library(easypackages)

libraries("INLA","spdep","INLABMA","maptools","rstan")

setwd("C:/R Files BHMRA")

**# shapefile East London electoral wards**

ELmap <- readShapePoly("Example\_6\_1")

ELnb <- poly2nb(ELmap, queen=F)

**# 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)

# LLTI totals

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)

# LLTI population 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)

**# Long Term Illness Rate per 1000**

y=1000\*y/T

**# Dataset**

EL=list(y=y,IMD=IMD)

**# neighbours list**

lw=nb2listw(ELnb, glist=NULL, style="W", zero.policy=NULL)

**# Sparse Adjacency matrix**

W = as(as\_dgRMatrix\_listw(nb2listw(ELnb)), "CsparseMatrix")

**# Maximum Likelihood estimation**

**# Spatial Error Model**

MLE.SEM <- errorsarlm(y ~ IMD,listw=lw, method="eigen", quiet=F)

summary(MLE.SEM)

**# Spatial Lag Model**

MLE.SLM <- lagsarlm(y ~ IMD,listw=lw, method="eigen", type="lag",quiet=F)

summary(MLE.SLM)

**# Spatial Durbin Model**

MLE.SDM = lagsarlm(y ~ IMD, listw=lw, method="eigen", type="mixed", quiet=F)

summary(MLE.SDM)

**#**

**# INLA estimation**

**#**

**# Spatial Durbin Model**

**#**

grid.rho = seq(0.2, 0.9, length.out=20)

sdminla = mclapply(grid.rho, function(rho){ sdm.inla(form, d=EL, W=W, rho=rho,

family = "gaussian", impacts=FALSE,control.family = list(hyper = zero.variance),

control.predictor=list(compute=TRUE), control.compute=list(dic=TRUE, cpo=TRUE),

control.inla=list(print.joint.hyper=TRUE), verbose=FALSE )})

SDM.BMA = INLABMA(sdminla, grid.rho, 0, usenormal=TRUE)

**# Estimates for spatial error parameter**

RHO.SDM=c(SDM.BMA$rho$mean,SDM.BMA$rho$sd,SDM.BMA$rho$quantiles)

**# Regression Parameters**

SDM.BMA$summary.fixed

# Fit

-2\*SDM.BMA$dic$dic

**#**

**# STAN Analysis**

**#**

**# Spatial Errors Model (Spatial Autoregressive Error Model)**

**#**

N=133

I <- diag(N)

W <- nb2mat(ELnb)

D=list(N=N,W=W,I=I,y=y,x=IMD)

model="

data {

int N;

vector[N] x;

vector[N] y;

matrix<lower=0>[N,N] W;

matrix<lower=0,upper=1>[N,N] I;

}

parameters {

real beta;

real alpha;

real<lower = 0> sigma;

real<lower=-1,upper=1> rho;

}

model {

y ~ multi\_normal\_prec(alpha + x \* beta, crossprod(I - rho \* W)/(sigma\*sigma));

}

generated quantities

{real LL;

LL= multi\_normal\_prec\_lpdf(y | alpha + x \* beta, crossprod(I - rho \* W)/(sigma\*sigma));

}

"

**# Estimation**

fit=stan(model\_code = model,data=D,iter=1500,warmup = 250,chains=2)

# Posterior Summary

print(fit,digits=3)

# mean deviance plus number of parameters

DIC=-2\*summary(fit, pars = c("LL"))$summary[1]+4

**#**

**# STAN Analysis**

**#**

**# Spatial Moving Average Error Model**

**#**

model="

data {

int N;

vector[N] x;

vector[N] y;

matrix<lower=0>[N,N] W;

matrix<lower=0,upper=1>[N,N] I;

}

parameters {

real beta;

real alpha;

real<lower = 0> sigma;

real<lower=-1,upper=1> rho;

}

model {

y ~ multi\_normal(alpha + x \* beta, crossprod(I + rho \* W)\*sigma^2);

}

generated quantities

{real LL;

LL= multi\_normal\_lpdf(y | alpha + x \* beta, crossprod(I + rho \* W)\*sigma^2);

}

"

**# Estimation**

fit=stan(model\_code = model,data=D,iter =1500,warmup = 250,chains=2)

# Posterior Summary

print(fit,digits=3)

# Fit

DIC=-2\*summary(fit, pars = c("LL"))$summary[1]+4

**#**

**# STAN analysis**

**#**

**# Spatial Lag Model**

**#**

N=133

I <- diag(N)

W <- nb2mat(ELnb)

eigW <- eigen(W)

eigval <- eigW$values

Wy=W%\*%y

Wy=as.vector(Wy)

D=list(N=N,W=W,I=I,y=y,x=IMD,Wy=Wy,eigval=eigval)

model="

data {

int N;

vector[N] x;

vector[N] y;

vector[N] eigval;

vector[N] Wy;

matrix<lower=0>[N,N] W;

matrix<lower=0,upper=1>[N,N] I;

}

parameters {

real beta;

real alpha;

real<lower = 0.001 > sigma;

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

}

transformed parameters {

vector[N] eps2;

vector[N] rhoeig;

vector[N] summands;

for (i in 1:N) { eps2[i] = (y[i]-rho\*Wy[i]-alpha-x[i]\*beta)\* (y[i]-rho\*Wy[i]-alpha-x[i]\*beta);}

for (i in 1:N) { rhoeig[i]=log(1-rho\*eigval[i]);}

for (i in 1:N) {summands[i]= -0.919-0.5\*log(sigma^2)+rhoeig[i]-eps2[i]/(2\*sigma\*sigma);}

}

model {

for (i in 1:N) {target += -0.919-0.5\*log(sigma^2)+rhoeig[i]-eps2[i]/(2\*sigma\*sigma);}

}

generated quantities {

real LL;

LL = sum(summands);

}"

**# Estimation**

fit=stan(model\_code = model,data=D,iter = 1500,warmup = 250,chains=2)

**# Posterior Summary**

print(fit,digits=3)

**# Fit**

DIC=-2\*summary(fit, pars = c("LL"))$summary[1]+4

#

# SPATIALLY VARYING BETA

#

e=rep(1,N)

D=list(N=N,W=W,I=I,y=y,x=IMD,e=e)

model="

data {

int N;

vector[N] x;

vector[N] y;

matrix<lower=0>[N,N] W;

vector<upper=1>[N] e;

matrix<lower=0,upper=1>[N,N] I;

}

parameters {

vector[N] beta;

real alpha;

real beta\_mu;

real<lower = 0> sigma;

real<lower = 0> sigma\_b;

real<lower=-1,upper=1> rho;

real<lower=-1,upper=1> rho\_b;

}

model {

beta ~ multi\_normal\_prec(e \* beta\_mu, tcrossprod(I - rho\_b \* W)/(sigma\_b\*sigma\_b));

y ~ multi\_normal\_prec(alpha + x .\* beta, tcrossprod(I - rho \* W)/(sigma\*sigma));

}

generated quantities

{

real LL;

LL= multi\_normal\_prec\_lpdf(y | alpha + x .\* beta, crossprod(I - rho \* W)/(sigma\*sigma));

}

"

**# Estimation**

fit=stan(model\_code = model,data=D,iter=1500,warmup = 250,chains=2)

# Posterior Summary

print(fit,digits=3)

beta.samps <- as.matrix(fit,pars="beta")

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

hist(beta.m, xlab = expression(paste("Posterior mean ", beta)),

main="Figure 6.1 Histogram of Spatially Varying Predictor Effect",breaks=19,

col="gray")