Package 'mgwrsar'

February 20, 2025

Type Package

Title GWR, Mixed GWR and Multiscale GWR with Spatial Autocorrelation

Version 1.1

Date 2025-02-18

Maintainer Ghislain Geniaux <ghislain.geniaux@inrae.fr>

Description
Functions for computing (Mixed and Multiscale) Geographically Weighted Regression with spatial autocorrelation, Geniaux and Martinetti (2017) <doi:10.1016/j.regsciurbeco.2017.04.001>.

License GPL (>= 2)

Depends R (>= 3.5.0), Rcpp, sp, leaflet, Matrix

Imports ggplot2, sf, knitr, methods, doParallel, foreach, htmltools,
nabor, mapview, microbenchmark, rlang, dplyr, gridExtra, grid,
mboost, mgcv, caret, stringr, SMUT

Suggests R.rsp

VignetteBuilder R.rsp

LinkingTo RcppEigen, Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

Encoding UTF-8

Author Ghislain Geniaux [aut, cre],
Davide Martinetti [aut],
César Martinez [aut]

Repository CRAN

Date/Publication 2025-02-20 12:20:06 UTC

Contents

`atds_gwr` *atds_gwr Top-Down Scaling approach of GWR*

Description

This function performs a Geographically Weighted Regression (GWR) using a top-down scaling approach, adjusting GWR coefficients with a progressively decreasing bandwidth as long as the AICc criterion improves.

Usage

```
atds_gwr(formula,data,coords,kernels='triangle',fixed_vars=NULL,
control_tds=list(nns=30),control=list(adaptive=TRUE,verbose=FALSE))
```

Arguments

`formula`	a formula.
`data`	a dataframe.
`coords`	default NULL, a dataframe or a matrix with coordinates
`kernels`	A vector containing the kernel types. Possible types: triangle ("triangle"), bisquare ("bisq"), tricube ("tcub"), epanechnikov ("epane").
`fixed_vars`	a vector with the names of spatiallay constant coefficient for mixed model. All other variables present in formula are supposed to be spatially varying. If empty or NULL (default), all variables in formula are supposed to be spatially varying.

`control_tds`	list of extra control arguments for tds_mgwr model - see tds_gwr Help
`control`	list of extra control arguments for MGWRSAR wrapper - see MGWRSAR Help

See Also

tds_mgwr, gwr_multiscale, MGWRSAR, bandwidths_mgwrsar, summary_mgwrsar.

`coef,mgwrsar-method` *coef for mgwrsar model*

Description

coef for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar'
coef(object, ...)
```

Arguments

`object`	A model of class `mgwrsar-class`.
`...`	coef parameters forwarded.

Value

A named list with a matrix of varying coefficients and a vector or non varying coefficients.

`find_TP` *Search of a suitable set of target points. find_TP is a wrapper function that identifies a set of target points based on spatial smoothed OLS residuals.*

Description

Search of a suitable set of target points. find_TP is a wrapper function that identifies a set of target points based on spatial smoothed OLS residuals.

Usage

```
find_TP(formula, data,coords,kt,ks=16,Wtp=NULL,type='residuals',
model_residuals=NULL,verbose=0,prev_TP=NULL,nTP=NULL)
```

Arguments

formula a formula

data a dataframe or a spatial dataframe (SP package)

coords a dataframe or a matrix with coordinates, not required if data is a spatial dataframe

kt the minimum number of first neighbors with lower (resp.higer) absolute value of the smoothed residuals.

ks the number of first neighbors for computing the smoothed residuals, default 16.

Wtp a precomputed matrix of weights, default NULL.

type method for choosing TP, could be ’residuals’, ’ kdtree’,’random’, default ’residuals’

model_residuals (optional) a vector of residuals.

verbose verbose mode, default FALSE.

prev_TP index of already used TP (version length(kt)>1), default NULL.

nTP numbeer of target points for random choice of target points, default NULL.

Details

find_TP is a wrapper function that identifies a set of target points, based on spatial smoothed residuals by default. If no vector of residuals are provided, OLS residuals are computed. The function first computes the smooth of model residuals using a Shepard’s kernel with ks neighbors (default 16). Then it identifies local maxima (resp. minima) that fits the requirement of having at least kt neighbors with lower (resp.higer) absolute value of the smoothed residuals. As kt increases the number of target points decreases.

Value

find_TP returns an index vector of Target Points set.

Examples

```
library(mgwrsar)
## loading data example
data(mydata)
coords=as.matrix(mydata[,c(”x”,”y”)])
TP=find_TP(formula = 'Y_gwr~X1+X2+X3', data =mydata,coords=coords,kt=6,
type='residuals')
# only 60 targets points are used
length(TP)

model_GWR_tp<-MGWRSAR(formula = 'Y_gwr~X1+X2+X3', data = mydata,
coords=coords, fixed_vars=NULL,kernels=c('gauss'),  H=0.03, Model = 'GWR',
control=list(SE=TRUE,TP=TP,ks=12))
summary(model_GWR_tp@Betav)
```

`fitted,mgwrsar-method` *fitted for mgwrsar model*

Description

fitted for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar'
fitted(object, ...)
```

Arguments

`object`	A model of class `mgwrsar-class`.
`...`	fitted parameters forwarded.

Value

A vector of fitted values.

`golden_search_bandwidth` *golden_search_bandwidth to be documented*

Description

golden_search_bandwidth to be documented

Usage

```
golden_search_bandwidth(formula,H2=NULL,data, coords, fixed_vars,
kernels, Model, control,lower.bound, upper.bound,tolerance=0.000001)
```

Arguments

`formula`	to be documented
`H2`	to be documented
`data`	to be documented
`coords`	to be documented
`fixed_vars`	to be documented
`kernels`	to be documented
`Model`	to be documented

`control` to be documented

`lower.bound` to be documented

`upper.bound` to be documented

`tolerance` to be documented

Value

a list(minimum=res,objective=objective,model=model).

---

kernel_matW

kernel_matW A function that returns a sparse weight matrix based computed with a specified kernel (gauss,bisq,tcub,epane,rectangle,triangle) considering coordinates provides in S and a given bandwidth. If NN<nrow(S) only NN firts neighbours are considered. If Type!='GD' then S should have additional columns and several kernels and bandwidths should be be specified by the user.

Description

kernel_matW A function that returns a sparse weight matrix based computed with a specified kernel (gauss,bisq,tcub,epane,rectangle,triangle) considering coordinates provides in S and a given bandwidth. If NN<nrow(S) only NN firts neighbours are considered. If Type!='GD' then S should have additional columns and several kernels and bandwidths should be be specified by the user.

Usage

```
kernel_matW(H,kernels,coords,NN,TP=NULL,Type='GD',adaptive=FALSE,
diagnull=TRUE,alpha=1,theta=1,dists=NULL,indexG=NULL,extrapol=FALSE,QP=NULL,K=0)
```

Arguments

`H` A vector of bandwidths

`kernels` A vector of kernel types

`coords` A matrix with variables used in kernel (reference)

`NN` Number of spatial Neighbours for kernels computations

`TP` A vector with index of target points

`Type` Type of Genelarized kernel product ('GD' only spatial,'GDC' spatial + a categorical variable,'GDX' spatial + a continuous variable, 'GDT' spatial + a time index, and other combinations 'GDXXC','GDTX',...)

`adaptive` A vector of boolean to choose adaptive version for each kernel

`diagnull` Zero on diagonal, default FALSE

`alpha` TO BE DOCUMENTED

`theta` TO BE DOCUMENTED

`dists` TO BE DOCUMENTED

`indexG` TO BE DOCUMENTED

`extrapol` TO BE DOCUMENTED

`QP` A matrix with variables used in kernel (neighbors), default NULL (if NULL coord_j=coord_i)

`K` TO BE DOCUMENTED

Value

A sparse Matrix of weights (dgCMatrix).

Examples

```
 library(mgwrsar)
 ## loading data example
 data(mydata)
 coords=as.matrix(mydata[,c("x","y")])
## Creating a spatial weight matrix (sparce dgCMatrix) of 4 nearest neighbors with 0 in diagonal
 W=kernel_matW(H=4,kernels='rectangle',coords=coords,NN=4,adaptive=TRUE,diagnull=TRUE)
```

`MGWRSAR` *Estimation of linear and local linear model with spatial autocorrelation model (mgwrsar).*

Description

MGWRSAR is is a wrapper function for estimating linear and local linear models with spatial autocorrelation (SAR models with spatially varying coefficients).

Usage

```
MGWRSAR(formula, data, coords, fixed_vars = NULL, kernels, H,
Model = "GWR", control = list())
```

Arguments

`formula` a formula.

`data` a dataframe or a spatial dataframe (sp package).

`coords` default NULL, a dataframe or a matrix with coordinates, not required if data is a spatial dataframe.

`fixed_vars` a vector with the names of spatiallay constant coefficient for mixed model. All other variables present in formula are supposed to be spatially varying. If empty or NULL (default), all variables in formula are supposed to be spatially varying.

`kernels` A vector containing the kernel types. Possible types: rectangle ("rectangle"), bisquare ("bisq"), tricube ("tcub"), epanechnikov ("epane"), gaussian ("gauss")) .

`H` vector containing the bandwidth parameters for the kernel functions.

`Model` character containing the type of model: Possible values are "OLS", "SAR", "GWR" (default), "MGWR" , "MGWRSAR_0_0_kv","MGWRSAR_1_0_kv", "MGWRSAR_0_kc_kv", "MGWRSAR_1_kc_kv", "MGWRSAR_1_kc_0". See Details for more explanation.

`control` list of extra control arguments for MGWRSAR wrapper - see Details below

Details

Z A matrix of variables for genralized kernel product, default NULL.

W A row-standardized spatial weight matrix for Spatial Aurocorrelation, default NULL.

type Verbose mode, default FALSE.

adaptive A vector of boolean to choose adaptive version for each kernel.

kernel_w The type of kernel for computing W, default NULL.

h_w The bandwidth value for computing W, default 0.

Method Estimation method for computing the models with Spatial Dependence. '2SLS' or 'B2SLS', default '2SLS'.

TP Avector of target points, default NULL.

doMC Parallel computation, default FALSE. If TRUE and control_tds$doMC is also TRUE, then control$doMC is set to FALSE.

ncore Number of CPU core for parallel computation, default 1

isgcv If TRUE, compute a LOOCV criteria, default FALSE.

isfgcv If TRUE, simplify the computation of CV criteria (remove or not i when using local instruments for model with lambda spatially varying), default TRUE.

maxknn When n >NmaxDist, only the maxknn first neighbours are used for distance compution, default 500.

NmaxDist When n >NmaxDist only the maxknn first neighbours are used for distance compution, default 5000

verbose Verbose mode, default FALSE.

Value

MGWRSAR returns an object of class mgwrsar with at least the following components:

Betav matrix of coefficients of dim(n,kv) x kv.

Betac vector of coefficients of length kc.

Model The sum of square residuals.

Y The dependent variable.

XC The explanatory variables with constant coefficients.

XV The explanatory variables with varying coefficients.

X The explanatory variables.

W The spatial weight matrix for spatial dependence.

isgcv if gcv has been computed.

edf The estimated degrees of freedom.

formula The formula.

data The dataframe used for computation.

Method The type of model.

coords The spatial coordinates of observations.

H The bandwidth vector.

fixed_vars The names of constant coefficients.

kernels The kernel vector.

SSR The sum of square residuals.

residuals The vector of residuals.

fit the vector of fitted values.

sev local standard error of parameters.

get_ts Boolean, if trace of hat matrix Tr(S) should be stored.

NN Maximum number of neighbors for weights computation

MGWRSAR is is a wrapper function for estimating linear and local linear model with spatial autocorrelation that allows to estimate the following models : $y = \beta_c X_c + \epsilon_i$ (OLS)

$y = \beta_v(u_i, v_i) X_v + \epsilon_i$ (GWR)

$y = \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i$ (MGWR)

$y = \lambda W y + \beta_c X_c + \epsilon_i$ (MGWR-SAR(0,k,0))

$y = \lambda W y + \beta_v(u_i, v_i) X_v + \epsilon_i$ (MGWR-SAR(0,0,k))

$y = \lambda W y + \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i$ (MGWR-SAR(0,k_c,k_v))

$y = \lambda(u_i, v_i) W y + \beta_c X_c + \epsilon_i$ (MGWR-SAR(1,k,0))

$y = \lambda(u_i, v_i) W y + \beta_v(u_i, v_i) X_v + \epsilon_i$ (MGWR-SAR(1,0,k))

$y = \lambda(u_i, v_i) W y + \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i$ (MGWR-SAR(1,k_c,k_v))

When model imply spatial autocorrelation, a row normalized spatial weight matrix must be provided. 2SLS and Best 2SLS method can be used. When model imply local regression, a bandwidth and a kernel type must be provided. Optimal bandwidth can be estimated using bandwidths_mgwrsar function. When model imply mixed local regression, the names of stationary covariates must be provided.

#' In addition to the ability of considering spatial autocorrelation in GWR/MGWR like models, MGWRSAR function introduces several useful technics for estimating local regression with space coordinates:

- it uses RCCP and RCCPeigen code that speed up computation and allows parallel computing via doMC package;

- it allows to drop out variables with not enough local variance in local regression, which allows to consider dummies in GWR/MGWR framework without trouble.
- it allows to drop out local outliers in local regression.
- it allows to consider additional variable for kernel, including time (asymetric kernel) and categorical variables (see Li and Racine 2010). Experimental version.

References

Geniaux, G. and Martinetti, D. (2017). A new method for dealing simultaneously with spatial autocorrelation and spatial heterogeneity in regression models. Regional Science and Urban Economics. (https://doi.org/10.1016/j.regsciurbeco.2017.04.001)

McMillen, D. and Soppelsa, M. E. (2015). A conditionally parametric probit model of microdata land use in chicago. Journal of Regional Science, 55(3):391-415.

Loader, C. (1999). Local regression and likelihood, volume 47. springer New York.

Franke, R. and Nielson, G. (1980). Smooth interpolation of large sets of scattered data. International journal for numerical methods in engineering, 15(11):1691-1704.

See Also

bandwidths_mgwrsar, summary, plot, predict, kernel_matW

Examples

```
library(mgwrsar)
## loading data example
data(mydata)
coords=as.matrix(mydata[,c(”x”,”y”)])
## Creating a spatial weight matrix (sparce dgCMatrix)
## of 4 nearest neighbors with 0 in diagonal
W=kernel_matW(H=4,kernels='rectangle',coords=coords,NN=4,adaptive=TRUE,
diagnull=TRUE)
mgwrsar_0_kc_kv<-MGWRSAR(formula = 'Y_mgwrsar_0_kc_kv~X1+X2+X3', data = mydata,
coords=coords, fixed_vars='X2',kernels=c('gauss'),H=20, Model = 'MGWRSAR_0_kc_kv',
control=list(SE=FALSE,adaptive=TRUE,W=W))
summary(mgwrsar_0_kc_kv)
```

---

mgwrsar-class *Class of mgwrsar Model.*

Description

Class of mgwrsar Model.

Slots

- `Betav` matrix, the estimated varying coefficients, dim(n,kv).
- `Betac` numeric, the estimated constant coefficients, length kc.
- `Model` character, The type of model.
- `fixed_vars` character, a vector with name of constant covarariate.
- `Y` numeric, the dependent variable.
- `XC` matrix, the explanatory variables with constant coefficients.
- `XV` matrix, the explanatory variables with varying coefficients.
- `X` matrix, the explanatory variables.
- `W` SparseMatrix, the spatial weight matrix for spatial dependence.
- `isgcv` logical, if gcv has been computed.
- `edf` numeric, the estimated degrees of freedom.
- `formula` `formula`
- `data` dataframe, The dataframe used for computation.
- `Method` character, the estimation technique for computing the models with Spatial Dependence. '2SLS' or 'B2SLS', default '2SLS'.
- `coords` matrix, the spatial coordinates of observations.
- `H` numeric, the bandwidth vector.
- `H2` numeric, the time bandwidth vector.
- `kernels` character, the type of kernel.
- `adaptive` logical, adaptive kernel.
- `Type` character, the type of General Kernel Product.
- `TP` numeric, index of target points.
- `SSRtp` numeric, the sum of square residuals for TP.
- `SSR` numeric, the sum of square residuals.
- `residuals` numeric, the vector of residuals.
- `fit` numeric, the vector of fitted values.
- `pred` numeric, the vector of predicted values.
- `sev` matrix, local standard error of varying coefficients.
- `se` numeric, standard error of constant coefficients.
- `tS` numeric, Trace(S).
- `Shat,` hat matrix
- `R_k,` list of hat matrix by var
- `h_w` numeric, the bandwidth value for computing W, default 0.
- `kernel_w` the type of kernel for computing W, default NULL.
- `RMSE` numeric, Root Mean Square Error for Target Points.
- `RMSEtp` numeric, Root Mean Square Error for all Points.

CV numeric, Leave One Out CV.

AIC numeric, Akaike Criteria.

AICc numeric, Corrected Akaike Criteria.

AICctp numeric, Corrected Akaike Criteria for TP

BIC numeric, Bayesian Information Criteria.

R2 numeric, R2.

R2_adj numeric, adjusted R2.

get_ts logical, if trace of hat matrix Tr(S) should be stored.

NN numeric, the maximum number of neighbors for weights computation

doMC logical, parallel computation.

ncore numeric, number of cores.

mycall a call, the call of the model.

ctime numeric, the computing times in seconds.

HRMSE matrix, RMSE log.

HBETA list, estimated BETA at each iteration.

loglik numeric, value of loglik.

G list, list of neighboring index and distances (knn object from nabor package).

V numeric, neighbors sequence for TDS.

Vt numeric, neighbors sequence for TDS.

Z numeric, time for GDT kernel type

TS numeric, Diagonal of Hat Matrix

alpha numeric, ratio for GDT kernels

theta numeric, ratio for GDT kernels

---

mgwrsar_bootstrap_test

A bootstrap test for Betas for mgwrsar class model.

Description

A bootstrap test for Betas for mgwrsar class model.

Usage

```
mgwrsar_bootstrap_test(x0,x1,B=100,doMC=FALSE,ncore=1,type='standard'
,eps='H1',df='H1',focal='median',D=NULL)
```

Arguments

`x0`	The H0 mgwrsar model
`x1`	The H1 mgwrsar model
`B`	number of bootstrap repetitions, default 100
`doMC`	If TRUE, doParallel parallelization
`ncore`	number of cores
`type`	type of bootstap : 'wild','Rademacher','spatial' or 'standard' (default)
`eps`	Hypothesis under wich residuals are simulated, 'H0' or 'H1' (default)
`df`	Hypothesis under wich degree of freedom is estimated.
`focal`	see sample_stat help
`D`	A matrix of distance

Value

The value of the statictics test and a p ratio.

See Also

mgwrsar_bootstrap_test_all

`mgwrsar_bootstrap_test_all` *A bootstrap test for testing nullity of all Betas for mgwrsar class model,*

Description

A bootstrap test for testing nullity of all Betas for mgwrsar class model,

Usage

```
mgwrsar_bootstrap_test_all(model,B=100,doMC=FALSE,ncore=1,
type='standard')
```

Arguments

`model`	A mgwrsar model
`B`	number of bootstrap replications, default 100
`doMC`	If TRUE, doMC parallelization
`ncore`	number of cores.
`type`	type of boostrap ('spatial','wild','random')

Value

a matrix with statistical test values and p ratios

See Also

mgwrsar_bootstrap_test

---

modc *modc is a set of models to correct approximation of hat matrix trace*

Description

modc is a set of models to correct approximation of hat matrix trace

Author(s)

Ghislain Geniaux and Davide Martinetti <ghislain.geniaux@inra.fr>

References

doi:10.1016/j.regsciurbeco.2017.04.001

---

multiscale_gwr *multiscale_gwr This function adapts the multiscale Geographically Weighted Regression (GWR) methodology proposed by Fotheringam et al. in 2017, employing a backward fitting procedure within the MGWRSAR subroutines. The consecutive bandwidth optimizations are performed by minimizing the corrected Akaike criteria.*

Description

multiscale_gwr This function adapts the multiscale Geographically Weighted Regression (GWR) methodology proposed by Fotheringam et al. in 2017, employing a backward fitting procedure within the MGWRSAR subroutines. The consecutive bandwidth optimizations are performed by minimizing the corrected Akaike criteria.

Usage

```
multiscale_gwr(formula,data,coords,kernels='bisq',init='GWR',
maxiter=20,nstable=6,tolerance=0.000001,doMC=FALSE,ncore=1,HF=NULL,
H0=NULL,H2=NULL,Model=NULL,model=NULL,get_AICg=FALSE,verbose=FALSE,
control=list(SE=FALSE,adaptive=TRUE,NN=800,isgcv=FALSE,family=gaussian()))
```

Arguments

`formula` A formula.

`data` A dataframe.

`coords` default NULL, a dataframe or a matrix with coordinates.

`kernels` A vector containing the kernel types. Possible types: rectangle ("rectangle"), bisquare ("bisq"), tricube ("tcub"), epanechnikov ("epane")

`init` starting model (lm or GWR)

`maxiter` maximum number of iterations in the back-fitting procedure.

`nstable` required number of consecutive unchanged optimal bandwidth (by covariate) before leaving optimisation of bandwidth size, default 3.

`tolerance` value to terminate the back-fitting iterations (ratio of change in RMSE)

`doMC` A boolean for Parallel computation, default FALSE.

`ncore` number of CPU cores for parallel computation, default 1.

`HF` if available, a vector containing the optimal bandwidth parameters for each covariate, default NULL.

`H0` A bandwidth value for the starting GWR model, default NULL.

`H2` A bandwidth temporal value for the starting GWR model, default NULL.

`Model` Type of Model.

`model` A previous model estimated using multiscale_gwr function, default NULL

`get_AICg` Boolean, should Global AICc be estimated.

`verbose` Boolean, verbose mode.

`control` a list of extra control arguments, see MGWRSAR help.

Value

Return an object of class mgwrsar

`mydata` *mydata is a simulated data set of a mgwrsar model*

Description

mydata is a simulated data set of a mgwrsar model

Format

A data frames with 1000 rows 22 variables and a matrix of coordinates with two columns

Author(s)

Ghislain Geniaux and Davide Martinetti `<ghislain.geniaux@inrae.fr>`

References

doi:10.1016/j.regsciurbeco.2017.04.001

`mydatasf` *mydataf is a Simple Feature object with real estate data in south of France.*

Description

mydataf is a Simple Feature object with real estate data in south of France.

Format

A sf object with 1403 rows, 5 columns

Author(s)

Ghislain Geniaux <ghislain.geniaux@inrea.fr>

References

https://www.data.gouv.fr/fr/datasets/demandes-de-valeurs-foncieres/

`normW` *normW row normalization of dgCMatrix*

Description

normW row normalization of dgCMatrix

Usage

```
normW(W)
```

Arguments

W	A dgCMatrix class matrix

Value

A row normalized dgCMatrix

plot,mgwrsar,missing-method

Plot method for mgwrsar model

Description

Plot method for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar,missing'
plot(
  x,
  y,
  type = "coef",
  var = NULL,
  crs = NULL,
  mypalette = "RdYlGn",
  opacity = 0.5,
  fopacity = 0.5,
  nbins = 8,
  radius = 500,
  mytile = "Stadia.StamenTonerBackground",
  myzoom = 8,
  myresolution = 150,
  LayersControl = TRUE,
  myzoomControl = TRUE,
  mytile2 = NULL,
  ScaleBar = NULL,
 ScaleBarOptions = list(maxWidth = 200, metric = TRUE, imperial = FALSE, updateWhenIdle
    = TRUE),
  MyLegendTitle = NULL,
  lopacity = 0.5
)
```

Arguments

`x`	A model of class `mgwrsar-class`.
`y`	missing
`type`	default 'coef', for plotting the value of the coefficients. Local t-Student could also be plot using 't_coef', residuals using 'residuals' and fitted using 'fitted'.
`var`	Names of variable to plot.
`crs`	A CRS projection.
`mypalette`	A leaflet palette.
`opacity`	Opacity of border color.

`fopacity`	Opacity of fill color.
`nbins`	nbins.
`radius`	radius of circle for plot of points.
`mytile`	tile 1.
`myzoom`	level of zoom for tile 1.
`myresolution`	resolution for tile 1.
`LayersControl`	layers contols.
`myzoomControl`	zoem control.
`mytile2`	tile 2.
`ScaleBar`	ScaleBar.
`ScaleBarOptions`	options for ScaleBar.
`MyLegendTitle`	Legend title.
`lopacity`	opacity for legend.

Value

A Interactive Web Maps with local parameters plot and Open Street Map layer.

`plot_effect` *plot_effect plot_effect is a function that plots the effect of a variable X_k with spatially varying coefficient, i.e X_k * Beta_k(u_i,v_i) for comparing the magnitude of effects of between variables.*

Description

plot_effect plot_effect is a function that plots the effect of a variable X_k with spatially varying coefficient, i.e X_k * Beta_k(u_i,v_i) for comparing the magnitude of effects of between variables.

Usage

```
plot_effect(model,sampling=TRUE,nsample=2000,nsample_max=5000,title='')
```

Arguments

`model`	a model of mgwrsar class with some spatially varying coefficients.
`sampling`	Bolean, if nrow(model@Betav)> nsample_max a sample of size nsample is randomly selected, default TRUE.
`nsample`	integer, size of the sample if sampling is TRUE, default 2000.
`nsample_max`	integer, size max to engage sampling if sampling is TRUE, default 5000.
`title`	a title for the plot.

Examples

```
library(mgwrsar)
## loading data example
data(mydata)
coords=as.matrix(mydata[,c("x","y")])
## Creating a spatial weight matrix (sparce dgCMatrix)
## of 8 nearest neighbors with 0 in diagonal
model_GWR0<-MGWRSAR(formula = 'Y_gwr~X1+X2+X3', data = mydata,coords=coords,
fixed_vars=NULL,kernels=c('gauss'),H=0.13, Model = 'GWR',control=list(SE=TRUE))
plot_effect(model_GWR0)
```

---

predict,mgwrsar-method

predict method for mgwrsar model

Description

predict method for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar'
predict(
  object,
  newdata,
  newdata_coords,
  W = NULL,
  type = "BPN",
  h_w = 100,
  kernel_w = "rectangle",
  maxobs = 4000,
  beta_proj = FALSE,
  method_pred = "TP",
  k_extra = 8,
  ...
)
```

Arguments

`object` A model of class `mgwrsar-class`.

`newdata` a matrix or data.frame of new data.

`newdata_coords` a matrix of new coordinates, and eventually other variables if a General Kernel Product is used.

`W` the spatial weight matrix for models with spatial autocorrelation.

`type` Type for BLUP estimator, default "BPN". If NULL use predictions without spatial bias correction.

`h_w`	A bandwidth value for the spatial weight matrix
`kernel_w`	kernel type for the spatial weight matrix. Possible types: rectangle ("rectangle"), bisquare ("bisq"), tricube ("tcub"), epanechnikov ("epane"), gaussian ("gauss")) .
`maxobs`	maximum number of observations for exact calculation of solve(I- rho*W), default maxobs=4000.
`beta_proj`	A boolean, if TRUE the function then return a two elements list(Y_predicted,Beta_proj_out)
`method_pred`	If method_pred = 'TP' (default) prediction is done by recomputing a MGWRSAR model with new-data as target points, else if method_pred in ('tWtp_model','model','shepard') a matrix for projecting estimated betas is used (see details).
`k_extra`	number of neighboors for local parameter extrapolation if shepard kernel is used, default 8.
`...`	predict parameters forwarded.

Details

if method_pred ='tWtp_model', the weighting matrix for prediction is based on the expected weights of outsample data if they were had been added to insample data to estimate the corresponding MGWRSAR (see Geniaux 2022 for further detail), if method_pred ='shepard'a shepard kernel with k_extra neighbours (default 8) is used and if method_pred='kernel_model' the same kernel and number of neighbors as for computing the MGWRSAR model is used.

Value

A vector of predictions if beta_proj is FALSE or a list with a vector named Y_predicted and a matrix named Beta_proj_out.

A vector of predictions.

`residuals,mgwrsar-method`

residuals for mgwrsar model

Description

residuals for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar'
residuals(object, ...)
```

Arguments

`object`	A model of class `mgwrsar-class`.
`...`	residuals parameters forwarded.

Value

A vector of residuals.

simu_multiscale *Estimation of linear and local linear model with spatial autocorrelation model (mgwrsar).*

Description

The simu_multiscale function is designed for simulating a spatially varying coefficient DGP (Data Generating Process) based on formulations proposed by Fotheringam et al. (2017), Gao et al. (2021), or Geniaux (2024).

Usage

```
simu_multiscale(n=1000,myseed=1,type='GG2024',constant=NULL,
nuls=NULL,config_beta='default',config_snr=0.7,config_eps='normal',
ratiotime=1)
```

Arguments

`n` An integer number of observations

`myseed` An integer seed used for the simulation.

`type` Type of DGP used 'FT2017', 'Gao2021' or 'GG2024', default 'GG2024'.

`constant` A boolean parameter indicating whether the intercept term should be spatially varying (TRUE) or not (FALSE).

`nuls` A vector of null parameters, default NULL

`config_beta` name of the type of spatial pattern of Beta coefficients

`config_snr` a value of signal noise ratio

`config_eps` name of the distribution of error ('normal','unif' or 'Chi2')

`ratiotime` multiplicating factor, for spacetime DGP.

Value

A named list with simulated data ('mydata') and coords ('coords')

Examples

```
library(mgwrsar)
library(ggplot2)
library(gridExtra)
library(grid)
simu=simu_multiscale(1000)
mydata=simu$mydata
coords=simu$coords
```

```
p1<-ggplot(mydata,aes(x,y,col=Beta1))+geom_point() +scale_color_viridis_c()
p2<-ggplot(mydata,aes(x,y,col=Beta2))+geom_point() +scale_color_viridis_c()
p3<-ggplot(mydata,aes(x,y,col=Beta3))+geom_point() +scale_color_viridis_c()
p4<-ggplot(mydata,aes(x,y,col=Beta4))+geom_point() +scale_color_viridis_c()
grid.arrange(p1,p2,p3,p4,nrow=2,ncol=2, top = textGrob("DGP Geniaux (2024)"
,gp=gpar(fontsize=20,font=3)))
```

summary,mgwrsar-method *summary for mgwrsar model*

Description

summary for mgwrsar model

Usage

```
## S4 method for signature 'mgwrsar'
summary(object, ...)
```

Arguments

`object`	A model of class `mgwrsar-class`.
`...`	summary parameters forwarded.

Value

A summary object.

summary_Matrix *summary_Matrix to be documented*

Description

summary_Matrix to be documented

Usage

```
summary_Matrix(object, ...)
```

Arguments

`object`	to be documented
`...`	to be documented

Value

to be documented

tds_mgwr *Top-Down Scaling approach of multiscale GWR*

Description

This function performs a multiscale Geographically Weighted Regression (GWR) using a top-down scaling approach, adjusting GWR coefficients with a progressively decreasing bandwidth as long as the AICc criterion improves.

Usage

```
tds_mgwr(formula,data,coords,Model='tds_mgwr',kernels='triangle',
fixed_vars=NULL,H2=NULL,control_tds=list(nns=30,get_AIC=FALSE),
control=list(adaptive=TRUE))
```

Arguments

`formula` a formula.

`data` a dataframe.

`coords` default NULL, a dataframe or a matrix with coordinates

`Model` character containing the type of model: Possible values are "tds_mgwr" and "atds_mgwr", See Details for more explanation.

`kernels` A vector containing the kernel types. Possible types: triangle ("triangle"), rectangle ("rectangle"), bisquare ("bisq"), tricube ("tcub"), gaussian ("gauss"), epanechnikov ("epane").

`fixed_vars` a vector with the names of spatiallay constant coefficient for mixed model. All other variables present in formula are supposed to be spatially varying. If empty or NULL (default), all variables in formula are supposed to be spatially varying.

`H2` A scalar or vector of time bandwidths.

`control_tds` list of extra control arguments for tds_mgwr models

`control` list of extra control arguments for MGWRSAR wrapper

Details

nns Length of the sequence of decreasing bandwidth. Should be between 20 and 100, default 30

get_AIC Boolean, if the Global AICc using Yu et al 2019 should be computed. Required if the second stage 'atds_mgwr' has to be estimated. default FALSE

init_model Starting model, 'GWR' or 'OLS', 'default OLS'.

model_stage1 If model='tds_mgwr', model_stage1 can be used as a starting model (either a GWR model or a preious tds_mgwr model). For model='atds_mgwr, the user can specified an tds_mgwr model already computed with get_AIC=TRUE. default NULL.

doMC Parallel computation, default FALSE.

ncore number of CPU core for parallel computation, default 1

tol Tolerance for stopping criteria, default 0.0001

nrounds Number of nrounds for 'atds_mgwr' model. Default 3.

verbose verbose mode, default FALSE.

V A vector of decreasing bandwidths given by the user, default NULL

first_nn The value of the highest bandwidth for the sequence of decreasing bandwidth, default NULL.

minv The value of the smallest bandwidth for the sequence of decreasing bandwidth, default number of covariates + 2 .

H A vector of bandwidth, default NULL

Z A matrix of variables for genralized kernel product, default NULL.

W A row-standardized spatial weight matrix for Spatial Aurocorrelation, default NULL.

type Verbose mode, default FALSE.

adaptive A vector of boolean to choose adaptive version for each kernel.

kernel_w The type of kernel for computing W, default NULL.

h_w The bandwidth value for computing W, default 0.

Method Estimation method for computing the models with Spatial Dependence. '2SLS' or 'B2SLS', default '2SLS'.

TP Avector of target points, default NULL.

doMC Parallel computation, default FALSE. If TRUE and control_tds$doMC is also TRUE, then control$doMC is set to FALSE.

ncore Number of CPU core for parallel computation, default 1

isgcv If TRUE, compute a LOOCV criteria, default FALSE.

isfgcv If TRUE, simplify the computation of CV criteria (remove or not i when using local instruments for model with lambda spatially varying), default TRUE.

maxknn When n >NmaxDist, only the maxknn first neighbours are used for distance compution, default 500.

NmaxDist When n >NmaxDist only the maxknn first neighbours are used for distance compution, default 5000

verbose Verbose mode, default FALSE.

See Also

gwr_multiscale, MGWRSAR, bandwidths_mgwrsar, summary_mgwrsar.

Index