library(loo)

library(R2OpenBUGS)

library(MCMCvis)

library(maptools)

library(spdep)

library(CARBayes)

setwd("C:/R Files BHMRA")

**# shapefile East London electoral wards**

ELmap <- readShapePoly("Example\_6\_3")

ELnb <- poly2nb(ELmap, queen=F)

**# Binary Interaction (Adjacency) Matrix**

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

**# Number of areas**

N =dim(W)[1]

N1=N+1

**# Obtain adjacency list from interaction 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)}

map=adjlist(W,N)

**# Other spatial data**

d=C=numeric()

d=as.vector(rowSums(W)) # d[1:N] contains numbers of neighbours (locality sizes)

NN=sum(d) # total number of neighbours (locality size)

**# cumulative index of locality sizes**

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

**# Area Deprivation Index**

IMD=c(33.7,35.2,32.5,33.3,23.8,31.9,42.2,34.3,35.9,18.7,33.0,35.7,33.1,39.5,34.8,36.0,22.8,

38.5,39.1,48.8,37.5,47.7,42.5,45.1,46.6,47.4,44.3,40.8,43.0,36.5,43.3,48.9,42.3,39.6,42.5,42.3,18.4,6.9,16.9,8.5,31.6,10.5,18.0,18.6,26.6,11.3,16.1,9.9,17.4,16.4,11.9,20.6,9.8,6.5,44.5,38.3,

50.5,52.5,43.4,35.5,38.3,34.9,34.1,39.3,35.5,36.4,39.0,38.4,37.2,38.0,39.4,43.1,32.9,38.5,17.0,13.5,15.3,15.9,10.7,11.4,26.0,15.0,18.0,18.2,21.4,27.7,28.6,16.5,8.1,18.7,16.6,20.3,14.9,22.5,

14.7,47.3,48.9,35.2,48.0,38.9,53.2,55.5,47.2,40.9,51.1,31.8,48.6,31.9,48.2,50.6,48.0,45.8,34.2,38.3,25.9,19.4,18.6,28.3,25.8,24.1,25.8,30.9,37.3,35.1,24.3,33.5,38.5,26.4,34.8,28.7,29.8,34.5, 16.4)

**# Binomial response (LLTI among people aged 50-59)**

y=c(301,349,317,268,304,336,272,353,280,244,313,312,321,211,284,278,267,323,255,

365,306,303,318,327,310,336,355,398,292,281,316,408,279,299,351,350,340,257,325,284,

436,262,349,408,428,316,303,296,329,303,303,351,265,249,284,361,402,433,364,342,302,

338,395,399,302,323,375,352,397,350,164,336,294,394,314,290,302,227,203,319,285,376,

317,374,299,401,360,365,206,364,266,330,233,328,265,321,313,319,316,299,322,346,328,

279,324,246,325,198,292,201,361,238,318,298,276,239,257,311,270,244,255,299,279,325,

253,291,320,272,307,284,279,286,160)

**# Binomial denominator**

T=c(881,946,929,949,1158,1151,786,1125,899,1122,948,900,1086,687,877,971,1064,

838,727,931,840,854,940,812,830,759,870,1006,818,846,818,965,748,838,957,879,1380,

1734,1612,1881,1399,1652,1521,1650,1482,1656,1558,1917,1527,1531,1683,1599,1675,

1900,781,1090,993,1097,1033,994,737,1054,1066,991,832,858,987,975,1042,1047,432,902,

924,1109,1464,1438,1387,1135,1176,1712,950,1443,1344,1491,1089,1348,992,1440,1523,

1475,1205,1184,1205,1094,1596,829,745,895,739,928,726,854,843,729,755,830,838,830,

751,447,771,588,936,851,1087,1255,1411,1031,948,1229,1164,935,947,930,1186,1036,984,

941,911,1208,910,1000,1105)

**#**

**# Leroux et al (1999) Model**

**#**

D=list(N=N,NN=NN,C=C,map=map,d=d,y=y,T=T)

model1 <- function() {for (i in 1:N) {**# model for observed data**

y[i] ~ dbin(p[i],T[i])

logit(p[i]) <- alpha+s[i]

**# predictive checks**

yrep[i] ~ dbin(p[i],T[i]);

check[i] <- step(yrep[i]-y[i]-0.001)+0.5\*equals(yrep[i],y[i])

**# log-likelihood**

loglik[i] <- logfact(T[i])-logfact(y[i])-logfact(T[i]-y[i])

+y[i]\*log(p[i])+(T[i]-y[i])\*log(1-p[i])}

# error vector and deprivation over neighbours (listed in map vector)

for (i in 1:NN) { Ws[i] <- s[map[i]]}

**# Priors**

tau.s ~ dgamma(1,0.001)

sd.s <- 1/sqrt(tau.s)

alpha ~ dnorm(0,0.001)

lam ~ dbeta(1,1)

**# Leroux et al prior**

for (i in 1:N) {s[i] ~ dnorm(S[i],tau[i])

tau[i] <- tau.s \* (1-lam+lam\*d[i])

S[i] <- (lam/(1-lam+lam\*d[i]))\*sum(Ws[C[i]+1:C[i+1] ])}}

**# Initial Values**

init1= list(tau.s=10,alpha=0,s=rep(0,133))

init2= list(tau.s=5,alpha=0,s=rep(0,133))

inits=list(init1,init2)

**# Estimation**

pars = c("loglik","check","lam","sd.s","p")

n.iters=10000; 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$sims.list$loglik)

waic(R1$sims.list$loglik)

**# predictive checks**

sum(apply(R1$sims.list$check,2,mean) < 0.05)+ sum(apply(R1$sims.list$check,2,mean) > 0.95)

#

**# CARBayes, Leroux et al (1999)**

#

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

f = y ~1

M= S.CARleroux(formula=f, family="binomial", data=D, W=W, trials=T,

burnin=10000,n.sample=100000,thin=1, prior.tau2=c(1,0.001))

**# Results**

M

**# Fit**

M$modelfit

#

**# Adaptive Leroux model**

#

D=list(N=N,NN=NN,C=C,map=map,d=d,y=y,T=T,IMD=IMD)

model2 <- function() {for (i in 1:N) {

**# model for observed data**

y[i] ~ dbin(p[i],T[i])

logit(p[i]) <- alpha+s[i]

**# predictive checks**

yrep[i] ~ dbin(p[i],T[i]);

check[i] <- step(yrep[i]-y[i]-0.001)+0.5\*equals(yrep[i],y[i])

**# log-likelihood**

loglik[i] <- logfact(T[i])-logfact(y[i])-logfact(T[i]-y[i])

+y[i]\*log(p[i])+(T[i]-y[i])\*log(1-p[i])

**# Absolute difference in risk factor between area i and its locality**

absdif.IMD[i] <- abs(IMD[i]-sum(WIMD[C[i]+1:C[i+1] ])/d[i])}

**# error vector and deprivation over neighbours (listed in map vector)**

for (i in 1:NN) { Ws[i] <- s[map[i]]\*lam[map[i]];

WIMD[i] <- IMD[map[i]]}

**# Priors**

tau.s ~ dgamma(1,0.001);

alpha ~ dnorm(0,0.001);

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

tau.lam ~ dgamma(1,1)

lam.mu <- mean(lam[])

sd[1] <- sqrt(1/tau.lam)

sd[2] <- sqrt(1/tau.s)

**# Adaptive spatial prior**

for (i in 1:N) {lam[i] <- exp(lgt.lam[i])/(1+exp(lgt.lam[i]))

**# regression using risk factor discrepancy**

lgt.lam[i] ~ dnorm(m.lam[i],tau.lam)

m.lam[i] <- gamma[1]+gamma[2]\*(absdif.IMD[i]-mean(absdif.IMD[]))

s[i] ~ dnorm(S[i],tau[i])

tau[i] <- tau.s \* (1-lam[i]+lam[i]\*d[i])

S[i] <- (lam[i]/(1-lam[i]+lam[i]\*d[i]))\*sum(Ws[C[i]+1:C[i+1] ])}}

**# Initial Values**

init1= list(tau.s=10,alpha=0,tau.lam=10,gamma=c(2.5,-0.2),

lgt.lam=rep(0.75,133),s=rep(0,133))

init2= list(tau.s=5,alpha=0,tau.lam=20,gamma=c(2.7,-0.3),

lgt.lam=rep(0.5,133),s=rep(0,133))

inits=list(init1,init2)

**# Estimation**

pars = c("sd","loglik","gamma","check","lam.mu","p","lam")

n.iters=10000; n.burnin =1000; n.chains=2

R2 = bugs(D,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,

codaPkg = F,bugs.seed=10)

R2$summary

**# Fit**

loo(R2$sims.list$loglik)

waic(R2$sims.list$loglik)

**# mixed predictive checks**

sum(apply(R2$sims.list$check,2,mean) < 0.05)+ sum(apply(R2$sims.list$check,2,mean) > 0.95)

**# varying posterior mean lambda[i]**

pm.lam=apply(R2$sims.list$lam,2,mean)

hist(pm.lam,breaks=20)

**# Map of local lambda**

library(rgdal)

library(GISTools)

ELWDS <- readOGR(dsn=".",layer="Example\_6\_3")

# find out eastings, northings etc

summary(ELWDS)

plot(ELWDS)

shades = auto.shading(pm.lam)

par(mar = c(0, 0, 0, 0))

choropleth(ELWDS, pm.lam,shades)

choro.legend(532000,177000,shades,fmt="%4.2f",cex=0.75,title='Local lambda')

text(542000, 199000, "Figure 6.2 Local Leroux Dependence Parameters",cex=0.8)

library(RColorBrewer)

plot(ELWDS)

shades = auto.shading(pm.lam,n=4,cutter= quantileCuts,cols=brewer.pal(4,'Greys'))

par(mar = c(0, 0, 0, 0))

choropleth(ELWDS, pm.lam,shades)

choro.legend(532000,177000,shades,fmt="%4.2f",cex=0.75,title='Local lambda')

text(542000, 199000, "Figure 6.2 Local Leroux Dependence Parameters",cex=0.8)