Boosting_InversJarak_KNN_PDRB

Anandika Dian P.

3/29/2020

```
#KNN dengan range disesuaikan
#Inverse distance dengan power [0,4:4] dengan selisih 0,1
#Matriks PDRB selisih invers terstandardisasi (nonsimetris)
```

```
start.time <- Sys.time()
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.6.1, GDAL 2.2.3, PROJ 4.9.3
```

```
dataIGI <- read.csv("D:/Skripsweet/data_jateng_w_latlong.csv")
rownames(dataIGI) <- dataIGI$KABUPATEN
attach(dataIGI)
vy <- IGI
vx <- dataIGI[,4:8]
coords <- cbind(x_long,y_lat)
detach(dataIGI)
```

```
PDRB <- read.csv("D:/Skripsweet/Kak Along's/matriks_PDRB_Berlaku_inverse.csv")
w_PDRB <- as.matrix(PDRB)
```

```
get.list=function(coords,k,indist)
{
  #mengambil k tetangga terdekat
  nb <- knn2nb(knearneigh(coords, k=k),sym=T )
  #menentukan jarak ke masing2 tetangga
  jarak <- nbdists(nb, coords,longlat = TRUE)
  #menerapkan inverse distance
  jarak <- lapply(jarak,indist)
  #membuat list dengan matriks ketetanggaan jarak inverse
  w_jarak <- nb2listw(nb, glist=jarak, style="B", zero.policy=T)
  #mengembalikan ke bentuk matriks
  w_jarak <- listw2mat(w_jarak)
  #mengalikan matriks knn dengan PDRB
  m_cust <- w_PDRB*w_jarak
```

```
  m_cust_tot<-rowSums(m_cust,na.rm = TRUE)
  m_custom<-m_cust/m_cust_tot
  #mengubah ke bentuk listw
  w_custom <- mat2listw(m_custom)
}
```

```
create.instr=function(vy,vx,w_custom){
  ly <- lag.listw(w_custom,vy)
  res <- matrix(data = NA, nrow = nrow(vx), ncol = ncol(vx))
  for (i in 1:ncol(vx))
  {
    res[ ,i]=  lag.listw(w_custom,vx[,i])
  }
  instr=lm(ly~ res)$fitted.values
  instr
}
```

```
#sekuens untuk power invers distance
degs <- seq(from = 0.4, to = 4,by=0.1)

#membuat list berisi sequence
funs <- as.list(rep(NA,length(degs)))

#fungsi jarak invers dengan power tertentu
for(i in 1:length(degs))
{
  funs[[i]]=function(x) 1/(x^degs[i])
}

res<-matrix(NA,length(vy),length(funs))
X <- vx

library(rgdal)
```

```
## rgdal: version: 1.4-3, (SVN revision 828)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Program Files/R/R-3.6.2/library/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Program Files/R/R-3.6.2/library/rgdal/proj
##  Linking to sp version: 1.3-1
```

```
#mencari minimum dan maksimum tetangga
jt_lp<-readOGR("D:/KULIAH STIS/Tingkat 3/Semester 6/SIG/UAS SIG","Jawa_Tengah")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\KULIAH STIS\Tingkat 3\Semester 6\SIG\UAS SIG", layer: "Jawa_Tengah"
## with 35 features
## It has 6 fields
```

```
library(rgeos)
```

```
## rgeos version: 0.4-3, (SVN revision 595)
##  GEOS runtime version: 3.6.1-CAPI-1.10.1
##  Linking to sp version: 1.3-1
##  Polygon checking: TRUE
```

```
list.nb <- gTouches(jt_lp, byid = TRUE, returnDense = FALSE)
nn<-as.list(rep(NA,length(list.nb)))
for(i in 1:length(list.nb))
{
  nn[i]<-NROW(list.nb[[i]])
}
nn_max<-max(as.numeric(unlist(nn)))
nn_min<-min(as.numeric(unlist(nn)))
```

```
for(J in nn_min:nn_max)
{
  for(i in 1:length(funs))
  {
    #a1 sebagai fungsi invers jarak
    a1 <- funs[[i]]
    #a untuk membentuk matriks pembobot
    a <- get.list(coords,k=J,a1)
    #menghitung fitted values model pada masing2 matriks pembobot
    b <- create.instr(vy,vx,a)
    names(b)=NULL
    res[,i]=b
  }
  mn <- paste("s3_n",J,"w",degs,sep="")
  res <- data.frame(res)
  names(res) <- mn
  res <- as.matrix(res)
  X <- cbind(X,res)
  #X <- res
}
```

```
X1<-X[,-(1:5)]
n = nrow(X1)
#jumlah kombinasi matriks pembobot
#length(funs)*nn_max=37*8=296
#X berisi matriks dengan baris berupa wilayah dan kolom berupa macam2 matriks pembobot dengan elemen ma

save(X1,vy,n, file="d:/Skripsweet/Bimbingan 10/IGI_jateng.w3.Rdata")

load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w3.Rdata")
jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w3.Rdata"))
jat_w_mat<-as.matrix(jateng_w)
```

```
library(mboost)
```

```
## Loading required package: parallel
```

```
## Loading required package: stabs

## This is mboost 2.9-1. See 'package?mboost' and 'news(package  = "mboost")'
## for a complete list of changes.
```

```
#v 0.1
m1=glmboost(jat_w_mat,vy,control = boost_control(mstop = 1000,nu=0.1),center = FALSE)
```

```
aic1 <- AIC(m1,method="corrected")
aic1
```

```
## [1] -0.5637593
## Optimal number of boosting iterations: 1
## Degrees of freedom (for mstop = 1): 0.1
```

```
mbest1aic=m1[mstop(aic1)]
names(coef(mbest1aic)[abs(coef(mbest1aic)) > 0])
```

```
## [1] "s3_n8w4"
```

```
gMDL1 <- AIC(m1,method="gMDL")
gMDL1
```

```
## [1] -1.596627
## Optimal number of boosting iterations: 1
## Degrees of freedom (for mstop = 1): 0.1
```

```
mbest1gMDL=m1[mstop(gMDL1)]
names(coef(mbest1gMDL)[abs(coef(mbest1gMDL)) > 0])
```

```
## [1] "s3_n8w4"
```

```
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

```
## Time difference of 12.47258 secs
```

```
#bobot terpilih, k=8
knn <- knearneigh(coords,k = 8)
knn_nb <- knn2nb(knn)
dlist <- nbdists(knn_nb, coords,longlat = TRUE)
indis <- function(x) 1/(x^4)
dlist <- lapply(dlist, indis)
bobot1 <- nb2listw(knn_nb, glist=dlist, style="B", zero.policy=T)
#mengembalikan ke bentuk matriks
bobot1_mat <- listw2mat(bobot1)
#mengalikan matriks knn dengan PDRB
bobot_custom <- w_PDRB*bobot1_mat
b_cust_tot<-rowSums(bobot_custom,na.rm = TRUE)
bobot_custom<-bobot_custom/b_cust_tot
#mengubah ke bentuk listw
w_final <- mat2listw(bobot_custom,style = "W")
```

```r
moran.test(dataIGI$IGI,w_final,randomisation = TRUE)
```

```
##
##  Moran I test under randomisation
##
## data:  dataIGI$IGI
## weights: w_final
##
## Moran I statistic standard deviate = 2.3929, p-value = 0.008359
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.34240369       -0.02941176        0.02414478
```

```r
my<- dataIGI$IGI
mx<-cbind(dataIGI$INFLASI,dataIGI$PMTB,dataIGI$UMK,dataIGI$PP,dataIGI$PPS)
ols<-lm(my~mx)
moran.test(ols$residuals,w_final,randomisation = TRUE)
```

```
##
##  Moran I test under randomisation
##
## data:  ols$residuals
## weights: w_final
##
## Moran I statistic standard deviate = 0.051792, p-value = 0.4793
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       -0.02146149       -0.02941176        0.02356323
```

```r
r.ols <- lm(dataIGI$IGI ~ dataIGI$INFLASI+dataIGI$PMTB+dataIGI$UMK+dataIGI$PPS+dataIGI$PP)
r.lag <- lagsarlm(r.ols, dataIGI, w_final, zero.policy = TRUE)
```

```
## Warning: Function lagsarlm moved to the spatialreg package
```

```
## Registered S3 methods overwritten by 'spatialreg':
##   method                   from
##   residuals.stsls          spdep
##   deviance.stsls           spdep
##   coef.stsls               spdep
##   print.stsls              spdep
##   summary.stsls            spdep
##   print.summary.stsls      spdep
##   residuals.gmsar          spdep
##   deviance.gmsar           spdep
##   coef.gmsar               spdep
##   fitted.gmsar             spdep
##   print.gmsar              spdep
##   summary.gmsar            spdep
##   print.summary.gmsar      spdep
```

```
##   print.lagmess            spdep
##   summary.lagmess          spdep
##   print.summary.lagmess    spdep
##   residuals.lagmess        spdep
##   deviance.lagmess         spdep
##   coef.lagmess             spdep
##   fitted.lagmess           spdep
##   logLik.lagmess           spdep
##   fitted.SFResult          spdep
##   print.SFResult           spdep
##   fitted.ME_res            spdep
##   print.ME_res             spdep
##   print.lagImpact          spdep
##   plot.lagImpact           spdep
##   summary.lagImpact        spdep
##   HPDinterval.lagImpact    spdep
##   print.summary.lagImpact  spdep
##   print.sarlm              spdep
##   summary.sarlm            spdep
##   residuals.sarlm          spdep
##   deviance.sarlm           spdep
##   coef.sarlm               spdep
##   vcov.sarlm               spdep
##   fitted.sarlm             spdep
##   logLik.sarlm             spdep
##   anova.sarlm              spdep
##   predict.sarlm            spdep
##   print.summary.sarlm      spdep
##   print.sarlm.pred         spdep
##   as.data.frame.sarlm.pred spdep
##   residuals.spautolm       spdep
##   deviance.spautolm        spdep
##   coef.spautolm            spdep
##   fitted.spautolm          spdep
##   print.spautolm           spdep
##   summary.spautolm         spdep
##   logLik.spautolm          spdep
##   print.summary.spautolm   spdep
##   print.WXImpact           spdep
##   summary.WXImpact         spdep
##   print.summary.WXImpact   spdep
##   predict.SLX              spdep
```

```
summary(r.lag, Nagelkerke = TRUE)
```

```
##
## Call:spatialreg::lagsarlm(formula = formula, data = data, listw = listw,
##     na.action = na.action, Durbin = Durbin, type = type, method = method,
##     quiet = quiet, zero.policy = zero.policy, interval = interval,
##     tol.solve = tol.solve, trs = trs, control = control)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.933307 -0.147453  0.024772  0.203708  0.536011
```

```
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       5.5944463  1.0238328  5.4642 4.65e-08
## dataIGI$INFLASI -0.1986595  0.1216521 -1.6330 0.102466
## dataIGI$PMTB     0.0092599  0.0046367  1.9971 0.045814
## dataIGI$UMK     -0.0889950  0.0382273 -2.3280 0.019910
## dataIGI$PPS      0.3529068  0.1113202  3.1702 0.001523
## dataIGI$PP      -0.0050879  0.0062839 -0.8097 0.418124
##
## Rho: 0.26011, LR test value: 2.6462, p-value: 0.1038
## Asymptotic standard error: 0.13341
##     z-value: 1.9497, p-value: 0.051217
## Wald statistic: 3.8012, p-value: 0.051217
##
## Log likelihood: -7.584041 for lag model
## ML residual variance (sigma squared): 0.087801, (sigma: 0.29631)
## Nagelkerke pseudo-R-squared: 0.53884
## Number of observations: 35
## Number of parameters estimated: 8
## AIC: 31.168, (AIC for lm: 31.814)
## LM test for residual autocorrelation
## test value: 5.7541, p-value: 0.016451
```

```
W <- as(w_final, "CsparseMatrix")
trMatc <- trW(W, type = "mult")
```

```
## Warning: Function trW moved to the spatialreg package
```

```
set.seed(1)
summary(impacts(r.lag, tr=trMatc, R = 99), zstats = TRUE, short = TRUE)
```

```
## Warning: Method impacts.sarlm moved to the spatialreg package

## Impact measures (lag, trace):
##                        Direct     Indirect        Total
## dataIGI$INFLASI -0.204440238 -0.064057620 -0.268497858
## dataIGI$PMTB     0.009529327  0.002985841  0.012515168
## dataIGI$UMK     -0.091584653 -0.028696381 -0.120281034
## dataIGI$PPS      0.363175986  0.113794572  0.476970557
## dataIGI$PP      -0.005235977 -0.001640598 -0.006876575
## ========================================================
## Simulation results (asymptotic variance matrix):
## ========================================================
## Simulated standard errors
##                      Direct    Indirect       Total
## dataIGI$INFLASI 0.109524989 0.054887310 0.143844532
## dataIGI$PMTB    0.004701507 0.002320001 0.006302261
## dataIGI$UMK     0.038785803 0.026652524 0.060521376
## dataIGI$PPS     0.121362531 0.084807490 0.180927815
## dataIGI$PP      0.006654762 0.002910375 0.009290605
```

```
##
## Simulated z-values:
##                      Direct   Indirect     Total
## dataIGI$INFLASI -1.9405062 -1.1487976 -1.915876
## dataIGI$PMTB     1.9479922  1.2066369  1.897398
## dataIGI$UMK     -2.4223549 -1.1548674 -2.060976
## dataIGI$PPS      3.0450505  1.3690129  2.684262
## dataIGI$PP      -0.6645651 -0.5289538 -0.641721
##
## Simulated p-values:
##                  Direct    Indirect Total
## dataIGI$INFLASI 0.0523182 0.25064  0.055381
## dataIGI$PMTB    0.0514159 0.22757  0.057775
## dataIGI$UMK     0.0154203 0.24814  0.039305
## dataIGI$PPS     0.0023264 0.17100  0.007269
## dataIGI$PP      0.5063287 0.59684  0.521054
```