Boosting_InversJarak_Murni_Migrasi_2

Anandika Dian P.

3/29/2020

```
#Inverse distance dengan power [0,4:4] dengan selisih 0,1
#Matriks migrasi selisih 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)
```

```
migrasi <- read.csv("D:/Skripsweet/Kak Along's/migrasi_selisih.csv")
w_migrasi <- as.matrix(migrasi)
```

```
k<-dim(dataIGI)[1]-1
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 migrasi
  m_cust <- w_migrasi*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, style = "W")
}
```

```
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
```

```
for(i in 1:length(funs))
  {
  #a1 sebagai fungsi invers jarak
  a1 <- funs[[i]]
  #a untuk membentuk matriks pembobot
  a <- get.list(coords,k=k,a1)
  #menghitung fitted values model pada masing2 matriks pembobot
  b <- create.instr(vy,vx,a)
  names(b)=NULL
  res[,i]=b
}
```

```
## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points
```

```
## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points
```

```
## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points

## Warning in knearneigh(coords, k = k): k greater than one-third of the
## number of data points
```

```r
mn <- paste("s4_w",degs,sep="")
res <- data.frame(res)
names(res) <- mn
res <- as.matrix(res)
X <- cbind(X,res)
#X <- res
```

```r
X1<-X[,-(1:5)]
n = nrow(X1)
#jumlah kombinasi matriks pembobot
#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.w4_1.Rdata")

#load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w4_1.Rdata")
#jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w4_1.Rdata"))
save(X1,vy,n, file="d:/Skripsweet/Bimbingan 10/IGI_jateng.w4.Rdata")
jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w4.Rdata"))
#jat_w_mat<-as.matrix(X1)
jat_w_mat<-as.matrix(jateng_w)
```

```r
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.
```

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

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

```
## [1] -0.5637562
## Optimal number of boosting iterations: 1
## Degrees of freedom (for mstop = 1): 0.1
```

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

```
## [1] "s4_w3"
```

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

```
## [1] -1.596624
## Optimal number of boosting iterations: 1
## Degrees of freedom (for mstop = 1): 0.1
```

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

```
## [1] "s4_w3"
```

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

```
## Time difference of 3.212419 secs
```

```
k<-dim(dataIGI)[1]-1
knn <- knearneigh(coords,k)
```

```
## Warning in knearneigh(coords, k): k greater than one-third of the number of
## data points
```

```
knn_nb <- knn2nb(knn)
dlist <- nbdists(knn_nb, coords,longlat = TRUE)
indis <- function(x) 1/(x^3)
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 migrasi
bobot_custom <- w_migrasi*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")
```

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

```
##
##  Moran I test under randomisation
##
## data:  dataIGI$IGI
## weights: w_final
##
## Moran I statistic standard deviate = 3.2661, p-value = 0.0005452
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.43941531       -0.02941176        0.02060444
```

```
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 = 1.3259, p-value = 0.09244
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.15860734       -0.02941176        0.02010898
```

```
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
## -0.8658710 -0.1191646  0.0070093  0.1955460  0.5445156
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                   Estimate Std. Error z value  Pr(>|z|)
## (Intercept)      4.5358425  0.9107421  4.9804 6.346e-07
## dataIGI$INFLASI -0.1461108  0.1141262 -1.2803  0.200455
## dataIGI$PMTB     0.0099049  0.0041804  2.3693  0.017820
## dataIGI$UMK     -0.0860064  0.0349469 -2.4611  0.013853
## dataIGI$PPS      0.2990775  0.1021182  2.9287  0.003403
## dataIGI$PP      -0.0069698  0.0058334 -1.1948  0.232155
##
## Rho: 0.42439, LR test value: 7.6426, p-value: 0.0057004
## Asymptotic standard error: 0.11551
##     z-value: 3.6742, p-value: 0.00023859
## Wald statistic: 13.5, p-value: 0.00023859
##
## Log likelihood: -5.085806 for lag model
## ML residual variance (sigma squared): 0.073032, (sigma: 0.27024)
## Nagelkerke pseudo-R-squared: 0.60019
## Number of observations: 35
## Number of parameters estimated: 8
## AIC: 26.172, (AIC for lm: 31.814)
## LM test for residual autocorrelation
## test value: 0.997, p-value: 0.31804
```

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

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

```r
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$INFLASI -0.157208191 -0.096628920 -0.25383711
## dataIGI$PMTB     0.010657162  0.006550486  0.01720765
## dataIGI$UMK     -0.092538800 -0.056879506 -0.14941831
## dataIGI$PPS      0.321793013  0.197791929  0.51958494
## dataIGI$PP      -0.007499213 -0.004609434 -0.01210865
## ========================================================
## Simulation results (asymptotic variance matrix):
## ========================================================
## Simulated standard errors
##                       Direct    Indirect       Total
## dataIGI$INFLASI 0.121256344 0.078643881 0.190895559
## dataIGI$PMTB    0.004676086 0.003701302 0.007738363
## dataIGI$UMK     0.041669861 0.041404033 0.078917243
## dataIGI$PPS     0.104106581 0.111181108 0.199512342
## dataIGI$PP      0.006101593 0.005054323 0.010829960
##
## Simulated z-values:
##                     Direct   Indirect     Total
## dataIGI$INFLASI -1.381129 -1.2199278 -1.379867
## dataIGI$PMTB     2.154732  1.6151147  2.074565
## dataIGI$UMK     -2.219327 -1.4344074 -1.924412
## dataIGI$PPS      3.067633  1.8128934  2.610968
## dataIGI$PP      -1.180352 -0.9882365 -1.126218
##
## Simulated p-values:
##                 Direct    Indirect Total
## dataIGI$INFLASI 0.1672394 0.222492 0.1676276
## dataIGI$PMTB    0.0311828 0.106286 0.0380268
## dataIGI$UMK     0.0264645 0.151456 0.0543029
## dataIGI$PPS     0.0021576 0.069848 0.0090286
## dataIGI$PP      0.2378604 0.323037 0.2600734
```