Appendix-Code

Laha Ale

5/1/2019

Step 1: Read preproccess data

```
library(dplyr)
library(lubridate)


data <- read.csv("data/clean.csv")

# format time
data <- data %>% mutate(start_time = ymd_hms(start_time)) %>%
  mutate(end_time = ymd_hms(end_time))

# compute connected time
data["con_time"] <- as.numeric(difftime(ymd_hms(data$end_time),
                                        ymd_hms(data$start_time),units = "secs"))
head(data)
```

Step 2: plot 3D distribution

```
library(plotly)
sum_time <- data %>% group_by(long,lat) %>% summarise(sum_time = sum(con_time))
p <- plot_ly(sum_time, x =~long, y = ~lat, z = ~sum_time,type = "scatter3d")
p


#mesh3d
p <- plot_ly(sum_time, x =~long, y = ~lat, z = ~sum_time, type = 'mesh3d')
p
```

Step 3: plot the data on the map

```
Sys.setenv('MAPBOX_TOKEN' = 'pk.eyJ1IjoiYWluaWxhaGEiLCJhIjoiY2p1enYzbm9hMWNzM
           TQzbXcyNWt0M3VwNyJ9._PwdGJ4wfOnzfZ9Ux9Oxvw')

p <- plot_mapbox(data, x = ~long, y = ~lat) %>%
  add_paths(size = I(2)) %>%
  add_segments(x = -100, xend = -50, y = 50, 75)

p

p <- data %>%
  plot_mapbox(lat = ~lat, lon = ~long,
              split = ~class, size=2,
```

```
              mode = 'scattermapbox', hoverinfo='user_id') %>%
  layout(title = 'Shanghai Network (by Laha Ale)',
         font = list(color='white'),
         plot_bgcolor = '#191A1A', paper_bgcolor = '#191A1A',
         mapbox = list(style = 'dark'),
         legend = list(orientation = 'h',
                       font = list(size = 8)),
         margin = list(l = 25, r = 25,
                       b = 25, t = 25,
                       pad = 2))
p
```

Step 4: Remove the outlier with shapfiles

```
dev.off()
library(rgdal)     # R wrapper around GDAL/OGR
library(ggplot2)   # for general plotting
library(sf)
shapefile <- st_read("./data/shp/shang_dis_merged.shp")
plot(shapefile)


name_shape <- shapefile[,'Name']
plot_ly(name_shape,color = ~Name )

name_shape
data <- data %>% filter(long>=120.8544 &
                          long<=121.974 &
                          lat >=30.68889 &
                          lat<=31.86765)


data <- data %>% mutate(rnd_strat = minute(round_date(start_time, "minute"))) %>%
  mutate(rnd_end = minute(round_date(end_time, "minute")))

sum_time <- data %>% group_by(long,lat,rnd_strat) %>% summarise(sum_time = sum(con_time))
num_conn <- data %>% group_by(long,lat) %>% summarise(num_con = n())


ptx <- sum_time[,1:2]
crdref <- CRS('+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0')
ptx <- SpatialPoints(ptx, proj4string=crdref)

ptsdf <- SpatialPointsDataFrame(ptx, data=data.frame(sum_time$sum_time))
ptsdf
library(raster)
plot(name_shape$geometry,border="blue", col="gray")
points(ptsdf,pch=20, cex=1, col="red")
```

Step 5: Visualize the clean data

```
p <- data %>%
  plot_mapbox(lat = ~lat, lon = ~long,
              split = ~class, size=2,
              mode = 'scattermapbox', hoverinfo='user_id') %>%
  layout(title = 'Shanghai Network (by Laha Ale)',
         font = list(color='white'),
         plot_bgcolor = '#191A1A', paper_bgcolor = '#191A1A',
         mapbox = list(style = 'dark'),
         legend = list(orientation = 'h',
                       font = list(size = 8)),
         margin = list(l = 25, r = 25,
                       b = 25, t = 25,
                       pad = 2))
p

p <- plot_ly(sum_time, x =~long,
             y = ~lat, z = ~sum_time,
             color = ~rnd_strat,
             colors = c('#CCFF00', '#0000CC'),type = "scatter3d")
p
```

Step 6: Plot Interpolation

```
library(MBA)
library(fields)

coords <- as.matrix(data[,c("long","lat")])
x.res <- 200; y.res <- 200
surf <- mba.surf(cbind(coords,
                       data$con_time),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="long",
           ylab="lat",
           main="Connection Time" )
points(coords)
contour(surf,add = T)




library(rgl)
library(RColorBrewer)
col.br <- colorRampPalette(c("blue", "cyan", "yellow", "red"))
col <- rbind(0, cbind(matrix(drape.color(surf[[3]],
                                         col = col.br(25)), x.res - 1, y.res-1), 0))
surface3d(surf[[1]], surf[[2]], surf[[3]], col = abs(col))
axes3d()
```

```
title3d(main = "strata", xlab = "long", ylab = "lat", zlab = "conn_time")
drape.plot(surf[[1]], surf[[2]], surf[[3]],
           col = col.br(150), theta = 225, phi = 20,
           border = FALSE, add.legend = FALSE,
           xlab = "long", ylab = "lat", zlab = "conn_time")
image.plot(zlim = range(surf[[3]], na.rm = TRUE),
           legend.only = TRUE, horizontal = FALSE)
```

```
coords <- as.matrix(num_conn[,c("long","lat")])
x.res <- 200; y.res <- 200
surf <- mba.surf(cbind(coords,
                       num_conn$num_con),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="long",
           ylab="lat",
           main="Connection Number" )
points(coords)
contour(surf,add = T)
```

```
library(rgl)
library(RColorBrewer)
col.br <- colorRampPalette(c("blue", "cyan", "yellow", "red"))
col <- rbind(0, cbind(matrix(drape.color(surf[[3]],
                                         col = col.br(25)), x.res - 1, y.res-1), 0))
surface3d(surf[[1]], surf[[2]], surf[[3]], col = abs(col))
axes3d()
title3d(main = "strata", xlab = "long", ylab = "lat", zlab = "conn_num")
drape.plot(surf[[1]], surf[[2]], surf[[3]],
           col = col.br(150), theta = 225, phi = 20,
           border = FALSE, add.legend = FALSE,
           xlab = "long", ylab = "lat", zlab = "conn_num")
image.plot(zlim = range(surf[[3]], na.rm = TRUE),
           legend.only = TRUE, horizontal = FALSE)
```

Step 7: Variogram and Kriging

```
library(geoR)
library(spBayes)
cnn_portial <- head(num_conn,n=1000)
coords <- as.matrix(cnn_portial[,c("long","lat")])
bins = 50
max.dist <- 0.15*max(iDist(coords))
```

```
num_con.vario <- variog(coords = coords, data = cnn_portial$num_con,
                        uvec = (seq(0, max.dist, length = bins)))

plot(num_con.vario)
eyefit(num_con.vario,silent=TRUE)

sigma <- 8205.92
sill <- 0.02
nugget <- 2495.04


fit.num_con<- variofit(num_con.vario,cov.model="exponential",
                        fix.nugget=FALSE, ini.cov.pars=c(8205.92,0.02),
                        nugget=2495.04)
fit.num_con


point<-krige.conv(coords = coords, data = cnn_portial$num_con,
                  loc=c(length(cnn_portial$num_con),1),
                  krige=krige.control(cov.pars=c(8205.92,0.0811),
                                      cov.model="exponential",
                                      nugget=2495.04))
point

pred_low <-point$predict - 1.96*sqrt(point$krige.var)
pred_high <-point$predict + 1.96*sqrt(point$krige.var)
print(paste("The 95% PI is between",pred_low,"and",pred_high))
```

Step 8: Baysian

```
num_conn <- data %>% group_by(long,lat) %>% summarise(num = n())
num_conn <- head(num_conn,n=500)
coords <- as.matrix(num_conn[,c("long","lat")])

library(spBayes)
n.samples <- 1000

conn_num.sp <- spLM(num ~ 1,
                     data=num_conn, coords=coords,
                    starting=list("phi"=3/1100,"sigma.sq"=8205.92,
                                  "tau.sq"=2495.04),
                    tuning=list("phi"=0.1, "sigma.sq"=0.05,
                                "tau.sq"=0.05),
                    priors=list("phi.Unif"=c(3/1100, 3/50),
                                "sigma.sq.IG"=c(0.1,0.1),
                                "tau.sq.IG"=c(0.1, 0.1)),
                    cov.model="exponential",n.samples=n.samples)
round(summary(mcmc(conn_num.sp$p.theta.samples))$quantiles,3)



burn.in <- floor(0.75*n.samples)
```

```
conn_num.sp <- spRecover(conn_num.sp, start=burn.in, thin=2,verbose = T)
# The posterior samples of the regression coefficients and the spatial effects can then be obtained as
beta.samples = conn_num.sp$p.beta.recover.samples
w.samples = conn_num.sp$p.w.recover.samples

w.hat.mu <- apply(w.samples,1,mean)
w.hat.sd <- apply(w.samples,1,sd)
surf <- mba.surf(cbind(coords, w.hat.mu),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           main="Mean Spatial Effects")
contour(surf,add = T)




library(rgl)
library(RColorBrewer)
col.br <- colorRampPalette(c("blue", "cyan", "yellow", "red"))
col <- rbind(0, cbind(matrix(drape.color(surf[[3]],
                                         col = col.br(25)), x.res - 1, y.res-1), 0))
surface3d(surf[[1]], surf[[2]], surf[[3]], col = abs(col))
axes3d()
title3d(main = "mean", xlab = "long", ylab = "lat", zlab = "conn_num")
drape.plot(surf[[1]], surf[[2]], surf[[3]],
           col = col.br(150), theta = 225, phi = 50,
           border = FALSE, add.legend = FALSE,
           xlab = "long", ylab = "lat", zlab = "num_mean")
image.plot(zlim = range(surf[[3]], na.rm = TRUE),
           legend.only = TRUE, horizontal = FALSE)



surf <- mba.surf(cbind(coords, w.hat.sd),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           main="Residual Spatial Effects")
contour(surf,add = T)




spherical.sp <- spLM(num ~ 1,
                     data=num_conn, coords=coords,
                     starting=list("phi"=3/1100,"sigma.sq"=8205.92,
                                   "tau.sq"=2495.04),
                     tuning=list("phi"=0.1, "sigma.sq"=0.05,
                                 "tau.sq"=0.05),
                     priors=list("phi.Unif"=c(3/1100, 3/50),
                                 "sigma.sq.IG"=c(0.1,0.1),
                                 "tau.sq.IG"=c(0.1, 0.1)),
```

```
                    cov.model="spherical",n.samples=n.samples)
round(summary(mcmc(spherical.sp$p.theta.samples))$quantiles,3)

spherical.sp <- spRecover(spherical.sp, start=burn.in, thin=2,verbose = T)

spherical_dic = spDiag(spherical.sp,start=burn.in,verbose=FALSE)
spherical_dic

exponential_dic = spDiag(conn_num.sp,start=burn.in,verbose=FALSE)
exponential_dic




beta.samples = spherical.sp$p.beta.recover.samples
w.samples = spherical.sp$p.w.recover.samples


w.hat.sd <- apply(w.samples,1,sd)
surf <- mba.surf(cbind(coords, w.hat.sd),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           main="spherical residual")
contour(surf,add = T)
```

Step 9: Spatio-Temporal Bayesian

```
library(dplyr)
library(lubridate)


data <- read.csv("data/clean.csv")

# format time
data <- data %>% mutate(start_time = ymd_hms(start_time)) %>%
  mutate(end_time = ymd_hms(end_time))

# compute connected time
data["con_time"] <- as.numeric(difftime(ymd_hms(data$end_time),
                                        ymd_hms(data$start_time),units = "secs"))




data <- data %>% filter(long>=120.8544 &
                          long<=121.974 &
                          lat >=30.68889 &
                          lat<=31.86765)
```

```r
data <- data %>% mutate(rnd_strat = minute(round_date(start_time, "minute"))) %>%
  mutate(rnd_end = minute(round_date(end_time, "minute")))

num_conn <- data %>% group_by(long,lat,rnd_strat) %>% summarise(num_con = n()) %>% arrange(rnd_strat)
train_data <- data.frame(tail(num_conn,n=200))
test_data <- data.frame(head(num_conn,n=100))
#============================================================
library(spTimer)
post.gp <- spT.Gibbs(formula = num_con ~ 1 ,
                     data = train_data, model = "GP",
                     coords = ~ long + lat,
                     scale.transform = "SQRT",
                     spatial.decay = spT.decay(distribution = Gamm(2,1),
                                               tuning = 0.1))




grid.pred <- predict(post.gp, newdata = test_data, newcoords = ~long + lat)

crdref <- CRS('+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0')
ptx <- SpatialPoints(grid.pred$pred.coords, proj4string=crdref)

pre_mean = as.numeric(grid.pred$Mean)
ptsdf <- SpatialPointsDataFrame(ptx,data.frame(pre_mean))
ptsdf
par(bg=NA)
library(raster)
plot(name_shape$geometry,border="blue", col="gray")

points(ptsdf,pch=20, cex=1, col="red")



library(MBA)
library(fields)

coords <- as.matrix(grid.pred$pred.coords)
x.res <- 200; y.res <- 200
surf <- mba.surf(cbind(coords,
                       pre_mean = as.numeric(grid.pred$Mean)),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="long",
           ylab="lat",
           main="Predict mean" )
points(coords)
contour(surf,add = T)



pred <- data.frame(cbind(coords,as.numeric(grid.pred$Mean)))
```

```r
write.csv(pred,"pred.csv")
```

```r
pred <- read.csv("pred.csv")
coords <- as.matrix(pred[,2:3])
```

```r
surf <- mba.surf(cbind(coords, pred[,4]),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="long",
           ylab="lat",
           main="Predict mean" )
points(coords)
contour(surf,add = T)
```

```r
surface3d(surf[[1]], surf[[2]], surf[[3]], col = abs(col))
axes3d()
title3d(main = "strata", xlab = "long", ylab = "lat", zlab = "conn_time")
drape.plot(surf[[1]], surf[[2]], surf[[3]],
           col = col.br(150), theta = 225, phi = 50,
           border = FALSE, add.legend = FALSE,
           xlab = "long", ylab = "lat", zlab = "predict mean")
image.plot(zlim = range(surf[[3]], na.rm = TRUE),
           legend.only = TRUE, horizontal = FALSE)
```