

Package ‘hgwr’

May 26, 2023

Type Package

Title Hierarchical and Geographically Weighted Regression

Version 0.3-0

Date 2023-05-24

Author Yigong Hu, Richard Harris, Richard Timmerman

Maintainer Yigong Hu <yigong.hu@bristol.ac.uk>

Description This model divides coefficients into three types, i.e., local fixed effects, global fixed effects, and random effects (Hu et al., 2022)<doi:10.1177/23998083211063885>. If data have spatial hierarchical structures (especially are overlapping on some locations), it is worth trying this model to reach better fitness.

License GPL (>= 2)

URL <https://github.com/HPDell/hgwr/>, <https://hpdell.github.io/hgwr/>

Imports Rcpp (>= 1.0.8)

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5.0), stats, utils

SystemRequirements GNU make

RoxygenNote 7.2.3

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-05-26 10:10:02 UTC

R topics documented:

hgwr-package	2
coef.hgwr	3
fitted.hgwr	4
hgwr	4
matrix2char	6
multisampling	7

multisampling.large	8
parse.formula	9
print.hgwr	9
print.summary.hgwr	10
print.table.md	11
residuals.hgwr	12
summary.hgwr	13
wuhan.hp	14

Index	16
--------------	-----------

hgwr-package	<i>HGWR: Hierarchical and Geographically Weighted Regression</i>
--------------	--

Description

An R and C++ implementation of Hierarchical and Geographically Weighted Regression (HGWR) model is provided in this package. This model divides coefficients into three types: local fixed effects, global fixed effects, and random effects. If data have spatial hierarchical structures (especially are overlapping on some locations), it is worth trying this model to reach better fitness.

Details

The DESCRIPTION file:

```

Package:          hgwr
Type:            Package
Title:           Hierarchical and Geographically Weighted Regression
Version:         0.3-0
Date:           2023-05-24
Author:          Yigong Hu, Richard Harris, Richard Timmerman
Maintainer:     Yigong Hu <yigong.hu@bristol.ac.uk>
Description:     This model divides coefficients into three types, i.e., local fixed effects, global fixed effects, and random effects.
License:         GPL (>= 2)
URL:            https://github.com/HPDell/hgwr/, https://hpdell.github.io/hgwr/
Imports:         Rcpp (>= 1.0.8)
LinkingTo:      Rcpp, RcppArmadillo
Depends:        R (>= 3.5.0), stats, utils
SystemRequirements: GNU make
Roxygen:        list(markdown = TRUE)
RoxygenNote:    7.2.3

```

Index of help topics:

```

coef.hgwr      Get estimated coefficients.
fitted.hgwr    Get fitted response.
hgwr           Hierarchical and Geographically Weighted

```

	Regression
hgwrn-package	HGWR: Hierarchical and Geographically Weighted Regression
matrix2char	Convert a numeric matrix to character matrix according to a format string.
multisampling	Simulated Spatial Multisampling Data (DataFrame)
multisampling.large	Large Scale Simulated Spatial Multisampling Data (DataFrame)
parse.formula	Parse a HGWR formula.
print.hgwrn	Print description of a 'hgwrn' object.
print.summary.hgwrn	Print summary of an 'hgwrn' object.
print.table.md	Print a character matrix as a table.
residuals.hgwrn	Get residuals.
summary.hgwrn	Summary an 'hgwrn' object.
wuhan.hp	Wuhan Second-hand House Price and POI Data (DataFrame)

Author(s)

Yigong Hu, Richard Harris, Richard Timmerman

Maintainer: Yigong Hu <yigong.hu@bristol.ac.uk>

References

Hu, Y., Lu, B., Ge, Y., Dong, G., 2022. Uncovering spatial heterogeneity in real estate prices via combined hierarchical linear model and geographically weighted regression. *Environment and Planning B: Urban Analytics and City Science*. DOI: 10.1177/23998083211063885.

coef.hgwrn	<i>Get estimated coefficients.</i>
------------	------------------------------------

Description

Get estimated coefficients.

Usage

```
## S3 method for class 'hgwrn'
coef(object, ...)
```

Arguments

object	An hgwrn object returned by <code>hgwrn()</code> .
...	Parameter received from other functions.

Value

A DataFrame object consists of all estimated coefficients.

See Also

[hgwr\(\)](#), [summary.hgwr\(\)](#), [fitted.hgwr\(\)](#) and [residuals.hgwr\(\)](#).

<code>fitted.hgwr</code>	<i>Get fitted reponse.</i>
--------------------------	----------------------------

Description

Get fitted reponse.

Usage

```
## S3 method for class 'hgwr'
fitted(object, ...)
```

Arguments

<code>object</code>	An hgwr object returned by hgwr() .
<code>...</code>	Parameter received from other functions.

Value

A vector consists of fitted response values.

See Also

[hgwr\(\)](#), [summary.hgwr\(\)](#), [coef.hgwr\(\)](#) and [residuals.hgwr\(\)](#).

<code>hgwr</code>	<i>Hierarchical and Geographically Weighted Regression</i>
-------------------	--

Description

A Hierarchical Linear Model (HLM) with local fixed effects.

Usage

```

hgwr(
  formula,
  data,
  local.fixed,
  coords,
  bw = "CV",
  kernel = c("gaussian", "bisquared"),
  alpha = 0.01,
  eps_iter = 1e-06,
  eps_gradient = 1e-06,
  max_iters = 1e+06,
  max_retries = 1e+06,
  ml_type = c("D_Only", "D_Beta"),
  verbose = 0
)

```

Arguments

formula	A formula. Its structure is similar to lmer function in lme4 package.
data	A DataFrame.
local.fixed	A character vector. It contains names of local fixed effects.
coords	A 2-column matrix. It consists of coordinates for each group.
bw	A numeric value. It is the value of bandwidth or "CV". In this stage this function only support adaptive bandwidth. And its unit must be the number of nearest neighbours. If "CV" is specified, the algorithm will automatically select an optimized bandwidth value.
kernel	A character value. It specify which kernel function is used in GWR part. Possible values are gaussian Gaussian kernel function $k(d) = \exp\left(-\frac{d^2}{b^2}\right)$ bisquared Bi-squared kernel function. If $d < b$ then $k(d) = \left(1 - \frac{d^2}{b^2}\right)^2$ else $k(d) = 0$
alpha	A numeric value. It is the size of the first trial step in maximum likelihood algorithm.
eps_iter	A numeric value. Terminate threshold of back-fitting.
eps_gradient	A numeric value. Terminate threshold of maximum likelihood algorithm.
max_iters	An integer value. The maximum of iteration.
max_retries	An integer value. If the algorithm tends to be diverge, it stops automatically after trying <i>max_retries</i> times.
ml_type	An integer value. Represent which maximum likelihood algorithm is used. Possible values are: D_Only Only D is specified by maximum likelihood. D_Beta Both D and $beta$ is specified by maximum likelihood.

verbose An integer value. Determine the log level. Possible values are:
0 no log is printed.
1 only logs in back-fitting are printed.
2 all logs are printed.

Value

A list describing the model with following fields.

gamma Coefficients of local fixed effects.

beta Coefficients of global fixed effects.

mu Coefficients of random effects.

D Variance-covariance matrix of random effects.

sigma Variance of errors.

effects A list including names of all effects.

call Calling of this function.

frame The DataFrame object sent to this call.

frame.parsed Variables extracted from the data.

groups Unique group labels extracted from the data.

Examples

```
data(multisampling)
hgwr(formula = y ~ g1 + g2 + x1 + (z1 | group),
      data = multisampling$data,
      local.fixed = c("g1", "g2"),
      coords = multisampling$coords,
      bw = 10)
```

matrix2char	<i>Convert a numeric matrix to character matrix according to a format string.</i>
-------------	---

Description

Convert a numeric matrix to character matrix according to a format string.

Usage

```
matrix2char(m, fmt = "%.6f")
```

Arguments

m A numeric matrix.

fmt Format string. Passing to `base::sprintf()`.

See Also

`base::sprintf()`, `print.hgwrn()`, `print.summary.hgwrn()`.

multisampling

Simulated Spatial Multisampling Data (DataFrame)

Description

A simulation data of spatial hierarchical structure and samples overlapping on certain locations.

Usage

```
data(multisampling)
```

Format

A list of two items called "data" and "coord". Item "data" is a data frame with 484 observations at 16 locations on the following 6 variables.

y a numeric vector, dependent variable y

g1 a numeric vector, group level independent variable g_1

g2 a numeric vector, group level independent variable g_2

z1 a numeric vector, sample level independent variable z_1

x1 a numeric vector, sample level independent variable x_1

group a numeric vector, group id of each sample

where *g1* and *g2* are used to estimate local fixed effects; *x1* is used to estimate global fixed effects and *z1* is used to estimate random effects.

Author(s)

Yigong Hu <yigong.hu@bristol.ac.uk>

Examples

```
data(multisampling)
hgwr(formula = y ~ g1 + g2 + x1 + (z1 | group),
      data = multisampling$data,
      local.fixed = c("g1", "g2"),
      coords = multisampling$coords,
      bw = 10)
```

multisampling.large *Large Scale Simulated Spatial Multisampling Data (DataFrame)*

Description

A large scale simulation data of spatial hierarchical structure and samples overlapping on certain locations.

Usage

```
data(multisampling)
```

Format

A list of three items called "data", "coords" and "beta". Item "data" is a data frame with 13862 observations at 200 locations and the following 6 variables.

y a numeric vector, dependent variable *y*

g1 a numeric vector, group level independent variable *g*₁

g2 a numeric vector, group level independent variable *g*₂

z1 a numeric vector, sample level independent variable *z*₁

x1 a numeric vector, sample level independent variable *x*₁

group a numeric vector, group id of each sample

where *g1* and *g2* are used to estimate local fixed effects; *x1* is used to estimate global fixed effects and *z1* is used to estimate random effects.

Author(s)

Yigong Hu <yigong.hu@bristol.ac.uk>

Examples

```
## Not run:
data(multisampling.large)
hgwr(formula = y ~ g1 + g2 + x1 + (z1 | group),
      data = multisampling.large$data,
      local.fixed = c("g1", "g2"),
      coords = multisampling.large$coords,
      bw = 32, kernel = "bisquared")

## End(Not run)
```

parse.formula	<i>Parse a HGWR formula.</i>
---------------	------------------------------

Description

This function accept an R formula object and extract names of the group variable, local fixed effects, global fixed effects and random effects.

Usage

```
parse.formula(formula)

## S3 method for class 'push'
stack(s, x)

## S3 method for class 'pop'
stack(s)
```

Arguments

formula	A formula object. Its format is much like the formula used in lmer from package "lme4" .
s	A list, vector or any other object which works with function <code>c</code>
x	An object which can be appended to s.

Value

A list consists of: - response: name of dependent (response) variable. - group: name of group variable. - random.effects: a vector of names of random effects. - fixed.effects: a vector of names of fixed effects.

print.hgwrn	<i>Print description of a hgwrn object.</i>
-------------	---

Description

Print description of a hgwrn object.

Usage

```
## S3 method for class 'hgwrn'
print(x, decimal.fmt = "%.6f", ...)
```

Arguments

`x` An hgwrn object returned by `hgwr()`.

`decimal.fmt` The format string passing to `base::sprintf()`.

`...` Arguments passed on to `print.table.md`

`col.sep` Column separator. Default to `" "`.

`header.sep` Header separator. Default to `"-"`.

`row.begin` Character at the beginning of each row. Default to `col.sep`.

`row.end` Character at the ending of each row. Default to `col.sep`.

`table.style` Name of pre-defined style. Possible values are "plain", "md" or "latex". Default to "plain".

Value

No return.

See Also

`summary.hgwrn()`, `print.table.md()`.

Examples

```
data(multisampling)
model <- hgwr(formula = y ~ g1 + g2 + x1 + (z1 | group),
             data = multisampling$data,
             local.fixed = c("g1", "g2"),
             coords = multisampling$coords,
             bw = 10)
print(model)
print(model, table.style = "md")
```

`print.summary.hgwrn` *Print summary of an hgwrn object.*

Description

Print summary of an hgwrn object.

Usage

```
## S3 method for class 'summary.hgwrn'
print(x, decimal.fmt = "%.6f", ...)
```

Arguments

x An object returned from `summary.hgwr()`.
decimal.fmt The format string passing to `base::sprintf()`.
... Arguments passed on to `print.table.md`
col.sep Column separator. Default to `" "`.
header.sep Header separator. Default to `"-"`.
row.begin Character at the beginning of each row. Default to `col.sep`.
row.end Character at the ending of each row. Default to `col.sep`.
table.style Name of pre-defined style. Possible values are `"plain"`, `"md"` or `"latex"`. Default to `"plain"`.

Value

No return.

See Also

`summary.hgwr()`, `print.table.md()`.

Examples

```

data(multisampling)
model <- hgwr(formula = y ~ g1 + g2 + x1 + (z1 | group),
              data = multisampling$data,
              local.fixed = c("g1", "g2"),
              coords = multisampling$coords,
              bw = 10)
summary(model)

```

print.table.md *Print a character matrix as a table.*

Description

Print a character matrix as a table.

Usage

```

## S3 method for class 'table.md'
print(
  x,
  col.sep = "",
  header.sep = "",
  row.begin = "",
  row.end = "",

```

```

table.style = c("plain", "md", "latex"),
...
)

```

Arguments

<code>x</code>	A character matrix.
<code>col.sep</code>	Column separator. Default to <code>" "</code> .
<code>header.sep</code>	Header separator. Default to <code>"-"</code> .
<code>row.begin</code>	Character at the beginning of each row. Default to <code>col.sep</code> .
<code>row.end</code>	Character at the ending of each row. Default to <code>col.sep</code> .
<code>table.style</code>	Name of pre-defined style. Possible values are "plain", "md" or "latex". Default to "plain".
<code>...</code>	Additional style control arguments.

Details

When `table.style` is specified, `col.sep`, `header.sep`, `row.begin` and `row.end` would not take effects. Because this function will automatically set their values. For each possible value of `table.style`, its corresponding style settings are shown in the following table.

	plain	md	latex
<code>col.sep</code>	<code>" "</code>	<code>" "</code>	<code>"&"</code>
<code>header.sep</code>	<code>" "</code>	<code>"_"</code>	<code>" "</code>
<code>row.begin</code>	<code>" "</code>	<code>" "</code>	<code>" "</code>
<code>row.end</code>	<code>" "</code>	<code>" "</code>	<code>"\\ "</code>

In this function, characters are right padded by spaces.

Value

No return.

See Also

[print.hgwrn\(\)](#), [summary.hgwrn\(\)](#).

residuals.hgwrn

Get residuals.

Description

Get residuals.

Usage

```
## S3 method for class 'hgwrn'
residuals(object, ...)
```

Arguments

object An hgwrn object returned by [hgwr\(\)](#).
 ... Parameter received from other functions.

Value

A vector consists of residuals.

See Also

[hgwr\(\)](#), [summary.hgwrn\(\)](#), [coef.hgwrn\(\)](#) and [fitted.hgwrn\(\)](#).

summary.hgwrn	<i>Summary an hgwrn object.</i>
---------------	---------------------------------

Description

Summary an hgwrn object.

Usage

```
## S3 method for class 'hgwrn'
summary(object, ...)
```

Arguments

object An hgwrn object returned from [hgwr\(\)](#).
 ... Other arguments passed from other functions.

Value

A list containing summary informations of this hgwrn object with the following fields.

diagnostic A list of diagnostic information.

random.stddev The standard deviation of random effects.

random.corr The correlation matrix of random effects.

residuals The residual vector.

See Also

[hgwr\(\)](#).

wuhan.hp

Wuhan Second-hand House Price and POI Data (DataFrame)

Description

A data set of second-hand house price in Wuhan, China collected in 2018.

Usage

```
data(multisampling)
```

Format

A list of two items called "data" and "coords". Item "data" is a data frame with 13862 second-hand properties at 779 neighbourhoods and the following 22 variables.

Price House price per square metre.

Floor.High 1 if a property is on a high floor, otherwise 0.

Floor.Low 1 if a property is on a low floor, otherwise 0.

Decoration.Fine 1 if a property is well decorated, otherwise 0.

PlateTower 1 if a property is of the plate-tower type, otherwise 0.

Steel 1 if a property is of 'steel' structure, otherwise 0.

BuildingArea Building area in square metres.

Fee Management fee per square meter per month.

d.Commercial Distance to the nearest commercial area.

d.Greenland Distance to the nearest green land.

d.Water Distance to the nearest river or lake.

d.University Distance to the nearest university.

d.HighSchool Distance to the nearest high school.

d.MiddleSchool Distance to the nearest middle school.

d.PrimarySchool Distance to the nearest primary school.

d.Kindergarten Distance to the nearest kindergarten.

d.SubwayStation Distance to the nearest subway station.

d.Supermarket Distance to the nearest supermarket.

d.ShoppingMall Distance to the nearest shopping mall.

lon Longitude coordinates (Projected CRS: EPSG 3857).

lat Latitude coordinates (Projected CRS: EPSE 3857).

group Group id of each sample.

The following variables are group level:

- Fee - d.Commercial - d.Greenland - d.Water - d.University - d.HighSchool - d.MiddleSchool
- d.PrimarySchool - d.Kindergarten - d.SubwayStation - d.Supermarket - d.ShoppingMall

The following variables are sample level:

- Price - Floor.High - Floor.Low - Decoration.Fine - PlateTower - Steel - BuildingArea

Item "coords" is a 779-by-2 matrix of coordinates of all neighbourhoods.

Author(s)

Yigong Hu <yigong.hu@bristol.ac.uk>

Examples

```
## Not run:
data(wuhan.hp)
hgwr(
  formula = Price ~ d.Water + d.Commercial + d.PrimarySchool +
             d.Kindergarten + Fee + BuildingArea + (Floor.High | group),
  data = wuhan.hp$data,
  local.fixed = c("d.Water", "d.Commercial", "d.PrimarySchool",
                 "d.Kindergarten", "Fee"),
  coords = wuhan.hp$coords, bw = 50, kernel = "bisquared")

## End(Not run)
```

Index

- * **package**
 - hgwr-package, 2
- base::sprintf(), 6, 7, 10, 11
- c, 9
- coef.hgwr, 3
- coef.hgwr(), 4, 13
- fitted.hgwr, 4
- fitted.hgwr(), 4, 13
- hgwr, 4
- hgwr(), 3, 4, 10, 13
- hgwr (hgwr-package), 2
- hgwr-package, 2
- lmer, 5, 9
- matrix2char, 6
- multisampling, 7
- multisampling.large, 8
- parse.formula, 9
- print.hgwr, 9
- print.hgwr(), 7, 12
- print.summary.hgwr, 10
- print.summary.hgwr(), 7
- print.table.md, 10, 11, 11
- print.table.md(), 10, 11
- residuals.hgwr, 12
- residuals.hgwr(), 4
- stack.pop (parse.formula), 9
- stack.push (parse.formula), 9
- summary.hgwr, 13
- summary.hgwr(), 4, 10–13
- wuhan.hp, 14