require(loo)

library(R2OpenBUGS)

library(MCMCvis)

library(rstan)

data(faithful)

y = ifelse(faithful[,1] > 3,1,0)

D=list(T=272,y=y)

**#**

**# Parameter Driven, Random AR1 Effect**

**#**

model1 <- function() { for (t in 1:T) {y[t] ~ dbern(p[t])

beta2[t] <- beta[2]+g[t]

LL[t] <- y[t]\*log(p[t])+(1-y[t])\*log(1-p[t])}

for (t in 2:T) {logit(p[t]) <- beta[1]+(beta[2]+g[t])\*y[t-1]}

logit(p[1]) <- beta[1]+(beta[2]+g[1])\*y0

y0 ~ dbern(p0)

p0 ~ dbeta(1,1)

for (j in 1:2) {beta[j] ~ dnorm(0,0.01)}

**# Prior on RW1 random effects**

**# boundary points**

w[1] <- 1; adj[1] <- 2; n[1] <- 1

w[(T-2)\*2 + 2] <- 1; adj[(T-2)\*2 + 2] <- T-1; n[T] <- 1

**# interior points**

for (t in 2:T-1) {w[2+(t-2)\*2] <- 1; adj[2+(t-2)\*2] <- t-1

w[3+(t-2)\*2] <- 1; adj[3+(t-2)\*2] <- t+1;

n[t] <- 2}

g[1:T] ~ car.normal(adj[], w[], n[], tau.g)

tau.g ~ dgamma(1,0.01)}

**# Estimation**

init1= list(beta=c(3,-3),tau.g=1)

init2= list(beta=c(2.5,-2.5),tau.g=2)

inits=list(init1,init2)

pars <- c("tau.g","p","beta2","LL","beta","y0")

n.iters=50000; n.burnin =1000; n.chains=2

R1 = bugs(D,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F,bugs.seed=10)

R1$summary

**# Fit**

loo(R1$sims.list$LL)

beta.samps= R1$sims.list$beta2

mn.beta=apply(beta.samps,2,mean)

x=seq(1:272)

plot(x,mn.beta,xlab = expression(paste("Observation")),

ylab = expression(paste("Posterior mean ", beta[2])))

#

# BARMA(5,0)

# AR up to lag 5, excluding MA terms

# Horseshoe Prior on Coefficients

#

D=list(T=272,y=y,p=6)

BARMA50.stan <- "

data {

int<lower=0> T;

int<lower=0> p;

int y[T];

}

parameters {

real rho[p];

vector<lower=0,upper=1>[p] kappa;

real<lower=0> tau2;

real eps[5];

}

transformed parameters {

real<lower=0, upper=1> pi[T];

real eta[T];

real phi[p];

real<lower=0> tau;

tau = sqrt(tau2);

for (j in 1:p) {phi[j] = 1-kappa[j];}

eta[1] = rho[1] +eps[1];

pi[1] = exp(eta[1]) / (1+exp(eta[1]));

eta[2] = rho[1] + rho[2]\*y[1]+eps[2];

pi[2] = exp(eta[2]) / (1+exp(eta[2]));

eta[3] = rho[1] + rho[2]\*y[2]+rho[3]\*y[1]+eps[3];

pi[3] = exp(eta[3]) / (1+exp(eta[3]));

eta[4] = rho[1] + rho[2]\*y[3]+rho[3]\*y[2]+rho[4]\*y[1]+ eps[4];

pi[4] = exp(eta[4]) / (1+exp(eta[4]));

eta[5]=rho[1]+rho[2]\*y[4]+rho[3]\*y[3]+rho[4]\*y[2]+rho[5]\*y[1] +eps[5];

pi[5] = exp(eta[5]) / (1+exp(eta[5]));

for (i in 6:T) { eta[i] = rho[1] + rho[2]\*y[i-1] + rho[3]\*y[i-2]

+ rho[4]\*y[i-3] + rho[5]\*y[i-4] + rho[6]\*y[i-5];

pi[i] = exp(eta[i]) / (1+exp(eta[i])); }

}

model {

real lambda[p];

kappa ~ beta(0.5,0.5);

tau2 ~ inv\_gamma(1, 0.001);

for (i in 1:p) { lambda[i] = sqrt(1/kappa[i]-1);

rho[i] ~ normal(0, lambda[i] \* tau);}

eps ~ normal(0,1);

y ~ bernoulli(pi);

}

generated quantities{

vector[T] log\_lik;

for (t in 1:T) {log\_lik[t] =binomial\_logit\_log( y[t], 1 , pi[t] );}

}

"

sm = stan\_model(model\_code=BARMA50.stan)

R2 =sampling(sm,data =D,iter = 2000,warmup=500,chains = 2,seed= 12345)

summary(R2, pars = c("rho","phi"), probs = c(0.025,0.05, 0.95, 0.975))$summary

loo(as.matrix(R2,pars="log\_lik"))

#

# BARMA(5,1)

# Including MA(1) Term

# Horseshoe Prior on Coefficients

#

D=list(T=272,y=y,p=7)

BARMA51.stan <- "

data {

int<lower=0> T;

int<lower=0> p;

int y[T];

}

parameters {

real beta[p];

vector<lower=0,upper=1>[p] kappa;

real<lower=0> tau2;

real eps[5];

}

transformed parameters {

real<lower=0, upper=1> pi[T];

real eta[T];

real<lower=0> tau;

tau = sqrt(tau2);

eta[1] = beta[1] +eps[1];

pi[1] = exp(eta[1]) / (1+exp(eta[1]));

eta[2] = beta[1] + beta[2]\*y[1]+eps[2]

+ beta[7]\*(y[1]-pi[1]);

pi[2] = exp(eta[2]) / (1+exp(eta[2]));

eta[3] = beta[1] + beta[2]\*y[2]+beta[3]\*y[1]+eps[3]

+ beta[7]\*(y[2]-pi[2]);

pi[3] = exp(eta[3]) / (1+exp(eta[3]));

eta[4] = beta[1] + beta[2]\*y[3]+beta[3]\*y[2]+beta[4]\*y[1]+ eps[4]

+beta[7]\*(y[3]-pi[3]);

pi[4] = exp(eta[4]) / (1+exp(eta[4]));

eta[5] = beta[1] + beta[2]\*y[4]+beta[3]\*y[3]+beta[4]\*y[2]+beta[5]\*y[1]

+eps[5]+beta[7]\*(y[4]-pi[4]);

pi[5] = exp(eta[5]) / (1+exp(eta[5]));

for (i in 6:T) { eta[i] = beta[1] + beta[2]\*y[i-1] + beta[3]\*y[i-2]

+ beta[4]\*y[i-3] + beta[5]\*y[i-4] + beta[6]\*y[i-5] + beta[7]\*(y[i-1]-pi[i-1]);

pi[i] = exp(eta[i]) / (1+exp(eta[i])); }

}

model {

real lambda[p];

kappa ~ beta(0.5,0.5);

tau2 ~ inv\_gamma(1, 0.001);

for (i in 1:p) { lambda[i] = sqrt(1/kappa[i]-1);

beta[i] ~ normal(0, lambda[i] \* tau);}

eps ~ normal(0,1);

y ~ bernoulli(pi);

}

generated quantities{

vector[T] log\_lik;

for (t in 1:T) {log\_lik[t] =binomial\_logit\_log( y[t], 1 , pi[t] );}

}

"

sm = stan\_model(model\_code=BARMA51.stan)

R3 = sampling(sm,data =D,iter = 2000,warmup=500,chains = 2,seed= 12345)

print(R3)

loo(as.matrix(R3,pars="log\_lik"))