Boosting_InversJarak_KNN_TPT

Anandika Dian P.

3/29/2020

```
#KNN dengan range disesuaikan
#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)
```

```
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

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("s2_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.w2.Rdata")

load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w2.Rdata")
jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w2.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] -0.5637477
## 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] "s2_n6w0.4"
```

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

```
## [1] -1.596616
## 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] "s2_n6w0.4"
```

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

```
## Time difference of 18.82505 secs
```

```
#bobot terpilih, k=8
knn <- knearneigh(coords,k = 6)
knn_nb <- knn2nb(knn)
dlist <- nbdists(knn_nb, coords,longlat = TRUE)
indis <- function(x) 1/(x^0.4)
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")
```

```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 = 3.3457, p-value = 0.0004104
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.34950777       -0.02941176        0.01282700
```

```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 = 1.8775, p-value = 0.03022
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.18064466       -0.02941176        0.01251699
```

```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
## -0.8738658 -0.1309101 -0.0074467  0.1774446  0.5096666
```

```
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                    Estimate Std. Error z value  Pr(>|z|)
## (Intercept)      4.2516903  1.0199961  4.1683 3.068e-05
## dataIGI$INFLASI -0.1099451  0.1132599 -0.9707   0.33168
## dataIGI$PMTB     0.0115268  0.0042300  2.7250   0.00643
## dataIGI$UMK     -0.0945032  0.0352550 -2.6806   0.00735
## dataIGI$PPS      0.3245682  0.1026646  3.1614   0.00157
## dataIGI$PP      -0.0057974  0.0056997 -1.0171   0.30908
##
## Rho: 0.46272, LR test value: 7.2683, p-value: 0.0070182
## Asymptotic standard error: 0.13567
##     z-value: 3.4106, p-value: 0.00064827
## Wald statistic: 11.632, p-value: 0.00064827
##
## Log likelihood: -5.272977 for lag model
## ML residual variance (sigma squared): 0.074626, (sigma: 0.27318)
## Nagelkerke pseudo-R-squared: 0.59589
## Number of observations: 35
## Number of parameters estimated: 8
## AIC: 26.546, (AIC for lm: 31.814)
## LM test for residual autocorrelation
## test value: 0.13185, p-value: 0.71652
```

```
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$INFLASI -0.117220950 -0.087413570 -0.20463452
## dataIGI$PMTB     0.012289583  0.009164542  0.02145413
## dataIGI$UMK     -0.100757130 -0.075136232 -0.17589336
## dataIGI$PPS      0.346047077  0.258052938  0.60410002
## dataIGI$PP      -0.006181086 -0.004609337 -0.01079042
## ========================================================
## Simulation results (asymptotic variance matrix):
## ========================================================
## Simulated standard errors
##                       Direct    Indirect       Total
## dataIGI$INFLASI 0.126847704 0.120988311 0.238559752
## dataIGI$PMTB    0.004569323 0.005878009 0.009358321
## dataIGI$UMK     0.035724960 0.056961294 0.085943172
## dataIGI$PPS     0.098400488 0.159373205 0.228858398
## dataIGI$PP      0.006305897 0.006352849 0.012165229
```

```
##
## Simulated z-values:
##                      Direct   Indirect      Total
## dataIGI$INFLASI -0.9473901 -0.7068576 -0.8622400
## dataIGI$PMTB     2.5520873  1.4896678  2.1817581
## dataIGI$UMK     -2.8576392 -1.4189451 -2.1283133
## dataIGI$PPS      3.5491887  1.6870472  2.7008491
## dataIGI$PP      -0.9381524 -0.8202284 -0.9146296
##
## Simulated p-values:
##                 Direct     Indirect Total
## dataIGI$INFLASI 0.34344006 0.479655 0.3885555
## dataIGI$PMTB    0.01070797 0.136312 0.0291274
## dataIGI$UMK     0.00426805 0.155915 0.0333111
## dataIGI$PPS     0.00038642 0.091594 0.0069163
## dataIGI$PP      0.34816609 0.412086 0.3603861
```