Package ‘GWnnegPCA’

October 12, 2022

Type Package

Title Geographically Weighted Non-Negative Principal Components Analysis

Version 0.0.4

Description Implements a geographically weighted non-negative principal components analysis, which consists of the fusion of geographically weighted and sparse non-negative principal components analyses <doi:10.17608/k6.auckland.9850826.v1>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Language en-US

Depends R (>= 3.5.0)

Imports sp, sf, pracma, geodist, nsprcomp, methods, spData

SystemRequirements C++11, GDAL (>= 2.0.1), GEOS (>= 3.4.0), PROJ (>= 4.8.0)

NeedsCompilation no

Author Narumasa Tsutsumida [aut, cre] (<https://orcid.org/0000-0002-6333-0301>)

Maintainer Narumasa Tsutsumida <rsnaru.jp@gmail.com>

Repository CRAN

Date/Publication 2020-11-18 14:50:06 UTC

R topics documented:

gw_nsprcomp *Geographically Weighted non-negative Principal Component Analysis*

Description

Implementation of geographically weighted non-negative principal component analysis, whcih consists of the fusion of GWPCA and sparse non-negative PCA.

Usage

```
gw_nsprcomp(data, elocat, vars, bw, k = 2, kernel, adaptive = TRUE,
          p = 2, theta = 0, longlat = FALSE, geodisic_measure = "cheap",
          dMat = NULL, n.obs = NA, n.iter = 1, ncomp = k, nneg = TRUE,
          localcenter = TRUE, localscale = FALSE,...)
```

Arguments

`data` a Spatial*DataFrame either SpatialPointsDataFrame or SpatialPolygonsDataFrame as defined in package sp.

`elocat` Same as GWmodel::gwpca. two-column numeric array or Spatial*DataFrame object for providing evaluation locations, i.e. SpatialPointsDataFrame or SpatialPolygonsDataFrame as defined in package sp.

`vars` the number of retained components; k must be less than the number of variables.

`bw` bandwidth used in the weighting function, possibly calculated by bw.gwpca;fixed (distance) or adaptive bandwidth(number of nearest neighbours).

`k` the number of retained components; k must be less than the number of variables

`kernel` Same as GWmodel::gwpca. Function chosen as follows: gaussian: wgt = exp(-.5*(vdist/bw)^2); exponential: wgt = exp(-vdist/bw); bisquare: wgt = (1-(vdist/bw)^2)^2 if vdist < bw, wgt=0 otherwise; tricube: wgt = (1-(vdist/bw)^3)^3 if vdist < bw, wgt=0 otherwise; boxcar: wgt=1 if dist < bw, wgt=0 otherwise

see help(GWmodel::gw.weight) more detail.

`adaptive` if TRUE calculate an adaptive kernel where the bandwidth corresponds to the number of nearest neighbours (i.e. adaptive distance); default is FALSE, where a fixed kernel is found (bandwidth is a fixed distance).

`p` the power of the Minkowski distance, default is 2, i.e. the Euclidean distance.

`theta` an angle in radians to rotate the coordinate system, default is 0.

`longlat` if TRUE, great circle distances will be calculated.

`geodisic_measure`

geodisic_measure is used when latlon coordinate. The distance is cauclated by geodist::geodist(). One of "haversine" "vincenty", "geodesic", or "cheap" specifying desired method of geodesic distance calculation. "Cheap" is the fastest way but may have errors if the ROI is large.

`dMat` a pre-specified distance matrix, it can be calculated by the function gw.dist .

`n.obs` Number of observations used to find the correlation matrix if using a correlation matrix. Used for finding the goodness of fit statistics. Must be specified if using a correlaton matrix and finding confidence intervals.

`n.iter` Same as psych::fa. Number of bootstrap interations to do in fa or fa.poly

`ncomp` the number of principal components (PCs) to be computed. With the default setting, PCs are computed until x is fully deflated. ncomp can be specified implicitly if k is given as a vector.

`nneg` a logical value indicating whether the loadings should be non-negative, i.e. the PAs should be constrained to the non-negative orthant.

`localcenter` If TRUE, local weighted x is centered. The default is TRUE.

`localscale` If TRUE, local weighted x is scaled. The default is FALSE.

`...` arguments passed to or from other methods.

Value

`loadings` The localized loadings

`score` The PC score by the localized non-negative PCA.

`sdev` The localized standard deviation of the principal components.

Author(s)

N. Tsutsumida

References

Tsutsumida N., Murakami D.,Yoshida T., Nakaya T. Exploring geographically weighted non negative principal component analysis for producing index. The 27th meeting of GIS association of Japan, Tokyo, 20-21 October, 2018 (Japanese), http://www.gisa-japan.org/conferences/proceedings/2018/papers/C52.pdf

Tsutsumida N., Murakami D., Yoshida T., Nakaya T., Lu B., and P. Harris. Geographically Weighted Non-negative Principal Component Analysis for Exploring Spatial Variation in Multidimensional Composite Index, Geocomputation 2019, https://doi.org/10.17608/k6.auckland.9850826.v1

Examples

```
### This example is for demonstrating GWnnegPCA only.
### The application does not imply any reasonable meanings.

library(sp)
library(spData)
library(sf)

boston <- as(st_read(system.file("shapes/boston_tracts.shp",
                                package="spData")[1]), "Spatial")
Data.scaled <- scale(as.matrix(boston@data[, c("AGE","RAD","TAX")]))
Coords <- as.matrix(cbind(boston$LON, boston$LAT))
Data.scaled.spdf <- SpatialPointsDataFrame(Coords, as.data.frame(Data.scaled))
```

```
gwnnegpca_ans <- gw_nsprcomp(
  data = Data.scaled.spdf,
  vars = colnames(Data.scaled.spdf@data),
  bw = 0.25,
  k = 3,
  longlat = TRUE,
  kernel="bisquare",
  adaptive = TRUE,
  nneg=TRUE,
  center=FALSE
)

boston$PC1_load_AGE <- gwnnegpca_ans$loadings[,"AGE","PC1"]
boston$PC1_load_RAD <- gwnnegpca_ans$loadings[,"RAD","PC1"]
boston$PC1_load_TAX <- gwnnegpca_ans$loadings[,"TAX","PC1"]

plot(st_as_sf(boston)[,c("PC1_load_AGE","PC1_load_RAD","PC1_load_TAX")])
```

Index