library(INLA)

library(akima)

library(gplots)

library(spBayes)

library(MBA)

library(spNNGP)

set.seed(10)

setwd("C:/R files BHMRA")

**# earthquake locations (with magnitude)**

quake <- read.table("DS\_6\_6.txt",header=T)

quake.coord <- list(x= quake$Lon,y= quake$Lat)

**# Plot of earthquake locations**

par(mai=c(0.5,0.3,0.3,0))

plot(quake.coord, cex=0.1, main=NULL)

#

**# spde Model for magnitudes**

**#**

coordinates=cbind(quake$Lon,y= quake$Lat)

**#**

**# spde Model, Use k to vary density of mesh**

**#**

k=0.1

mesh=inla.mesh.2d(coordinates,max.edge=c(1/k,2/k),cutoff=0.1/k)

plot(mesh)

points(coordinates, col = "red")

**# Projector matrix**

A = inla.spde.make.A(mesh=mesh,loc=coordinates)

dim(A)

**# Spatial effect in spde regression**

spde=inla.spde2.matern(mesh=mesh)

y=quake$magnitude

quake$id=seq(1:29452)

stk.e <- inla.stack(tag='est',data=list(y=y),A=list(A, 1),effects=list(s=1:spde$n.spde,id=quake$id))

formula=y~0+f(s,model=spde)

res1=inla(formula,data=inla.stack.data(stk.e),control.predictor=list(A=inla.stack.A(stk.e)))

summary(res1)

**# random effect for each point**

gproj <- inla.mesh.projector(mesh, coordinates)

g.mean <- inla.mesh.project(gproj, res1$summary.random$s$mean)

yhat1=g.mean

cor(yhat1,y)

**# compare actual and predicted**

DF=data.frame(quake$id,y,yhat1)

**# Contour Plot of Predictions**

fld <- interp(quake$Lon,y= quake$Lat,yhat1,duplicate="median")

filled.contour(fld, col=colorpanel(7, "white", "grey10"), nlevels=7,xlab="Longitude",ylab="Latitude",main="Predicted Magnitude")

**#**

**# spde Model, Higher k to increase density of mesh**

**#**

k=1

mesh=inla.mesh.2d(coordinates,max.edge=c(1/k,2/k),cutoff=0.1/k)

plot(mesh)

points(coordinates, col = "red")

**# Projector matrix**

A = inla.spde.make.A(mesh=mesh,loc=coordinates)

dim(A)

**# Spatial effect in spde regression**

spde=inla.spde2.matern(mesh=mesh)

stk.e <- inla.stack(tag='est',data=list(y=y),A=list(A, 1),effects=list(s=1:spde$n.spde,id=quake$id))

formula=y~0+f(s,model=spde)

res2=inla(formula,data=inla.stack.data(stk.e),control.predictor=list(A=inla.stack.A(stk.e)))

**# random effect for each point**

gproj <- inla.mesh.projector(mesh, coordinates)

g.mean <- inla.mesh.project(gproj, res2$summary.random$s$mean)

yhat2=g.mean

cor(yhat2,y)

**#**

**# spBayes on 10% sample dataset**

**#**

N=nrow(quake)

quake$Lon= quake$Lon+runif(N,-0.001,0.001)

quake$Lat= quake$Lat+runif(N,-0.001,0.001)

nsamp=round(N/10)

quakes = quake[sample(1:N, nsamp,replace=F),]

Y <- quakes$magnitude

n = length(Y)

coords <- as.matrix(quakes[1:n,c("Lon", "Lat")])

**# define knots**

m=round(n/10)

knots = coords[sample(1:n, m,replace=F),]

**# estimate variogram for initial parameter values**

D=data.frame(lon=quakes$Lon,lat=quakes$Lat,Y)

library(sp); library(gstat)

coordinates(D) = ~lon+lat

v = variogram(Y~1, D)

fit.variogram(v, vgm("Exp"))

**#**

**# spBayes Exponential spatial decay**

**#**

n.samples=2000

m.1= spLM(Y~1,coords=coords,knots=knots,

starting=list("phi"=0.31,"sigma.sq"=0.17,"tau.sq"=0.27),

tuning=list("phi"=0.02, "sigma.sq"=0.01,"tau.sq"=0.01),

priors=list("phi.Unif"=c(0.001,1),"sigma.sq.IG"=c(1, 0.01),"tau.sq.IG"=c(1, 0.01)),

cov.model="exponential",n.samples=n.samples,n.report=100)

**# parameter estimates**

burn.in <- 0.5\*n.samples

m.1 <-spRecover(m.1, start=burn.in, verbose=FALSE)

round(summary(m.1$p.theta.recover.samples)$quantiles[,c(3,1,5)],2)

**# Fitted Values**

sp.hat <- rowMeans(m.1$p.w.recover.samples)

b0.mean= mean(m.1$p.beta.recover.samples)

yhat= b0.mean+sp.hat

cor(Y,yhat)

**# Fit measures**

m.1.DIC <- spDiag(m.1, start=burn.in, verbose=FALSE)

**# Compare observed and predicted surfaces**

par(mfrow=c(1,2))

obs.surf <-

mba.surf(cbind(coords, Y), no.X=100, no.Y=100, extend=T)$xyz.est

image(obs.surf, xaxs = "r", yaxs = "r", main="Observed response")

contour(obs.surf, add=T)

**# Plot Predicted Surface**

pred.surf <- mba.surf(cbind(coords, yhat), no.X=100, no.Y=100, extend=T)$xyz.est

image(pred.surf, xaxs = "r", yaxs = "r", main="Estimated response")

contour(pred.surf, add=T)

**#**

**# spBayes Spherical Covariance**

**#**

m.2= spLM(Y~1,coords=coords,knots=knots,

starting=list("phi"=0.12,"sigma.sq"=0.13,"tau.sq"=0.35),

tuning=list("phi"=0.01, "sigma.sq"=0.01,"tau.sq"=0.01),

priors=list("phi.Unif"=c(0.001,1),"sigma.sq.IG"=c(1, 0.01),"tau.sq.IG"=c(1, 0.01)),

cov.model="spherical",n.samples=n.samples,n.report=100)

**# parameter estimates**

m.2 <-spRecover(m.2, start=burn.in, verbose=FALSE)

round(summary(m.2$p.theta.recover.samples)$quantiles[,c(3,1,5)],2)

**# Fitted Values**

sp.hat <- rowMeans(m.2$p.w.recover.samples)

b0.mean= mean(m.2$p.beta.recover.samples)

yhat= b0.mean+sp.hat

cor(Y,yhat)

**# Fit measures**

m.2.DIC <- spDiag(m.2, start=burn.in, verbose=FALSE)

**#**

**# NNGP**

**#**

n=nrow(quake)

attach(quake)

set.seed(10)

**# Jigger coordinates**

Lon=Lon+runif(n,-0.001,0.001)

Lat=Lat+runif(n,-0.001,0.001)

coords <- cbind(Lon, Lat)

y=magnitude

**# initial parameter values, exponential covariance**

starting =list("phi"=0.31, "sigma.sq"=0.17, "tau.sq"=0.27)

cov.model = "exponential"

n.samples= 5000

tuning = list("phi"=0.025, "sigma.sq"=0.01, "tau.sq"=0.01)

priors = list("phi.Unif"=c(0.01, 5), "sigma.sq.IG"=c(1, 0.01), "tau.sq.IG"=c(1, 0.01))

m.nngp = spNNGP(y~1, coords=coords, starting=starting, method="sequential", n.neighbors=15,

tuning=tuning, priors=priors, cov.model=cov.model,n.samples=n.samples, n.omp.threads=1)

**# Parameter summary**

round(summary(m.nngp$p.theta.samples)$quantiles[,c(3,1,5)],2)

**# predictions vs actual magnitudes**

w.hat =apply(m.nngp$p.w.samples, 1, median)

b0=mean(m.nngp$p.beta.samples)

yhat=b0+w.hat

cor(y,yhat)

plot(y,yhat)

**# map NNGP predictions**

yhat.surf=mba.surf(cbind(coords, yhat), no.X=100, no.Y=100, extend=T)$xyz.est

library(gplots)

colorpanel(7, "white", "grey10")

filled.contour(yhat.surf, col=colorpanel(7, "white", "grey10"), nlevels=7, plot.title = title(main=" Figure 6.5 Magnitude Predictions from NNGP", xlab ="Longitude", ylab ="Latitude"))