

Boosting_InversJarak_KNN_Migrasi

Anandika Dian P.

3/29/2020

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

```
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

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("s1_n",J,"w",deg,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.w1.Rdata")

load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w1.Rdata")
jateng_w<-get(load("d:/Skripsweet/Bimbingan 10/IGI_jateng.w1.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] -0.5638026
## 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] "s1_n8w0.4"

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

## [1] -1.59667
## 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] "s1_n8w0.4"

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

## Time difference of 13.69992 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^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 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 = 3.9046, p-value = 4.719e-05
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.39895766      -0.02941176      0.01203606
```

```
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 = 1.9039, p-value = 0.02846
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.17694997      -0.02941176      0.01174872
```

```
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
## -0.861469 -0.131932  0.029941  0.168982  0.504212
```

```
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.8096891   1.0209558   3.7315 0.0001903
## dataIGI$INFLASI -0.0963059   0.1148840  -0.8383 0.4018684
## dataIGI$PMTB     0.0113006   0.0041141   2.7468 0.0060183
## dataIGI$UMK      -0.0873274   0.0347449  -2.5134 0.0119578
## dataIGI$PPS      0.2983172   0.1008178   2.9590 0.0030867
## dataIGI$PP       -0.0049266   0.0056083  -0.8784 0.3797018
##
## Rho: 0.51188, LR test value: 8.5278, p-value: 0.0034976
## Asymptotic standard error: 0.1389
##      z-value: 3.6851, p-value: 0.00022862
## Wald statistic: 13.58, p-value: 0.00022862
##
## Log likelihood: -4.643215 for lag model
## ML residual variance (sigma squared): 0.071419, (sigma: 0.26724)
## Nagelkerke pseudo-R-squared: 0.61018
## Number of observations: 35
## Number of parameters estimated: 8
## AIC: 25.286, (AIC for lm: 31.814)
## LM test for residual autocorrelation
## test value: 0.63079, p-value: 0.42706
```

```
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$INFLASI -0.103638939 -0.093659179 -0.19729812
## dataIGI$PMTB     0.012161060  0.010990028  0.02315109
## dataIGI$UMK      -0.093976726 -0.084927375 -0.17890410
## dataIGI$PPS      0.321031843  0.290118553  0.61115040
## dataIGI$PP       -0.005301724 -0.004791202 -0.01009293
## =====
## Simulation results (asymptotic variance matrix):
## =====
## Simulated standard errors
##           Direct      Indirect      Total
## dataIGI$INFLASI 0.108144364 0.117341460 0.21082892
## dataIGI$PMTB     0.004398356 0.008112035 0.01134841
## dataIGI$UMK      0.039303973 0.075331813 0.10833761
## dataIGI$PPS      0.108597024 0.189044157 0.26606461
## dataIGI$PP       0.006336569 0.009221223 0.01519491
```

```

##
## Simulated z-values:
##           Direct   Indirect   Total
## dataIGI$INFLASI -1.0225575 -0.7820057 -0.9597618
## dataIGI$PMTB     2.8733041  1.5218727  2.2014807
## dataIGI$UMK      -2.4035902 -1.2478997 -1.7397207
## dataIGI$PPS       3.0063385  1.6099003  2.3709341
## dataIGI$PP       -0.9497618 -0.7624743 -0.8587859
##
## Simulated p-values:
##           Direct   Indirect   Total
## dataIGI$INFLASI  0.3065171  0.43421  0.337175
## dataIGI$PMTB     0.0040620  0.12804  0.027702
## dataIGI$UMK      0.0162350  0.21207  0.081908
## dataIGI$PPS      0.0026441  0.10742  0.017743
## dataIGI$PP       0.3422333  0.44578  0.390459

```