[t,]) }

sumsq[t] <- inprod(u[t,], L1[,t])

# log-likelihood

LL[t] <- -P\*log(2\*pi)/2 -0.5\*logdet(Sigma[,]) -0.5\* sumsq[t]

y[t,1:P] ~ dmnorm(theta[t,1:P], invV[1:P,1:P])}

# error covariance

Sigma[1:P,1:P] <- inverse(invV[,]);

for (i in 1:P) {for (j in 1:P) { u.corr[i,j] <- Sigma[i,j]/sqrt(Sigma[i,i]\*Sigma[j,j])}}}

**# Initial Values and Estimation**

init1=list(alpha = array(0,dim=c(2,2,1)),gamma = c(0,0),invV =diag(5,2))

init2=list(alpha = array(0,dim=c(2,2,1)),gamma = c(0,0),invV =diag(5,2))

inits=list(init1,init2)

pars=c("alpha","gamma","LL","u.corr","Sigma")

R2= bugs(DS\_9\_11, inits, pars,25000,model2,n.chains = 2,n.burnin =1000, codaPkg=F,

debug=T, bugs.seed=1)

R2$summary

**# Fit**

LOO2=loo(R2$sims.list$LL)

**# MLE of VAR(1)**

z=as.matrix(DS\_9\_11$y)

require(vars)

**# Max Lkd Est**

VAR(z,p=1)

#

**# rstan code, VAR(1)**

#

data <- list(T = 64,P = 2, R = 1,Y = z)

VAR.stan <- "

data { int<lower=0> T; // No. of time periods

int<lower=0> P; // No. of variables

int<lower=1> R; // Lag order

vector[P] Y[T];

}

transformed data {vector[P] Y\_obs[T-R];

for (t in 1:(T-R))

Y\_obs[t] = Y[t + R];

}

parameters {matrix[P, P] alpha[R];

cholesky\_factor\_corr[P] L\_corr\_noise;

vector<lower=0>[P] sd\_noise;

vector[P] gamma;

}

transformed parameters {matrix[P,P] L\_sigma;

L\_sigma = diag\_pre\_multiply(sd\_noise, L\_corr\_noise);

}

model { vector[P] mu[T-R];

for (t in 1:(T-R)) {

mu[t] = gamma;

for (p in 1:R)

mu[t] = mu[t] + alpha[p] \* Y[t+p-1];

}

L\_corr\_noise ~ lkj\_corr\_cholesky(2.0);

sd\_noise ~ normal(0,5);

gamma ~ normal(0,10);

for (p in 1:R)

to\_vector(alpha[p]) ~ normal(0, 10);

Y\_obs ~ multi\_normal\_cholesky(mu,L\_sigma);

}

generated quantities { matrix[P,P] Sigma;

Sigma = L\_sigma \* L\_sigma';

}

"

sm = stan\_model(model\_code=VAR.stan)

fit = sampling(sm,data =data, pars = c("gamma","alpha","Sigma"), iter = 1000,chains = 2)

print(fit,digits=3)