library(loo)

library(R2OpenBUGS)

library(maptools)

library(spdep)

library(nimble)

library(coda)

library(rstan)

# shapefile London MSOAs

setwd("C:/R files BHMRA")

LDNmap <- readShapePoly("Example\_6\_4")

LDNnb <- poly2nb(LDNmap, queen=F)

**# Binary Adjacency Matrix**

W=nb2mat(LDNnb, style="B", zero.policy=T)

d=as.vector(rowSums(W))

N=length(d)

N1=N+1

NN=sum(d)

# cumulative index of locality sizes

C=c()

C[1]=0; for (j in 2:N1) {C[j]=C[j-1]+d[j-1]}

#

**# adjacency list from binary adjacency matrix**

**#**

adjlist = function(W,N){ adj=0

for(i in 1:N){ for(j in 1:N){ if(W[i,j]==1){adj = append(adj,j)}}}

adj = adj[-1]

return(adj)}

adj=adjlist(W,N)

**# Observed counts and expected events**

data=read.table("Example\_6\_4.txt",header=T)

y=data$Y

E=data$E

#

# Model 1, BYM Model, nimble

#

code <- nimbleCode({alpha ~ dflat()

tau.s ~ dgamma(1,0.1)

tau.u ~ dgamma(1,0.1)

for(k in 1:NN)

weights[k] <- 1

s[1:N] ~ dcar\_normal(adj[1:NN], weights[1:NN], d[1:N], tau.s, zero\_mean = 1)

var.s <- sd(s[])^2

sigma2.u <- 1/tau.u

# proportion of residual variation due to spatial dependence

r.spatial <- var.s/(var.s+sigma2.u)

# BYM prior

for(i in 1:N) {log(rho[i]) <- s[i]+u[i]

u[i] ~ dnorm(alpha,tau.u)

# rho are relative risks

nu[i] <- E[i]\*rho[i]

# log-likelihoods

LL[i] <- -nu[i]+y[i]\*log(nu[i])-lfactorial(y[i])

y[i] ~ dpois(nu[i])}})

params= c("rho","tau.s","tau.u","LL","r.spatial")

constants=list(N=N,NN=NN, adj=adj,d=d,E=E)

data=list(y=y)

inits <- list(alpha = 0, tau.s = 2, tau.u=0.5,s = rep(0, 983),u = rep(0, 983))

samples <- nimbleMCMC(code,constants = constants, data=data,inits = inits,

monitors = params,niter = 10000,nburnin = 500,nchains=2,samplesAsCodaMCMC = T)

summary(samples)

gelman.diag(samples, multivariate=F)

S=as.matrix(samples)

**# LOO-IC**

LOO1=loo(S[,1:983])

**# Poorly fitted observations**

loocase <- as.vector(LOO1$pointwise[,3])

**# percent of LOO-IC due to 5% worst fitted cases**

100\*sum(loocase[loocase>quantile(loocase,0.95)])/sum(loocase)

**# list individual LOO-IC**

area=seq(1:983)

list.loocase <- data.frame(area,loocase,y,E)

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

head(list.loocase,10)

**# posterior mean relative risks**

RR.BYM=apply(S[,984:(2\*983)],2,mean)

range(RR.BYM)

**# Model 2**

**#**

**# Proper CAR**

**#**

D2 = list(n = N, # number of observations

y = y, # observed number of cases

log\_offset = log(E), # log(expected) num. cases

W\_n = sum(W) / 2, # number of neighbor pairs

W = W)

propercar.stan ="

functions {

real sparse\_car\_lpdf(vector phi, real tau, real alpha,

int[,] W\_sparse, vector D\_sparse, vector lambda, int n, int W\_n) {

row\_vector[n] phit\_D; // phi' \* D

row\_vector[n] phit\_W; // phi' \* W

vector[n] ldet\_terms;

phit\_D = (phi .\* D\_sparse)';

phit\_W = rep\_row\_vector(0, n);

for (i in 1:W\_n) {

phit\_W[W\_sparse[i, 1]] = phit\_W[W\_sparse[i, 1]] + phi[W\_sparse[i, 2]];

phit\_W[W\_sparse[i, 2]] = phit\_W[W\_sparse[i, 2]] + phi[W\_sparse[i, 1]];

}

for (i in 1:n) ldet\_terms[i] = log1m(alpha \* lambda[i]);

return 0.5 \* (n \* log(tau)

+ sum(ldet\_terms)

- tau \* (phit\_D \* phi - alpha \* (phit\_W \* phi))); }}

data {

int<lower = 1> n;

int<lower = 0> y[n];

vector[n] log\_offset;

matrix<lower = 0, upper = 1>[n, n] W; // adjacency matrix

int W\_n; // number of adjacent region pairs

}

transformed data {

int W\_sparse[W\_n, 2]; // adjacency pairs

vector[n] D\_sparse; // diagonal of D (number of neigbors for each site)

vector[n] lambda; // eigenvalues of invsqrtD \* W \* invsqrtD

{ // generate sparse representation for W

int counter;

counter = 1;

// loop over upper triangular part of W to identify neighbor pairs

for (i in 1:(n - 1)) { for (j in (i + 1):n) {

if (W[i, j] == 1) {

W\_sparse[counter, 1] = i;

W\_sparse[counter, 2] = j;

counter = counter + 1; } } } }

for (i in 1:n) D\_sparse[i] = sum(W[i]);

{

vector[n] invsqrtD;

for (i in 1:n) { invsqrtD[i] = 1 / sqrt(D\_sparse[i]); }

lambda = eigenvalues\_sym(quad\_form(W, diag\_matrix(invsqrtD)));

}

}

parameters {

vector[n] phi; // spatial effects

vector[n] phi\_rep; // spatial effects

real<lower = 0> tau; //precision spatial effects

real beta0;

real<lower = 0, upper = 1> alpha;

}

transformed parameters {

vector[n] RR;

vector[n] mu;

vector[n] murep;

murep= exp(beta0 + phi\_rep + log\_offset);

mu= exp(beta0 + phi + log\_offset);

RR = exp(beta0+phi); // relative risks

}

model {

phi ~ sparse\_car(tau, alpha, W\_sparse, D\_sparse, lambda, n, W\_n);

phi\_rep ~ sparse\_car(tau, alpha, W\_sparse, D\_sparse, lambda, n, W\_n);

beta0 ~ normal(0, 1);

tau ~ gamma(2, 2);

y ~ poisson\_log(beta0 + phi + log\_offset);

}

generated quantities{real log\_lik[n];

int yrep[n];

real <lower=0, upper=1> exc[n];

for (i in 1:n) { log\_lik[i]= poisson\_lpmf(y[i] | mu[i]);

yrep[i] = poisson\_rng(murep[i]);

exc[i] = (yrep[i] > y[i])+0.5\*(yrep[i] ==y[i]);}} // mixed predictive checks

"

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

fitpropercar = sampling(sm,data =D2,iter = 2500,warmup=250,chains = 2,seed= 12345)

summary(fitpropercar,pars=c("beta0","RR","alpha"), probs=c(0.025,0.5,0.975))$summary

**# Relative Risks**

RRsamps <- as.matrix(fitpropercar,pars="RR")

RR.mn=apply(RRsamps,2,mean)

range(RR.mn)

**# Fit**

LLsamps <- as.matrix(fitpropercar,pars="log\_lik")

loo(LLsamps)

**# Poorly fitted observations**

loocase <- as.vector(LOO2$pointwise[,3])

**# pointwise lack of fit, % LOO-IC due to 5% worst fitting cases**

100\*sum(loocase[loocase>quantile(loocase,0.95)])/sum(loocase)

**# mixed predictive checks**

exc.samps= as.matrix(fitpropercar,pars="exc")

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

sum(exc.mn>0.95)

**#**

**# Model 3**

**#**

# data

D3=list(N=N,NN=NN,C=C,adj=adj,d=d,y=y,E=E,K=10,a=seq(1:10),b=5)

**# Code**

model3 <- function() {for (i in 1:N) { y[i] ~ dpois(nu[i]);

nu[i] <- E[i]\*rho[i]

log(rho[i]) <- log(mu[G[i]])+u[i]

u[i] ~ dnorm(0,tau)

# log-likelihoods

LL[i] <- -nu[i]+y[i]\*log(nu[i])-logfact(y[i])

y.s[i] <- equals(y[i],0)+(1-equals(y[i],0))\*y[i]

nu.s[i] <- equals(y[i],0)+(1-equals(y[i],0))\*nu[i]

# deviance contributions

dev[i] <- y[i]\*log(y.s[i]/nu.s[i])-(y[i]-nu[i])

# select cluster

G[i] ~ dcat(p[i,1:K])

for (k in 1:K) {J[i,k] <- equals(G[i],k)

# Potts cluster model

p[i,k] <- U[i,k] /sum(U[i,])

log(U[i,k]) <- omega\*sum(wJ[C[i] + 1 : C[i + 1],k ])}}

for (i in 1 : NN ) {for (k in 1:K) { wJ[i,k] <- J[adj[i],k]}}

for (k in 1:K) {mu.star[k] ~ dgamma(a[k],b)

Jsum[k] <- sum(J[,k])

# Latent cluster means

mu[k] <- ranked(mu.star[1:K],k)}

# scaled deviance

Dv <- 2\*sum(dev[])

# priors

omega ~ dexp(1)

tau ~ dgamma(1,0.001)}

**# Initial Values**

H=seq(1:10)

H=c(rep(H,98),seq(1:3))

init1=list(mu.star=c(0.2,0.5,0.7,0.9,1,1.1,1.5,2,2.5,3),omega=0.5,G=H,tau=100)

init2=list(mu.star=c(0.2,0.4,0.6,0.8,1,1.2,1.6,2,2.5,3),omega=0.3,G=H,tau=200)

inits=list(init1,init2)

**# Estimation**

pars <- c("mu","omega","rho","LL","Dv")

n.iters=10000; n.burnin=500; n.chains=2

R3 = bugs(D3,inits,pars,n.iters,model3,n.chains,n.burnin,debug=T,codaPkg=F,bugs.seed=10)

R3$summary

**# Relative Risk Estimates**

RR.Potts= R3$mean$rho

range(RR.Potts)

**# Hyperparameters**

R3$summary[1:(D$K+1),]

**# log-likelihoods**

LL.samps=R3$sims.matrix[,995:1977]

LOO3=loo(LL.samps)

**# Poorly fitted observations**

loocase <- as.vector(LOO3$pointwise[,3])

area=seq(1:983)

list.loocase <- data.frame(area,loocase,y,E)

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

head(list.loocase,10)

**#**

**# Model 4**

**#**

# data

D4=list(N=N,NN=NN, adj=adj,d=d,y=y,E=E,theta=0.5)

model4 <- function() { tau.s ~ dgamma(1,0.001)

delta~ dgamma(1,0.001)

alpha ~ dnorm(0,0.01)

for (i in 1:N){ y[i] ~ dpois(mu[i])

mu[i] <- E[i]\*rho[i]

LL[i] <- -mu[i]+y[i]\*log(mu[i])-logfact(y[i])

L[i] <- exp(LL[i])

rho[i] <- exp(eta[i])

eta[i] ~ dnorm(nu[i],tau[i])

nu[i] <- alpha+s[i]

w[i] ~ dexp(delta)

res.dist[i] <- (eta[i]-nu[i])\*sqrt(tau[i])

tau[i] <- (theta\*(1-theta)\*delta)/(2\*w[i])}

**# spatial effects prior**

s[1:N] ~ car.normal(adj[], weights[], d[], tau.s)

for (i in 1:NN) {weights[i] <- 1}}

init1=list(tau.s=100,alpha=0,delta=10,s=rep(0,983))

init2=list(tau.s=50,alpha=0,delta=5,s=rep(0,983))

inits=list(init1,init2)

**# Estimation**

pars <- c("delta","res.dist","rho","LL")

n.iters=5000; n.burnin=500; n.chains=2

R4 = bugs(D4,inits,pars,n.iters,model4,n.chains,n.burnin,debug=T,codaPkg=F,bugs.seed=10)

R4$summary

**# log-likelihoods and LOO-IC**

LL.samps=R4$sims.matrix[,1968:2950]

LOO4=loo(LL.samps)

**# Asymmetric Laplace Outlier Indicators**

OUTL.SPMD= R4$mean$res.dist

**# Poorly fitted observations**

loocase <- as.vector(LOO4$pointwise[,3])

area=seq(1:983)

list.loocase <- data.frame(area,loocase,y,E,OUTL.SPMD)

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

head(list.loocase,10)

**# Relative Risk Posterior Means**

RR.SPMD= R4$mean$rho

range(RR.SPMD)

**# Compare Relative Risks**

hist(RR.Potts, col='black', xlim=c(0.6, 1.6),xlab='Relative Risk',main='Figure 6.3 Posterior Mean Relative Risks, Potts vs BYM',ylim=c(0,500))

hist(RR.BYM, col='gray', add=T)

legend("topright", c("Potts", "BYM"), col=c("black", "gray"), lwd=10)

hist(RR.Potts, col='black', xlim=c(0.6, 1.6),xlab='Relative Risk',main='Figure 6.4 Posterior Mean Relative Risks, Potts vs Spatial Median',ylim=c(0,400))

hist(RR.SPMD, col='gray', add=T)

legend("topright", c("Potts", "Spatial Median"), col=c("black", "gray"), lwd=10)