Land Use regression_OCT

Guide for using Provided R Scripts for : Land use regression model development

For developing land use regression model forst we need is the monitrong station data and other crucial variable which will help explaining air pollution in the city.

1.We need to load the pacakages required for analysis

```
require(sp)
require(raster)
require(rgeos)
require(gstat)
require(rgdal)
require(geoR)
require(maptools)
```

2. We need the monitrong station loaction and data

```
head(monitor.raw.data)
#We will calculate the mean of each for taking yearly average for analysis
monitor.raw.data$annual<-rowMeans(monitor.raw.data@data[,4:15])

#Annual in the data frame
head(monitor.raw.data@data)
```

3.Now we need to extract the GI Variables which can be regressed with Air pollution concentration.

For that we need to load the GI variabels

```
head(major.roads)
head(minor.roads)
head(roads)
head(landuse)
head(buildings)
head(shape_mun)
```

We use these Spatial objects to extracts variable of interest: We are using the variabels used by ESCAPE project: Which require uses buffers of 25m,50m,100m,300m,500m,1000m,5000m for extracting variables

```
Monitor_data<-monitor.raw.data

m.buf_5000_pred<-gBuffer(Monitor_data,width = 5000,byid = TRUE)

m.buf_1000_pred<-gBuffer(Monitor_data,width = 1000,byid = TRUE)

m.buf_500_pred<-gBuffer(Monitor_data,width = 500,byid=TRUE)

m.buf_300_pred<-gBuffer(Monitor_data,width = 300,byid = TRUE)

m.buf_100_pred<-gBuffer(Monitor_data,width = 100,byid=TRUE)

m.buf_50_pred<-gBuffer(Monitor_data,width = 50,byid=TRUE)
```

```
m.buf_25_pred<-gBuffer(Monitor_data,width = 25,byid=TRUE)
```

After getting the Buffer we will extract the variabels we need for LUR development

Building counts in buffer

```
buildcount_5000_pred<-over(m.buf_5000_pred,buildings,fn=length)
buildcount_5000_pred[is.na(buildcount_5000_pred)]<-0
Monitor_data$buildcount_5000<- buildcount_5000_pred$name

buildcount_1000_pred<-over(m.buf_1000_pred,buildings,fn=length)
buildcount_1000_pred[is.na(buildcount_1000_pred)]<-0
Monitor_data$buildcount_1000<- buildcount_1000_pred$name

buildcount_500_pred<-over(m.buf_500_pred,buildings,fn=length)
buildcount_500_pred[is.na(buildcount_500_pred)]<-0
Monitor_data$buildcount_500<- buildcount_500_pred$name


buildcount_300_pred<-over(m.buf_300_pred,buildings,fn=length)
buildcount_300_pred[is.na(buildcount_300_pred)]<-0
Monitor_data$buildcount_300<- buildcount_300_pred$name


buildcount_100_pred<-over(m.buf_100_pred,buildings,fn=length)
buildcount_100_pred[is.na(buildcount_100_pred)]<-0
Monitor_data$buildcount_100<- buildcount_100_pred$name

buildcount_50_pred<-over(m.buf_50_pred,buildings,fn=length)
buildcount_50_pred[is.na(buildcount_50_pred)]<-0
Monitor_data$buildcount_50<- buildcount_50_pred$name

buildcount_25_pred<-over(m.buf_25_pred,buildings,fn=length)
buildcount_25_pred[is.na(buildcount_25_pred)]<-0
Monitor_data$buildcount_25<- buildcount_25_pred$name

Monitor_data$buildcount_50<-Monitor_data$buildcount_50-Monitor_data$buildcount_25
Monitor_data$buildcount_100<-Monitor_data$buildcount_100-Monitor_data$buildcount_50
Monitor_data$buildcount_300<-Monitor_data$buildcount_300-Monitor_data$buildcount_100
Monitor_data$buildcount_500<-Monitor_data$buildcount_500-Monitor_data$buildcount_300
Monitor_data$buildcount_1000<-Monitor_data$buildcount_1000-Monitor_data$buildcount_500
Monitor_data$buildcount_5000<-Monitor_data$buildcount_5000-Monitor_data$buildcount_1000
```

Number of roads in the buffer and there lengths

Function to extract he road information

```
road_4m_buffer<-function(line,buffer){
 result<-data.frame(road.count=double(),road.length=double())
 for(i in 1:length(buffer)){
   intersec<-raster::intersect(line,buffer[i,])
   if(is.null(intersec)){
```

```
      result[i,1]<-0
      result[i,2]<-0
      i=i+1
    }else{
      result[i,1]<-length(intersec)
      result[i,2]<-rgeos::gLength(intersec)
      i=i+1
    }}
  return(result)
}
```

Following command does extraction for all the roads(no classfification)

```
road.count.5000_pred<-road_4m_buffer(roads,m.buf_5000_pred)
road.count.5000_pred[is.na(road.count.5000_pred)]<-0
Monitor_data$rdcount_5000<-road.count.5000_pred$road.count

road.count.1000_pred<-road_4m_buffer(roads,m.buf_1000_pred)
road.count.1000_pred[is.na(road.count.1000_pred)]<-0
Monitor_data$rdcount_1000<-road.count.1000_pred$road.count

road.count.500_pred<-road_4m_buffer(roads,m.buf_500_pred)
road.count.500_pred[is.na(road.count.500_pred)]<-0
Monitor_data$rdcount_500<-road.count.500_pred$road.count

road.count.300_pred<-road_4m_buffer(roads,m.buf_300_pred)
road.count.300_pred[is.na(road.count.300_pred)]<-0
Monitor_data$rdcount_300<-road.count.300_pred$road.count

road.count.100_pred<-road_4m_buffer(roads,m.buf_100_pred)
road.count.100_pred[is.na(road.count.100_pred)]<-0
Monitor_data$rdcount_100<-road.count.100_pred$road.count

road.count.50_pred<-road_4m_buffer(roads,m.buf_50_pred)
road.count.50_pred[is.na(road.count.50_pred)]<-0
Monitor_data$rdcount_50<-road.count.50_pred$road.count

road.count.25_pred<-road_4m_buffer(roads,m.buf_25_pred)
road.count.25_pred[is.na(road.count.25_pred)]<-0
Monitor_data$rdcount_25<-road.count.25_pred$road.count

#count
Monitor_data$rdcount_50<-Monitor_data$rdcount_50-Monitor_data$rdcount_25
Monitor_data$rdcount_100<-Monitor_data$rdcount_100-Monitor_data$rdcount_50
Monitor_data$rdcount_300<-Monitor_data$rdcount_300-Monitor_data$rdcount_100
Monitor_data$rdcount_500<-Monitor_data$rdcount_500-Monitor_data$rdcount_300
Monitor_data$rdcount_1000<-Monitor_data$rdcount_1000-Monitor_data$rdcount_500
Monitor_data$rdcount_5000<-Monitor_data$rdcount_5000-Monitor_data$rdcount_1000
#length
Monitor_data$rdlength_5000<-road.count.5000_pred$road.length
Monitor_data$rdlength_1000<-road.count.1000_pred$road.length
Monitor_data$rdlength_500<-road.count.500_pred$road.length
Monitor_data$rdlength_300<-road.count.300_pred$road.length
Monitor_data$rdlength_100<-road.count.100_pred$road.length
```

```
Monitor_data$rdlength_50<-road.count.50_pred$road.length
Monitor_data$rdlength_25<-road.count.25_pred$road.length
#substracting the inner buffer from out buffer

Monitor_data$rdlength_50<-Monitor_data$rdlength_50-Monitor_data$rdlength_25
Monitor_data$rdlength_100<-Monitor_data$rdlength_100-Monitor_data$rdlength_50
Monitor_data$rdlength_300<-Monitor_data$rdlength_300-Monitor_data$rdlength_100
Monitor_data$rdlength_500<-Monitor_data$rdlength_500-Monitor_data$rdlength_300
Monitor_data$rdlength_1000<-Monitor_data$rdlength_1000-Monitor_data$rdlength_500
Monitor_data$rdlength_5000<-Monitor_data$rdlength_5000-Monitor_data$rdlength_1000
```

Now we will extract information only about Major roads

```
mjroad.count.5000_pred<-road_4m_buffer(major.roads,m.buf_5000_pred)
Monitor_data$mjrdcount_5000<-mjroad.count.5000_pred$road.count

mjroad.count.1000_pred<-road_4m_buffer(major.roads,m.buf_1000_pred)
Monitor_data$mjrdcount_1000<-mjroad.count.1000_pred$road.count

mjroad.count.500_pred<-road_4m_buffer(major.roads,m.buf_500_pred)
Monitor_data$mjrdcount_500<-mjroad.count.500_pred$road.count

mjroad.count.300_pred<-road_4m_buffer(major.roads,m.buf_300_pred)
Monitor_data$mjrdcount_300<-mjroad.count.300_pred$road.count

mjroad.count.100_pred<-road_4m_buffer(major.roads,m.buf_100_pred)
Monitor_data$mjrdcount_100<-mjroad.count.100_pred$road.count

mjroad.count.50_pred<-road_4m_buffer(major.roads,m.buf_50_pred)
Monitor_data$mjrdcount_50<-mjroad.count.50_pred$road.count

mjroad.count.25_pred<-road_4m_buffer(major.roads,m.buf_25_pred)
Monitor_data$mjrdcount_25<-mjroad.count.25_pred$road.count

Monitor_data$mjrdcount_50<-Monitor_data$mjrdcount_50-Monitor_data$mjrdcount_25
Monitor_data$mjrdcount_100<-Monitor_data$mjrdcount_100-Monitor_data$mjrdcount_50
Monitor_data$mjrdcount_300<-Monitor_data$mjrdcount_300-Monitor_data$mjrdcount_100
Monitor_data$mjrdcount_500<-Monitor_data$mjrdcount_500-Monitor_data$mjrdcount_300
Monitor_data$mjrdcount_1000<-Monitor_data$mjrdcount_1000-Monitor_data$mjrdcount_500
Monitor_data$mjrdcount_5000<-Monitor_data$mjrdcount_5000-Monitor_data$mjrdcount_1000
#moajor road length
Monitor_data$mjrdlength_5000<-mjroad.count.5000_pred$road.length
Monitor_data$mjrdlength_1000<-mjroad.count.1000_pred$road.length
Monitor_data$mjrdlength_500<-mjroad.count.500_pred$road.length
Monitor_data$mjrdlength_300<-mjroad.count.300_pred$road.length
Monitor_data$mjrdlength_100<-mjroad.count.100_pred$road.length
Monitor_data$mjrdlength_50<-mjroad.count.50_pred$road.length
Monitor_data$mjrdlength_25<-mjroad.count.25_pred$road.length



Monitor_data$mjrdlength_50<-Monitor_data$mjrdlength_50-Monitor_data$mjrdlength_25
Monitor_data$mjrdlength_100<-Monitor_data$mjrdlength_100-Monitor_data$mjrdlength_50
Monitor_data$mjrdlength_300<-Monitor_data$mjrdlength_300-Monitor_data$mjrdlength_100
```

```
Monitor_data$mjrdlength_500<-Monitor_data$mjrdlength_500-Monitor_data$mjrdlength_300
Monitor_data$mjrdlength_1000<-Monitor_data$mjrdlength_1000-Monitor_data$mjrdlength_500
Monitor_data$mjrdlength_5000<-Monitor_data$mjrdlength_5000-Monitor_data$mjrdlength_1000
```

Now we will extract the data about minor roads

```
min.road.count.5000_pred<-road_4m_buffer.sf(minor.roads,m.buf_5000_pred)
Monitor_data$minrdcount_5000<-min.road.count.5000_pred$road.count

min.road.count.1000_pred<-road_4m_buffer.sf(minor.roads,m.buf_1000_pred)
Monitor_data$minrdcount_1000<-min.road.count.1000_pred$road.count

min.road.count.500_pred<-road_4m_buffer(minor.roads,m.buf_500_pred)
Monitor_data$minrdcount_500<-min.road.count.500_pred$road.count

min.road.count.300_pred<-road_4m_buffer(minor.roads,m.buf_300_pred)
Monitor_data$minrdcount_300<-min.road.count.300_pred$road.count

min.road.count.100_pred<-road_4m_buffer(minor.roads,m.buf_100_pred)
Monitor_data$minrdcount_100<-min.road.count.100_pred$road.count

min.road.count.50_pred<-road_4m_buffer(minor.roads,m.buf_50_pred)
Monitor_data$minrdcount_50<-min.road.count.50_pred$road.count

min.road.count.25_pred<-road_4m_buffer(minor.roads,m.buf_25_pred)
Monitor_data$minrdcount_25<-min.road.count.25_pred$road.count



Monitor_data$minrdcount_50<-Monitor_data$minrdcount_50-Monitor_data$minrdcount_25
Monitor_data$minrdcount_100<-Monitor_data$minrdcount_100-Monitor_data$minrdcount_50
Monitor_data$minrdcount_300<-Monitor_data$minrdcount_300-Monitor_data$minrdcount_100
Monitor_data$minrdcount_500<-Monitor_data$minrdcount_500-Monitor_data$minrdcount_300
Monitor_data$minrdcount_1000<-Monitor_data$minrdcount_1000-Monitor_data$minrdcount_500
Monitor_data$minrdcount_5000<-Monitor_data$minrdcount_5000-Monitor_data$minrdcount_1000
#road length
Monitor_data$minrdlength_5000<-min.road.count.5000_pred.sf$road.length
Monitor_data$minrdlength_1000<-min.road.count.1000_pred$road.length
Monitor_data$minrdlength_500<-min.road.count.500_pred$road.length
Monitor_data$minrdlength_300<-min.road.count.300_pred$road.length
Monitor_data$minrdlength_100<-min.road.count.100_pred$road.length
Monitor_data$minrdlength_50<-min.road.count.50_pred$road.length
Monitor_data$minrdlength_25<-min.road.count.25_pred$road.length
#substracting inner buffer values from outer buffer

Monitor_data$minrdlength_50<-Monitor_data$minrdlength_50-Monitor_data$minrdlength_25
Monitor_data$minrdlength_100<-Monitor_data$minrdlength_100-Monitor_data$minrdlength_50
Monitor_data$minrdlength_300<-Monitor_data$minrdlength_300-Monitor_data$minrdlength_100
Monitor_data$minrdlength_500<-Monitor_data$minrdlength_500-Monitor_data$minrdlength_300
Monitor_data$minrdlength_1000<-Monitor_data$minrdlength_1000-Monitor_data$minrdlength_500
Monitor_data$minrdlength_5000<-Monitor_data$minrdlength_5000-Monitor_data$minrdlength_1000
```

We also need to extract the information about the distance of monitoring stations to roads

Function to calculate the distance to road

```
distance2road<-function(rd,pt){
  distan<-data.frame(distance=double())
  for(i in 1:length(pt)){
    distan[i,]<-rgeos::gDistance(rd,pt[i,])
    i=i+1
  }
  return(distan)
}
```

Using this Function we will extract the values now

```
#for all roads
dist_rd_pred<-distance2road(roads,Monitor_data)
Monitor_data$dist.rd<- dist_rd_pred$distance


#for major roads
dist_mjrd_pred<-distance2road(major.roads,Monitor_data)
Monitor_data$dist.mjrd<- dist_mjrd_pred$distance

# for minor roads
dist_minrd_pred<-distance2road(minor.roads,Monitor_data)
Monitor_data$dist.minrd<- dist_minrd_pred$distance
```

We also need the information about the traffic congestion on roads in buffer

Function below extract the congestion information from the inputted road network

```
congestion_nearest_road<-function(rd,pt){
  cong<-data.frame(congestion=double())
  for(i in 1:length(pt)){
    dist2road<-gDistance(rd,pt[i,])
    for ( j in 1 : length(rd))
    {
      if (gWithinDistance(rd[j,],pt[i,], dist2road))
        rownumber<-j
    }
    cong[i,1]<-rd[rownumber,]$congestion
    i=i+1
  }
  return(cong)
}
```

Using this function we extract the values for major and minor roads

```
#calculate the congestiona and put it into Monitor_data layer
congest_dat_pred<-congestion_nearest_road(major.roads,Monitor_data)
Monitor_data$congestion_mjrd<-congest_dat_pred$congestion

congest_dat<-congestion_nearest_road(minor.roads,Monitor_data)
Monitor_data$congestion_minrd<-congest_dat$congestion
```

We will also need data for predicting values at unmonitored places ,so we will extract the vales at unmeasured sample points

```
sample.points<-spsample(shape_mun,600,"regular")
```

We use the same function as before to extract all the variables we extracted for Monitoring stations

Now we will use these variable to perform Regression

```
LM.all_above<-lm(annual~buildcount_5000+buildcount_1000+buildcount_500+ buildcount_300+buildcount_100+bu
```

There are various ways to select the variabels for regression. We choose the backward selection procedure to select the final regressors for the further process .We developed the following regression equation.

```
lm_monitor_2<-lm(annual~buildcount_125+rdcount_1000+rdcount_500+rdlength_500+mjrdlength_500+mjrdlength_1
```

```
summary(lm_monitor_2)
```

We will predict the values for unsamoled location using this model and data generated for all locations of city (sample.point).

```
prediction.LM.muenster<-predict(lm_monitor_2,newdata=sample.points)
sample.points<-spTransform(sample.points,"+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs ")
prediciton.df<-data.frame(sample.points@coords,prediction.LM.muenster)
```

Plotting the result

```
coordinates(prediciton.df)<-~x1+x2
proj4string(prediciton.df)<-proj4string(shape_mun)
class(prediciton.df)
#Creating the grid of the object
gridded(prediciton.df) <- TRUE
plot(prediciton.df,main="Predicted map using simple regression")
plot(roads,add=T)
plot(major.roads,add=T)
```

Now if we want to check for the Spatial relation we will use the residuals and plot them for creatign variogram

```
residual.df=data.frame(sample.points@coords,res=rstudent(lm_monitor_2))
geo.residual=as.geodata(residual.df,coords.col=1:2,data.col=3)
plot(geo.residual)
```

Now we can prepare the variogram:

```
var1=variog(geo1,estimator.type = 'modulus')
var_clas<-variog(geo.residual)
plot(var1)
```

fitting the model

```
ev=eyefit(var1)
mod_vg=as.vgm.variomodel(ev[[1]])
```

```
pred_rk=krige( annual ~ buildcount_125 + rdcount_1000 + rdcount_500 +
    rdlength_500 + mjrdlength_500 + mjrdlength_250 + minrdlength_250 +
    congestion_mjrd,Monitor_data,newdata=sample.points,model=mod_vg)

plot(pred_rk)
```