setwd("C:/R files BHMRA")

options(scipen=999)

require(mcmcse)

require(spatialprobit)

require(INLA)

library(R2OpenBUGS)

library(coda)

library(mcmcplots)

attach("DS\_7\_14.Rdata")

**# DISTANCE BASED ADJACENCY**

britishLA <- read.table("Example\_7\_14\_brexitSAR.txt",header=T)

knn <- knearneigh(cbind(britishLA$east, britishLA$north), k=5)

nb <- knn2nb(knn)

lw <- nb2listw(nb, style="W")

W <- as(as\_dgRMatrix\_listw(lw), "CsparseMatrix")

**# SPATIAL LAG PROBIT**

**# univariate binary regression on UK non-born**

X <- matrix(,380,2)

ybin <- britishLA$hibrex

X[,1] <- 1

X[,2] <- britishLA$nonUKborn/100

SARP1 = sar\_probit\_mcmc(ybin,X,W, ndraw=5000, burn.in=500)

summary(SARP1)

**# multiple binary spatial lag regression**

X <- matrix(,380,5)

X[,1] <- 1

X[,2] <- britishLA$hiqual/100

X[,3] <- britishLA$ov65/100

X[,4] <- britishLA$nonUKborn/100

X[,5] <- britishLA$popden/1000

SARP2 = sar\_probit\_mcmc(ybin,X,W, ndraw=5000, burn.in=500)

summary(SARP2)

**# SPATIAL LAG MODEL, LOGIT BREXIT PROPORTION AS RESPONSE**

n=380

britishLA$idx <- 1:n

**# response**

logit <-function(x) log(x/(1-x))

ylogit <- logit(britishLA$brex)

**# predictors**

X1 <- britishLA$hiqual/100

X2 <- britishLA$ov65/100

X3 <- britishLA$nonUKborn/100

X4 <- britishLA$popden/1000

**# regression model**

M = ylogit ~ X1+X2+X3+X4

mmatrix <- model.matrix(M, britishLA)

# Zero-variance for error term

zero.variance = list(prec=list(initial = 1, fixed=T))

**# Compute eigenvalues : obtain rho.min and rho.max**

e = eigenw(lw)

re.idx = which(abs(Im(e)) < 1e-6)

rho.max = 1/max(Re(e[re.idx]))

rho.min = 1/min(Re(e[re.idx]))

rho = mean(c(rho.min, rho.max))

betaprec <- 0.1

Q.beta = Diagonal(n=ncol(mmatrix), betaprec)

**# Priors on hyperparameters**

hyper = list(prec = list(prior = "loggamma",param = c(0.01, 0.01)),

rho = list(initial=0,prior = "logitbeta",param = c(1,1)))

SLM <- inla( ylogit ~ -1 + f(idx, model="slm",args.slm=list(rho.min = rho.min,

rho.max = rho.max,W=W,X=mmatrix,Q.beta=Q.beta),hyper=hyper),

data=britishLA, family="gaussian",control.family = list(hyper=zero.variance),

control.compute=list(dic=T, cpo=T))

summary(SLM)

**# regression coefficient estimates**

SLM$summary.random$idx[n+1:ncol(mmatrix),]

**# NO PREDICTOR SELECTION, BINARY ADJACENCY BASED ON CONTIGUITY**

model1 <- function() {for (i in 1:N) { logit(pi[i]) <- beta0+sum(beta.terms[i,])+r[i]

y[i] ~ dbin(pi[i],V[i])

LL[i] <- y[i]\*log(pi[i])+(V[i]-y[i])\*log(1-pi[i])+logfact(V[i])-logfact(V[i]-y[i])-logfact(y[i])

# predictors: % high educ, % over 65, % nonUK born, popn density

for (j in 1:p) {beta.terms[i,j] <- beta[j]\*x.s[i,j]}

x.s[i,1] <- (x1[i]-mean(x1[]))/sd(x1[])

x.s[i,2] <- (x2[i]-mean(x2[]))/sd(x2[])

x.s[i,3] <- (x3[i]-mean(x3[]))/sd(x3[])

x.s[i,4] <- (x4[i]-mean(x4[]))/sd(x4[])}

# priors

beta0 ~ dnorm(0,0.001)

for (j in 1:p) {beta[j] ~dnorm(0,0.001)

pos.beta[j] <- step(beta[j])}

# Leroux et al prior

for (i in 1:NN) {r.neigh[i] <- r[adj[i]]}

for (i in 1:N) {r[i] ~ dnorm(R[i],taur[i]);

R[i] <- lambda\*sum(r.neigh[cum[i]+1:cum[i+1] ])/(1-lambda+lambda\*num[i])

taur[i] <- tau \* (1-lambda+lambda\*num[i])}

tau ~ dgamma(1,0.01)

lambda ~ dunif(0,1)}

# Initial Values and Estimation

init1 = list(beta0=0,beta=c(0,0,0,0),tau=100,lambda=0.7,r=rep(0,380))

init2 = list(beta0=0.2,beta=c(-0.5,0.5,0,0),tau=200, lambda=0.5,r=rep(0,380))

inits = list(init1,init2); params = list("beta","beta0","lambda","pos.beta")

n.iters=10000; n.burnin =n.iters/2; n.chains=2; n.thin=10; n.samps = n.chains\*(n.iters-n.burnin)

M1 = bugs(DS\_7\_14,inits,params,n.iters,model1,n.chains, n.burnin,n.thin,debug=T,

codaPkg = F, DIC=T,bugs.seed=10)

M1$summary

**# NON-SPATIAL SSVS PREDICTOR SELECTION**

model2 <- function() {for(i in 1:N) { logit(pi[i]) <- sum(beta.terms[i,])+r[i]

y[i] ~ dbin(pi[i],V[i])

LL[i] <- y[i]\*log(pi[i])+(V[i]-y[i])\*log(1-pi[i])+logfact(V[i])-logfact(V[i]-y[i])-logfact(y[i])

# predictors: % high educ, % over 65, % nonUK born, popn density

for (j in 1:p) {beta.terms[i,j] <- x.s[i,j]\*beta.r[j]}

x.s[i,1] <- (x1[i]-mean(x1[]))/sd(x1[])

x.s[i,2] <- (x2[i]-mean(x2[]))/sd(x2[])

x.s[i,3] <- (x3[i]-mean(x3[]))/sd(x3[])

x.s[i,4] <- (x4[i]-mean(x4[]))/sd(x4[])}

# Priors

beta0 ~ dnorm(0,0.001)

for (j in 1:p) {beta[j] ~ dnorm(0,tau.beta[j]);

tau.beta[j] <- equals(gam[j],1)+equals(gam[j],0)\*1000

beta.r[j] <- beta[j]\*gam[j];

# selection indicator

gam[j] ~ dbern(omega)}

omega ~ dbeta(1,1)

# Leroux et al Prior

for (i in 1:NN) {r.neigh[i] <- r[adj[i]]}

for (i in 1:N) {r[i] ~ dnorm(R[i],taur[i]);

R[i] <- beta0+lambda\*sum(r.neigh[cum[i]+1:cum[i+1] ])/(1-lambda+lambda\*num[i])

taur[i] <- tau \* (1-lambda+lambda\*num[i])}

tau ~ dgamma(1,0.01)

lambda ~ dunif(0,1)}

# Initial Values and Estimation

init1 <- list(beta0=0,beta=c(0,0,0,0),tau=100,r=rep(0,380))

init2 <- list(beta0=0.2,beta=c(-0.5,0.5,0,0),tau=200,r=rep(0,380))

inits <- list(init1,init2)

params <- list("beta.r","beta0","gam","omega","lambda")

n.iters= 100000; n.burnin =5000; n.chains=2; n.thin=1

M2 <- bugs(DS\_7\_14, inits, params,n.iters,model2,n.chains, n.burnin,n.thin,debug=T, codaPkg = F,DIC=T,bugs.seed=10)

M2$summary

**# SPATIAL PREDICTOR SELECTION; BINARY ADJACENCY**

model3 <- function() {for(i in 1:N) { logit(pi[i]) <- beta0+sum(beta.terms[i,])+s[i]

y[i] ~ dbin(pi[i],V[i])

LL[i] <- y[i]\*log(pi[i])+(V[i]-y[i])\*log(1-pi[i])+logfact(V[i])-logfact(V[i]-y[i])-logfact(y[i])

# predictors: % high educ, % over 65, % nonUK born, popn density

x.s[i,1] <- (x1[i]-mean(x1[]))/sd(x1[])

x.s[i,2] <- (x2[i]-mean(x2[]))/sd(x2[])

x.s[i,3] <- (x3[i]-mean(x3[]))/sd(x3[])

x.s[i,4] <- (x4[i]-mean(x4[]))/sd(x4[])

for (j in 1:p) {beta.terms[i,j] <- beta[i,j]\*x.s[i,j]\*gamma[i,j]

beta[i,j] ~ dnorm(mu.beta[j],tau.beta[j]);

gamma[i,j] ~ dbern(rho[i,j])

logit(rho[i,j]) <- omega[j]+ r[i,j]

beta.r[i,j] <- beta[i,j]\*gamma[i,j]}}

for (j in 1:p) {mu.beta[j] ~ dnorm(0,0.1)

omega[j] ~ dnorm(0,0.1)

tau.beta[j] ~ dgamma(1,0.1)

indic.retain[j] <- mean(gamma[,j])

beta.mean[j] <- mean(beta.r[,j])}

beta0 ~ dnorm(0,0.001)

# Leroux prior for residual

tau.s ~ dgamma(1,0.01)

lambda.s ~ dunif(0,1)

for (i in 1:NN) {s.neigh[i] <- s[adj[i]]}

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

S[i] <- lambda.s\*sum(s.neigh[cum[i]+1:cum[i+1]])/(1-lambda.s+lambda.s\*num[i])

taus[i] <- tau.s\* (1-lambda.s+lambda.s\*num[i])}

# ICAR priors for variable selection

for (j in 1:4) { tau[j] ~ dgamma(1,0.01)}

for (i in 1:N) {r[i,1] <- r1[i]

r[i,2] <- r2[i]

r[i,3] <- r3[i]

r[i,4] <- r4[i]}

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

r1[1:N] ~ car.normal(adj[], weights[], num[], tau[1])

r2[1:N] ~ car.normal(adj[], weights[], num[], tau[2])

r3[1:N] ~ car.normal(adj[], weights[], num[], tau[3])

r4[1:N] ~ car.normal(adj[], weights[], num[], tau[4])

}

# Initial Values and Estimation

init1 <- list(beta0=0,mu.beta=rep(0,4),tau.beta=rep(100,4), tau=rep(100,4),

tau.s=10, lambda.s=0.9,omega=c(4,-4,-4,4),s=rep(0,380),

r1=rep(0,380), r2=rep(0,380), r3=rep(0,380), r4=rep(0,380))

init2 <- list(beta0=0.2,mu.beta=rep(0.1,4),tau.beta=rep(10,4), tau=rep(10,4),

tau.s=100, omega=c(3,-2,-2,3),s=rep(0,380),

r1=rep(0,380), r2=rep(0,380), r3=rep(0,380), r4=rep(0,380))

inits <- list(init1,init2)

params <- list("beta.mean","indic.retain","lambda.s","omega")

n.iters= 100000; n.burnin =1000; n.chains=2; n.thin=1

DS\_7\_14$NN=1994

M3 <- bugs(DS\_7\_14,inits,params,n.iters,model3,n.chains, n.burnin,n.thin,debug=T,

codaPkg = F,DIC=T,bugs.seed=10)

M3$summary