Boosting_InversJarak_Murni_TPT_2

Anandika Dian P.

3/29/2020

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

```
tpt <- read.csv("D:/Skripsweet/Kak Along's/Matriks_TPT_inv.csv")
w_tpt <- as.matrix(tpt)
```

```
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 tpt
  m_cust <- w_tpt*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
```

```
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("s5_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.w5.Rdata")

#load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w5.Rdata")
#jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w5.Rdata"))

save(X1,vy,n, file="d:/Skripsweet/Bimbingan 10/IGI_jateng.w5.Rdata")
jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w5.Rdata"))

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] -0.5637312
## 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] "s5_w4"
```

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

```
## [1] -1.596599
## 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] "s5_w4"
```

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

```
## Time difference of 3.134654 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^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 tpt
bobot_custom <- w_tpt*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 = 2.9856, p-value = 0.001415
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.39834045       -0.02941176        0.02052642
```

```
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 = 1.4948, p-value = 0.06748
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.18214140       -0.02941176        0.02002986
```

```
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
## -0.8582185 -0.1476460 -0.0071843  0.1955885  0.5149014
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                   Estimate Std. Error z value  Pr(>|z|)
## (Intercept)      4.4391431  0.8853538  5.0140 5.332e-07
## dataIGI$INFLASI -0.1462537  0.1094515 -1.3362  0.181470
## dataIGI$PMTB     0.0106106  0.0041389  2.5637  0.010358
## dataIGI$UMK     -0.0909329  0.0343766 -2.6452  0.008164
## dataIGI$PPS      0.3166146  0.1001511  3.1614  0.001570
## dataIGI$PP      -0.0058473  0.0056182 -1.0408  0.297980
##
## Rho: 0.44348, LR test value: 8.2187, p-value: 0.004146
## Asymptotic standard error: 0.10976
##     z-value: 4.0404, p-value: 5.3349e-05
## Wald statistic: 16.325, p-value: 5.3349e-05
##
## Log likelihood: -4.797758 for lag model
## ML residual variance (sigma squared): 0.071111, (sigma: 0.26667)
## Nagelkerke pseudo-R-squared: 0.60672
## Number of observations: 35
## Number of parameters estimated: 8
## AIC: 25.596, (AIC for lm: 31.814)
## LM test for residual autocorrelation
## test value: 0.079809, p-value: 0.77756
```

```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$INFLASI -0.159149791 -0.103650311 -0.26280010
## dataIGI$PMTB     0.011546193  0.007519749  0.01906594
## dataIGI$UMK     -0.098950984 -0.064444321 -0.16339530
## dataIGI$PPS      0.344532379  0.224385392  0.56891777
## dataIGI$PP      -0.006362862 -0.004143974 -0.01050684
## ========================================================
## Simulation results (asymptotic variance matrix):
## ========================================================
## Simulated standard errors
##                       Direct    Indirect       Total
## dataIGI$INFLASI 0.126121633 0.099809852 0.218242959
## dataIGI$PMTB    0.004572345 0.004265987 0.008175879
## dataIGI$UMK     0.038512756 0.043068453 0.077304008
## dataIGI$PPS     0.097321182 0.110083473 0.186847438
## dataIGI$PP      0.006035044 0.005041982 0.010833620
##
## Simulated z-values:
##                      Direct   Indirect      Total
## dataIGI$INFLASI -1.2807075 -1.0223592 -1.2076744
## dataIGI$PMTB     2.4560750  1.7325662  2.2775687
## dataIGI$UMK     -2.6484092 -1.6022953 -2.2121223
## dataIGI$PPS      3.5692047  2.0599633  3.0727054
## dataIGI$PP      -0.9468973 -0.7915223 -0.8958601
##
## Simulated p-values:
##                 Direct     Indirect Total
## dataIGI$INFLASI 0.20029642 0.306611 0.2271725
## dataIGI$PMTB    0.01404638 0.083173 0.0227523
## dataIGI$UMK     0.00808716 0.109090 0.0269582
## dataIGI$PPS     0.00035807 0.039402 0.0021213
## dataIGI$PP      0.34369113 0.428639 0.3703275
```