Package: spind (via r-universe)

October 24, 2024

Title Spatial Methods and Indices

Version 2.2.1

Author Gudrun Carl [cre, aut], Ingolf Kuehn [aut], Sam Levin [aut]

Maintainer Sam Levin <`levisc8@gmail.com`>

URL `https://github.com/levisc8/spind`

BugReports `https://github.com/levisc8/spind/issues`

Description Functions for spatial methods based on generalized estimating equations (GEE) and wavelet-revised methods (WRM), functions for scaling by wavelet multiresolution regression (WMRR), conducting multi-model inference, and stepwise model selection. Further, contains functions for spatially corrected model accuracy measures.

Depends R (>= 3.0.0)

Imports gee (>= 4.13.19), geepack (>= 1.2.1), MASS (>= 7.3.49), splancs (>= 2.1.40), lattice (>= 0.20.35), waveslim (>= 1.7.5), rje (>= 1.9), stringr (>= 1.3.1), ggplot2 (>= 3.0.0), RColorBrewer (>= 1.1.2), rlang(>= 0.2.1)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr (>= 1.15.1), rmarkdown (>= 1.9), gridExtra (>= 2.3), testthat (>= 2.0.0), PresenceAbsence (>= 1.1.9), sp (>= 1.2.7), covr (>= 3.1.0)

VignetteBuilder knitr

Repository https://levisc8.r-universe.dev

RemoteUrl https://github.com/levisc8/spind

RemoteRef HEAD

RemoteSha 427df394b4ea348ae07b6372958b2f05c327e9dd

Contents

`acfft` *Spatial autocorrelation diagnostics*

Description

A function for calculating spatial autocorrelation using Moran's I.

Usage

```
acfft(coord, f, lim1 = 1, lim2 = 2, dmax = 10)
```

Arguments

`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`f`	A vector which is the same length as `x` and `y`
`lim1`	Lower bound for first bin. Default is 1
`lim2`	Upper bound for first bin. Default is 2
`dmax`	Number of distance bins to examine. Bins are formed by annuli of gradually increasing radii. Default is 10.

Value

A vector of Moran's I values for each distance bin.

Author(s)

Gudrun Carl

Examples

```
data(musdata)
coords <- musdata[ ,4:5]
mglm <- glm(musculus ~ pollution + exposure, "poisson", musdata)

ac <- acfft(coords, resid(mglm, type = "pearson"), lim1 = 0, lim2 = 1)
ac
```

`adjusted.actuals` *Adjusted actual values*

Description

Adjusts actual presence/absence data based on the autocorrelation in the predictions of a model. The function will optionally plot results of model predictions, un-modified actual presence/absence, and adjusted values.

Usage

```
adjusted.actuals(data, coord, plot.maps = FALSE, color.maps = FALSE)
```

Arguments

`data`	a dataframe or matrix containing actual presence/absence (binary, 0 or 1) values in 1st column and predicted values (numeric between 0 and 1) in 2nd column.
`coord`	a matrix of two columns of the same length providing integer, consecutively numbered coordinates for each occurrence and prediction in `data`.
`plot.maps`	A logical indicating whether maps should be plotted. Default is FALSE.
`color.maps`	A logical value. If `TRUE`, produces colorful maps. If `FALSE`, produces grayscale maps. Default is grayscale. NOW DEPRECATED, color maps will not be produced in future versions.

Value

A vector of adjusted actual values.

Author(s)

Gudrun Carl

Examples

```
data(hook)
data <- hook[ ,1:2]
coord <- hook[ ,3:4]
aa <- adjusted.actuals(data, coord, plot.maps = TRUE)
```

`aic.calc` *Akaike Information Criterion with correction for sample size*

Description

Calculates AIC and AICc

Usage

```
aic.calc(formula, family, data, mu, n.eff = NULL)
```

Arguments

`formula`	A model formula
`family`	Family used to fit the model. `gaussian`, `binomial`, or `poisson` are supported
`data`	A data frame
`mu`	Fitted values from a model
`n.eff`	Effective number of observations. Default is NULL

Value

A list with the following components

`loglik` Log likelihood of the model

`df` Degrees of freedom

`AIC` AIC score for the specified model

`AICc` AIC score corrected for small sample sizes

Author(s)

Gudrun Carl, Sam Levin

Examples

```
data(musdata)
coords <- musdata[ ,4:5]
mglm <- glm(musculus ~ pollution + exposure, "poisson", musdata)

aic <- aic.calc(musculus ~ pollution + exposure, "poisson", musdata,
                mglm$fitted)
aic$AIC
```

`carlinadata` *Carlina data set*

Description

A data frame containing simulated count data for the thistle, Carlina horrida.

Usage

```
carlinadata
```

Format

A data frame with 961 rows and 5 columns

carlina.horrida integer - Simulated count data

aridity numeric - Simulated aridity index values. This variable has high spatial autocorrelation values.

land.use numeric - Simulated land use intensity. This variable has no spatial autocorrelation.

x integer - x-coordinates for each grid cell

y integer - y-coordinates for each grid cell

`covar.plot` *Plot wavelet variance/covariance*

Description

Plots the wavelet variance or covariance for the specified formula. The scale-dependent results are graphically displayed.

Usage

```
covar.plot(
  formula,
  data,
  coord,
  wavelet = "haar",
  wtrafo = "dwt",
  plot = "covar",
  customize_plot = NULL
)
```

Arguments

`formula`	With specified notation according to names in data frame.
`data`	Data frame.
`coord`	A matrix of 2 columns with corresponding x,y-coordinates which have to be integer.
`wavelet`	Type of wavelet: `haar`, `d4`, or `la8`.
`wtrafo`	Type of wavelet transform: `dwt` or `modwt`.
`plot`	Either `var` for wavelet variance analysis or `covar` for wavelet covariance analysis.
`customize_plot`	Additional plotting parameters passed to `ggplot`. NOW DEPRECATED

Details

Each variable or pair of variables in `formula` is passed to `wavevar` or `wavecovar` internally, and the result is plotted as a function of `level`.

Value

A list containing

1. `result` = a vector of results.
2. `plot` = a `ggplot` object

Author(s)

Gudrun Carl

See Also

`wavevar`, `wavecovar`

Examples

```
data(carlinadata)
coords <- carlinadata[,4:5]

covariance <- covar.plot(carlina.horrida ~ aridity + land.use - 1,
                         data = carlinadata,
                         coord = coords,
                         wavelet = "d4",
                         wtrafo = 'modwt',
                         plot = 'covar')

covariance$plot
covariance$result

variance <- covar.plot(carlina.horrida ~ aridity + land.use - 1,
                       data = carlinadata,
                       coord = coords,
```

```
                     wavelet = "d4",
                     wtrafo = 'modwt',
                     plot = 'var')

variance$plot
variance$result
```

`GEE` *GEE (Generalized Estimating Equations)*

Description

`GEE` provides GEE-based methods from the packages **gee** and **geepack** to account for spatial autocorrelation in multiple linear regressions

Usage

```
GEE(
  formula,
  family,
  data,
  coord,
  corstr = "fixed",
  cluster = 3,
  moran.params = list(),
  plot = FALSE,
  scale.fix = FALSE,
  customize_plot = NULL
)

## S3 method for class 'GEE'
plot(x, ...)

## S3 method for class 'GEE'
predict(object, newdata, ...)

## S3 method for class 'GEE'
summary(object, ..., printAutoCorPars = TRUE)
```

Arguments

`formula`	Model formula. Variable names must match variables in `data`.
`family`	`gaussian`, `binomial`, or `poisson` are supported. Called using a quoted character string (i.e. `family` = "gaussian").
`data`	A data frame with variable names that match the variables specified in `formula`.

`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`corstr`	Expected autocorrelation structure: `independence`, `fixed`, `exchangeable`, and `quadratic` are possible. • `independence` - This is the same as a GLM, i.e. correlation matrix = identity matrix. • `fixed` - Uses an adapted isotropic power function specifying all correlation coefficients. • `exchangeable` and `quadratic` for clustering, i.e. the correlation matrix has a block diagonal form: – `exchangeable` - All intra-block correlation coefficients are equal. – `quadratic` - Intra-block correlation coefficients for different distances can be different.
`cluster`	Cluster size for cluster models `exchangeable` and `quadratic`. Values of 2, 3, and 4 are allowed. • 2 - a 2*2 cluster • 3 - a 3*3 cluster • 4 - a 4*4 cluster
`moran.params`	A list of parameters for calculating Moran's I. • `lim1` Lower limit for first bin. Default is 0. • `increment` Step size for calculating I. Default is 1.
`plot`	A logical value indicating whether autocorrelation of residuals should be plotted. NOW DEPRECATED in favor of `plot.GEE` method.
`scale.fix`	A logical indicating whether or not the scale parameter should be fixed. The default is `FALSE`. Use `TRUE` when planning to use stepwise model selection procedures in `step.spind`.
`customize_plot`	Additional plotting parameters passed to `ggplot`. NOW DEPRECATED in favor `plot.GEE` method.
`x`	An object of class `GEE` or `WRM`
`...`	Not used.
`object`	An object of class `GEE`.
`newdata`	A data frame containing variables to base the predictions on.
`printAutoCorPars`	A logical indicating whether to print the working autocorrelation parameters

Details

GEE can be used to fit linear models for response variables with different distributions: `gaussian`, `binomial`, or `poisson`. As a spatial model, it is a generalized linear model in which the residuals may be autocorrelated. It accounts for spatial (2-dimensional) autocorrelation of the residuals in cases of regular gridded datasets and returns corrected parameter estimates. The grid cells are assumed to be square. Furthermore, this function requires that **all predictor variables be continuous**.

Value

An object of class `GEE`. This consists of a list with the following elements:

`call`	Call
`formula`	Model formula
`family`	Family
`coord`	Coordinates used for the model
`corstr`	User-selected correlation structure
`b`	Estimate of regression parameters
`s.e.`	Standard errors of the estimates
`z`	Depending on the `family`, either a z or t value
`p`	p-values for each parameter estimate
`scale`	Scale parameter (dispersion parameter) of the distribution's variance
`scale.fix`	Logical indicating whether `scale` has fixed value
`cluster`	User-specified cluster size for clustered models
`fitted`	Fitted values from the model
`resid`	Normalized Pearson residuals
`w.ac`	Working autocorrelation parameters
`Mat.ac`	Working autocorrelation matrix
`QIC`	Quasi Information Criterion. See `qic.calc` for further details
`QLik`	Quasi-likelihood. See `qic.calc` for further details
`plot`	Logical value indicating whether autocorrelation should be plotted
`moran.params`	Parameters for calculating Moran's I
`v2`	Parameter variance of the `GEE` model
`var.naive`	Parameter variance of the `independence` model
`ac.glm`	Autocorrelation of GLM residuals
`ac.gee`	Autocorrelation of GEE residuals
`plot`	An object of class `ggplot` containing information on the autocorrelation of residuals from the fitted `GEE` and a `GLM`

Elements can be viewed using the `summary.GEE` methods included in the package.

Note

When using `corstr = "fixed"` on large data sets, the function may return an error, as the resulting variance-covariance matrix is too large for R to handle. If this happens, one will have to use one of the cluster models (i.e `quadratic`, `exchangeable`).

Author(s)

Gudrun Carl, Sam Levin

References

Carl G & Kuehn I, 2007. Analyzing Spatial Autocorrelation in Species Distributions using Gaussian and Logit Models, Ecol. Model. 207, 159 - 170

Carey, V. J., 2006. Ported to R by Thomas Lumley (versions 3.13, 4.4, version 4.13)., B. R. gee: Generalized Estimation Equation solver. R package version 4.13-11.

Yan, J., 2004. geepack: Generalized Estimating Equation Package. R package version 0.2.10.

See Also

`qic.calc`, `summary.GEE`, `gee`

Examples

```
data(musdata)
coords<- musdata[,4:5]

## Not run:
mgee <- GEE(musculus ~ pollution + exposure,
            family = "poisson",
            data =  musdata,
            coord = coords,
            corstr = "fixed",
            scale.fix = FALSE)

summary(mgee, printAutoCorPars = TRUE)

pred <- predict(mgee, newdata = musdata)

library(ggplot2)

plot(mgee)

my_gee_plot <- mgee$plot

# move the legend to a new position
print(my_gee_plot + ggplot2::theme(legend.position = 'top'))


## End(Not run)
```

---

hook *Hook data set*

Description

A data frame containing actual presence absence data and predicted probability of occurrence values.

Usage

```
hook
```

Format

A data frame with 100 rows and 4 columns

actuals integer - Presence/absence records

predictions numeric - predicted probabilities of occurrence

x integer - x-coordinates for each grid cell

y integer - y-coordinates for each grid cell

`mmiGEE` *Multi-model inference for GEE models*

Description

mmiGEE is a multimodel inference approach evaluating the relative importance of predictors used in `GEE`.

@details It performs automatically generated model selection and creates a model selection table according to the approach of multi-model inference (Burnham & Anderson, 2002). QIC is used to obtain model selection weights and to rank the models. Moreover, mmiGEE calculates relative variable importance of a given model. Finally, this function requires that **all predictor variables be continuous**.

Usage

```
mmiGEE(object, data, trace = FALSE)
```

Arguments

`object`	A model of class `GEE`.
`data`	A data frame or set of vectors of equal length.
`trace`	A logical indicating whether or not to print results to console.

Details

Calculates the relative importance of each variable using multi-model inference methods in a Generalized Estimating Equations framework implemented in `GEE`.

Value

`mmiGEE` returns a list containing the following elements

`result` A matrix containing slopes, degrees of freedom, quasilikelihood, QIC, delta, and weight values for the set of candidate models. The models are ranked by QIC.

`rvi` A vector containing the relative importance of each variable in the regression.

Author(s)

Gudrun Carl, Sam Levin

References

Burnham, K.P. & Anderson, D.R. (2002) Model selection and multimodel inference. Springer, New York.

Carl G & Kuehn I, 2007. Analyzing Spatial Autocorrelation in Species Distributions using Gaussian and Logit Models, Ecol. Model. 207, 159 - 170

See Also

`GEE`, `qic.calc`, **MuMIn**

Examples

```
# data (for demonstration only)
library(MASS)
data(birthwt)

# impose an artificial (not fully appropriate) grid structure

x <- rep(1:14, 14)
y <- as.integer(gl(14, 14))
coords <- cbind(x[-(190:196)], y[-(190:196)])

## Not run:

formula <- formula(low ~ race + smoke +  bwt)

mgee <- GEE(formula,
            family = "gaussian",
            data = birthwt,
            coord = coords,
            corstr = "fixed",
            scale.fix = TRUE)

mmi <- mmiGEE(mgee, birthwt)


## End(Not run)
```

`mmiWMRR` *Multi-model inference for wavelet multiresolution regression*

Description

mmiWMRR is a multimodel inference approach evaluating the relative importance of predictors used in `scaleWMRR`.

Usage

```
mmiWMRR(object, data, scale, detail = TRUE, trace = FALSE)
```

Arguments

`object`	A model of class `WRM`.
`data`	Data frame.
`scale`	0 or higher integers possible (limit depends on sample size). `scale=1` is equivalent to `WRM` with `level=1`.
`detail`	Remove smooth wavelets? If `TRUE`, only detail components are analyzed. If set to `FALSE`, smooth and detail components are analyzed. Default is `TRUE`.
`trace`	Logical value indicating whether to print results to console.

Details

It performs automatically generated model selection and creates a model selection table according to the approach of multi-model inference (Burnham & Anderson, 2002). The analysis is carried out for scale-specific regressions (i.e. where `scaleWMRR` can be used). AIC is used to obtain model selection weights and to rank the models. Furthermore, this function requires that **all predictor variables be continuous**.

Value

`mmiWMRR` returns a list containing the following elements

`result` A matrix containing slopes, degrees of freedom, likelihood, AIC, delta, and weight values for the set of candidate models. The models are ranked by AIC.

`level` An integer corresponding to scale

Author(s)

Gudrun Carl

References

Burnham, K.P. & Anderson, D.R. (2002) Model selection and multimodel inference. Springer, New York.

Carl G, Doktor D, Schweiger O, Kuehn I (2016) Assessing relative variable importance across different spatial scales: a two-dimensional wavelet analysis. Journal of Biogeography 43: 2502-2512.

See Also

`aic.calc`, `rvi.plot`, **MuMIn**, `WRM`

Examples

```
data(carlinadata)
coords <- carlinadata[ ,4:5]

## Not run:


wrm <- WRM(carlina.horrida ~ aridity + land.use,
           family = "poisson",
           data = carlinadata,
           coord = coords,
           level = 1,
           wavelet = "d4")

mmi <- mmiWMRR(wrm,
               data = carlinadata,
               scale = 3,
               detail = TRUE,
               trace = FALSE)


## End(Not run)
```

`musdata` *Mus musculus data set*

Description

A data frame containing simulated count data of a house mouse.

Usage

`musdata`

Format

A data frame with 400 rows and 5 columns

musculus integer - Simulated count data for Mus musculus

pollution numeric - Simulated variable that describes degree of pollution in corresponding grid cell

exposure numeric - Simulated variable that describes degree of exposure for each grid cell

x integer - x-coordinates for each grid cell

y integer - y-coordinates for each grid cell

`qic.calc` *Quasi-Information Criterion for Generalized Estimating Equations*

Description

A function for calculating quasi-likelihood and Quasi-Information Criterion values based on the method of Hardin & Hilbe (2003).

Usage

```
qic.calc(formula, family, data, mu, var.robust, var.indep.naive)
```

Arguments

`formula`	a model formula
`family`	`gaussian`, `binomial`, or `poisson`
`data`	a data frame
`mu`	fitted values from a model
`var.robust`	variance of model parameters
`var.indep.naive`	naive variance of model parameters under the `independence` model

Value

A list with the following components:

`QIC`	quasi-information criterion
`loglik`	quasi-likelihood

References

Hardin, J.W. & Hilbe, J.M. (2003) Generalized Estimating Equations. Chapman and Hall, New York.

Barnett et al. Methods in Ecology & Evolution 2010, 1, 15-24.

`rvi.plot` *Relative Variable Importance*

Description

Creates model selection tables, calculates and plots relative variable importance based on the scale level of a given model.

Usage

```
rvi.plot(
  formula,
  family,
  data,
  coord,
  maxlevel,
  detail = TRUE,
  wavelet = "haar",
  wtrafo = "dwt",
  n.eff = NULL,
  trace = FALSE,
  customize_plot = NULL
)
```

Arguments

`formula`	A model formula
`family`	`gaussian`, `binomial`, and `poisson` are supported.
`data`	A data frame or set of vectors of equal length.
`coord`	X,Y coordinates for each observation. Coordinates should be consecutive integers.
`maxlevel`	An integer for maximum scale level
`detail`	Remove smooth wavelets? If `TRUE`, only detail components are analyzed. If set to `FALSE`, smooth and detail components are analyzed. Default is `TRUE`.
`wavelet`	Type of wavelet: `haar`, `d4`, or `la8`
`wtrafo`	Type of wavelet transform: `dwt` or `modwt`
`n.eff`	A numeric value of effective sample size
`trace`	Should R print progress updates to the console? Default is FALSE
`customize_plot`	Additional plotting parameters passed to `ggplot`. NOW DEPRECATED.

Details

Calculates the relative importance of each variable using multi-model inference methods in a wavelet multi-resolution regression framework implemented in `mmiWMRR`. The scale level dependent results are then graphically displayed.

Value

A list containing

1. A matrix containing the relative importance of each variable in the regression at each value of the scale level.

2. A `ggplot` object containing a plot of the relative variable importance

Examples

```
data(carlinadata)
coords<- carlinadata[,4:5]

## Not run:

wrm <- WRM(carlina.horrida ~ aridity + land.use,
           family = "poisson",
           data = carlinadata,
           coord = coords,
           level = 1,
           wavelet = "d4")

mmi <- mmiWMRR(wrm, data = carlinadata, scale = 3, detail = TRUE)


# Plot scale-dependent relative variable importance
rvi <- rvi.plot(carlina.horrida ~ aridity + land.use,
                family = "poisson",
                data = carlinadata,
                coord = coords,
                maxlevel = 4,
                detail = TRUE,
                wavelet = "d4")

rvi$plot
rvi$rvi


## End(Not run)
```

`scaleWMRR` *Scaling by wavelet multiresolution regression (WMRR)*

Description

scaleWMRR performs a scale-specific regression based on a wavelet multiresolution analysis.

Usage

```
scaleWMRR(
  formula,
  family,
  data,
  coord,
  scale = 1,
  detail = TRUE,
  wavelet = "haar",
  wtrafo = "dwt",
  b.ini = NULL,
  pad = list(),
  control = list(),
  moran.params = list(),
  trace = FALSE
)
```

Arguments

`formula`	With specified notation according to names in data frame.
`family`	`gaussian`, `binomial`, or `poisson`.
`data`	Data frame.
`coord`	Corresponding coordinates which have to be integer.
`scale`	0 (which is equivalent to GLM) or higher integers possible (limit depends on sample size).
`detail`	Remove smooth wavelets? If `TRUE`, only detail components are analyzed. If set to `FALSE`, smooth and detail components are analyzed. Default is `TRUE`.
`wavelet`	Type of wavelet: `haar` or `d4` or `la8`
`wtrafo`	Type of wavelet transform: `dwt` or `modwt`.
`b.ini`	Initial parameter values. Default is `NULL`.

`pad`	A list of parameters for padding wavelet coefficients.
	• padform - 0, 1, and 2 are possible. `padform` is automatically set to 0 when either `level`=0 or the `formula` includes an intercept and has a non-`gaussian` `family`.
	  – 0 - Padding with 0s.
	  – 1 - Padding with mean values.
	  – 2 - Padding with mirror values.
	• padzone - Factor for expanding the padding zone

`control`	A list of parameters for controlling the fitting process. • eps - Positive convergence tolerance. Smaller values of `eps` provide better parameter estimates, but also reduce the probability of the iterations converging. In case of issues with convergence, test larger values of `eps`. Default is 10^-5. • `denom.eps` - Default is 10^-20.

	- `itmax` - Integer giving the maximum number of iterations. Default is 200.
`moran.params`	A list of parameters for calculating Moran's I. - `lim1` - Lower limit for first bin. Default is 0. - `increment` - Step size for calculating Moran's I. Default is 1.
`trace`	A logical value indicating whether to print parameter estimates to the console

Details

This function fits generalized linear models while taking the two-dimensional grid structure of datasets into account. The following error distributions (in conjunction with appropriate link functions) are allowed: `gaussian`, `binomial`, or `poisson`. The model provides scale-specific results for data sampled on a contiguous geographical area. The dataset is assumed to be regular gridded and the grid cells are assumed to be square. A function from the package 'waveslim' is used for the wavelet transformations (Whitcher, 2005). Furthermore, this function requires that **all predictor variables be continuous**.

Value

scaleWMRR returns a list containing the following elements

`call` Model call

`b` Estimates of regression parameters

`s.e.` Standard errors of the parameter estimates

`z` Z values (or corresponding values for statistics)

`p` p-values for each parameter estimate

`df` Degrees of freedom

`fitted` Fitted values

`resid` Pearson residuals

`converged` Logical value whether the procedure converged

`trace` Logical. If TRUE:

- `ac.glm` Autocorrelation of glm.residuals
- `ac` Autocorrelation of wavelet.residuals

Author(s)

Gudrun Carl

References

Carl G, Doktor D, Schweiger O, Kuehn I (2016) Assessing relative variable importance across different spatial scales: a two-dimensional wavelet analysis. Journal of Biogeography 43: 2502-2512.

Whitcher, B. (2005) Waveslim: basic wavelet routines for one-, two- and three-dimensional signal processing. R package version 1.5.

See Also

waveslim,`mra.2d`

Examples

```
data(carlinadata)
coords <- carlinadata[ ,4:5]

## Not run:

# scaleWMRR at scale = 0 is equivalent to GLM

ms0 <- scaleWMRR(carlina.horrida ~ aridity + land.use,
                 family = "poisson",
                 data = carlinadata,
                 coord = coords,
                 scale = 0,
                 trace = TRUE)

# scale-specific regressions for detail components
ms1 <- scaleWMRR(carlina.horrida ~ aridity + land.use,
                 family = "poisson",
                 data = carlinadata,
                 coord = coords,
                 scale = 1,
                 trace = TRUE)

ms2 <- scaleWMRR(carlina.horrida ~ aridity + land.use,
                 family = "poisson",
                 data = carlinadata,
                 coord = coords,
                 scale = 2,
                 trace = TRUE)

ms3<- scaleWMRR(carlina.horrida ~ aridity + land.use,
                 family = "poisson",
                 data = carlinadata,
                 coord = coords,
                 scale = 3,
                 trace = TRUE)


## End(Not run)
```

`spind` *spind: Spatial Methods and Indices*

Description

The `spind` package provides convenient implementation of Generalized estimating equations (GEEs) and Wavelet-revised models (WRMs) in the context of spatial models. It also provides tools for multi-model inference, stepwise model selection, and spatially corrected model diagnostics. This help section provides brief descriptions of each function and is organized by the type of model they apply to or the scenarios in which you might use them. Of course, these are recommendations - feel free to use them as you see fit. For a more detailed description of the package and its functions, please see the vignette *Intro to spind* (`browseVignettes('spind')`).

GEEs

The `GEE` function fits spatial models using a generalized estimating equation and a set of gridded data. The package also includes S3 methods for `summary` and `predict` so you can interact with these models in the same way you might interact with a `glm` or `lm`.

WRMs

The `WRM` function fits spatial models using a wavelet-revised model and a set of gridded data. The package also includes S3 methods for `summary` and `predict` so you can interact with these models in the same way you might interact with a `glm` or `lm`. There are also a number of helper functions that help you fine tune the fitting process that are specific to WRMs. Please see the documentation for `WRM` for more details on those.

`WRM` also has a few other features specific to it. For example, if you are interested in viewing the variance or covariance of your variables as a function of `level`, `covar.plot` is useful. `upscale` will plot your matrices as a function of `level` so you can examine the effect of cluster resolution on your results.

Multi-model inference and stepwise model selection

`spind` includes a couple of functions to help you find the best fit for your data. The first two are multimodel inference tools specific to GEEs and WRMs and are called `mmiGEE` and `mmiWMRR`. These generate outputs very similar to those from the `MuMIn` package. If you would like to see how variable importance changes as a function of the `scale` of the WMRR, you can call `rvi.plot`. This will generate a model selection table for each degree of `level` (from 1 to `maxlevel`) and then plot the weight of each variable as a function of `level`.

`spind` also includes a function for stepwise model selection that is loosely based on `step` and `stepAIC`. `step.spind` differs from these in that it is specific to classes `WRM` and `GEE`. It performs model selection using AIC or AICc for WRMs and QIC for GEEs.

Spatial indices of goodness of fit

Finally, `spind` has a number of functions that provide spatially corrected goodness of fit diagnostics for any type of model (i.e. they are not specific to classes `WRM` or `GEE`). These first appeared in `spind v1.0` and have not been updated in this release. The first two are divided into whether or not they are threshold dependent or not. Threshold dependent metrics can be calculated using `th.dep` and threshold independent metrics can be calculated using `th.indep`.

`acfft` calculates spatial autocorrelation of residuals from a model using *Moran's I*. You can set the number of distance bins you'd like to examine using `dmax` argument and the size of those bins using `lim1` and `lim2`.

Conclusion

The vignette titled *Intro to spind* provides more information on these functions and some example workflows that will demonstrate them in greater depth than this document. Of course, if you have suggestions on how to improve this document or any of the other ones in here, please don't hesitate to contact us.

`step.spind` *Stepwise model selection for GEEs and WRMs*

Description

Stepwise model selection by AIC or AICc for WRMS and QIC for GEEs

Usage

```
step.spind(object, data, steps = NULL, trace = TRUE, AICc = FALSE)
```

Arguments

`object`	A model of class `WRM` or `GEE`.
`data`	The data used to fit that model.
`steps`	Number of iterations the procedure should go through before concluding. The default is to use the number of variables as the number of iterations.
`trace`	Should R print progress updates and the final, best model found to the console? Default is `TRUE`.
`AICc`	Logical. In the case of model selection with `WRM`s, should AICc be used to determine which model is best rather than AIC? This argument is ignored for `GEE`s. Default is `FALSE`.

Details

This function performs stepwise variable elimination for model comparison. Each iteration will try to find the best combination of predictors for a given number of variables based on AIC, AICc, or QIC, and then use that as the base model for the next iteration until there are no more variables to eliminate. Alternatively, it will terminate when reducing the number of variables while respecting the model hierarchy no longer produces lower information criterion values.

Value

A list with components `model` and `table`. `model` is always formula for the best model found by the procedure. `table` is always a data frame, but the content varies for each type of model. For `WRM`'s, the columns returned are

- `Deleted.Vars` Variables retained from the previous iteration which were tested in the current iteration.
- `LogLik` Log-likelihood of the model.
- `AIC` AIC score for the model.
- `AICc` AICc score for the model.

For `GEE`s:

- `Deleted.Vars` Variables retained from the previous iteration which were tested in the current iteration.
- `QIC` Quasi-information criterion of the model.
- `Quasi.Lik` Quasi-likelihood of the model.

Note

Currently, the function only supports backwards model selection (i.e. one must start with a full model and subtract variables). Forward and both directions options may be added later.

Author(s)

Sam Levin

References

Hardin, J.W. & Hilbe, J.M. (2003) Generalized Estimating Equations. Chapman and Hall, New York.

See Also

`qic.calc`, `aic.calc`, `add1`, `step`, `stepAIC`

Examples

```
# For demonstration only. We are artificially imposing a grid structure
# on data that is not actually spatial data

library(MASS)
data(birthwt)

x <- rep(1:14, 14)
y <- as.integer(gl(14, 14))
coords <- cbind(x[-(190:196)], y[-(190:196)])

## Not run:
formula <- formula(low ~ age + lwt + race + smoke + ftv + bwt)
```

```
mgee <- GEE(formula,
            family = "gaussian",
            data = birthwt,
            coord = coords,
            corstr = "fixed",
            scale.fix = TRUE)

ss <- step.spind(mgee, birthwt)

best.mgee <- GEE(ss$model,
                 family = "gaussian",
                 data = birthwt,
                 coord = coords,
                 corstr = "fixed",
                 scale.fix = TRUE)

summary(best.mgee, printAutoCorPars = FALSE)

## End(Not run)
```

`th.dep` *Spatial threshold-dependent accuracy measures*

Description

Calculates spatially corrected, threshold-dependent metrics for an observational data set and model predictions (Kappa and confusion matrix)

Usage

```
th.dep(data, coord, thresh = 0.5, spatial = TRUE)
```

Arguments

`data`	A data frame or matrix with two columns. The first column should contain actual presence/absence data (binary, 0 or 1) and the second column should contain model predictions of probability of occurrence (numeric, between 0 and 1).
`coord`	A data frame or matrix with two columns containing x,y coordinates for each actual and predicted value. Coordinates must be integer and consecutively numbered.
`thresh`	A cutoff value for classifying predictions as modeled presences or modeled absences. Default is 0.5.
`spatial`	A logical indicating whether spatially corrected indices (rather than classical indices) should be computed.

Value

A list with the following components:

`kappa`	Kappa statistic
`cm`	Confusion matrix
`sensitivity`	Sensitivity
`specificity`	Specificity
`actuals`	Actual occurrence data or adjusted actual occurrence data
`splitlevel.pred`	Level splitting of predicted values
`splitlevel.act`	Level splitting of actuals or adjusted actuals
`splitposition.pred`	Position splitting of predicted values
`splitposition.act`	Position splitting of actuals or adjusted actuals

Author(s)

Gudrun Carl

References

Carl G, Kuehn I (2017) Spind: a package for computing spatially corrected accuracy measures. Ecography 40: 675-682. DOI: 10.1111/ecog.02593

See Also

`th.indep`

Examples

```
data(hook)
data <- hook[ ,1:2]
coord <- hook[ ,3:4]
si1 <- th.dep(data, coord, spatial = TRUE)
si1$kappa
si1$cm
```

`th.indep` *Spatial threshold-independent accuracy measures*

Description

Calculates spatially corrected, threshold-independent metrics for an observational data set and model predictions (AUC, ROC, max-TSS)

Usage

```
th.indep(data, coord, spatial = TRUE, plot.ROC = FALSE, customize_plot = NULL)
```

Arguments

`data`	A data frame or matrix with two columns. The first column should contain actual presence/absence data (binary, 0 or 1) and the second column should contain model predictions of probability of occurrence (numeric, between 0 and 1).
`coord`	A data frame or matrix with two columns containing x,y coordinates for each actual and predicted value. Coordinates must be integer and consecutively numbered.
`spatial`	A logical value indicating whether spatial corrected indices (rather than classical indices) should be computed.
`plot.ROC`	A logical indicating whether the ROC should be plotted. NOW DEPRECATED.
`customize_plot`	Additional plotting parameters passed to `ggplot`. NOW DEPRECATED.

Value

A list with the following components:

`AUC` Area under curve

`opt.thresh` optimal threshold for maximum TSS value

`TSS` Maximum TSS value

`sensitivity` Sensitivity

`Specificity` Specificity

`AUC.plot` A ggplot object

Author(s)

Gudrun Carl

References

Carl G, Kuehn I (2017) Spind: a package for computing spatially corrected accuracy measures. Ecography 40: 675-682. DOI: 10.1111/ecog.02593

See Also

`th.dep`

Examples

```
data(hook)
data <- hook[ ,1:2]
coord <- hook[ ,3:4]
si2 <- th.indep(data, coord, spatial = TRUE)
si2$AUC
```

```
si2$TSS
si2$opt.thresh
si2$plot
```

---

upscale *Upscaling of smooth components*

Description

The analysis is based a wavelet multiresolution analysis using only smooth wavelet components. It is a 2D analysis taking the grid structure and provides scale-specific results for data sampled on a contiguous geographical area. The dataset is assumed to be regular gridded and the grid cells are assumed to be square. The scale-dependent results are graphically displayed.

Usage

```
upscale(
  f,
  coord,
  wavelet = "haar",
  wtrafo = "dwt",
  pad = mean(f),
  color.maps = FALSE
)
```

Arguments

`f`	A vector.
`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`wavelet`	Name of wavelet family. `haar`, `d4`, and `la8`. are possible. `haar` is the default.
`wtrafo`	Type of wavelet transform. Either `dwt` or `modwt`. `dwt` is the default.
`pad`	A numeric value for padding the matrix into a bigger square. Default is set to mean(f).
`color.maps`	A logical value. If `TRUE`, produces colorful maps. If `FALSE`, produces grayscale maps. Default is grayscale. NOW DEPRECATED, color maps will not be produced in future versions.

Value

A set of plots showing the matrix image at each value for `level`.

Author(s)

Gudrun Carl

Examples

```
data(carlinadata)
coords <- carlinadata[ ,4:5]

# Upscaling of smooth components
upscale(carlinadata$land.use, coord = coords)
```

`wavecovar` *Wavelet covariance analysis*

Description

Calculates the wavelet covariance based on a wavelet multiresolution analysis.

Usage

```
wavecovar(f1, f2, coord, wavelet = "haar", wtrafo = "dwt")
```

Arguments

`f1`	A vector of length *n*.
`f2`	A vector of length *n*.
`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`wavelet`	Name of wavelet family. `haar`, `d4`, and `la8`. are possible. `haar` is the default.
`wtrafo`	Type of wavelet transform. Either `dwt` or `modwt`. `dwt` is the default.

Value

Wavelet covariance for `f1` and `f2`.

Author(s)

Gudrun Carl

See Also

waveslim, `WRM`, `covar.plot`, `scaleWMRR`

Examples

```
data(carlinadata)

coords <- carlinadata[ ,4:5]
pc <- covar.plot(carlina.horrida ~ aridity + land.use,
                 data = carlinadata,
                 coord = coords,
                 wavelet = 'd4',
                 wtrafo = 'modwt',
                 plot = 'covar')

pc$plot
```

`wavevar` *Wavelet variance analysis*

Description

Calculates the wavelet variance based on a wavelet multiresolution analysis.

Usage

```
wavevar(f, coord, wavelet = "haar", wtrafo = "dwt")
```

Arguments

`f`	A vector
`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`wavelet`	Name of wavelet family. `haar`, `d4`, and `la8`. are possible. `haar` is the default.
`wtrafo`	Type of wavelet transform. Either `dwt` or `modwt`. `dwt` is the default.

Value

Wavelet variance for `f`.

Author(s)

Gudrun Carl

See Also

waveslim, `WRM`, `covar.plot`, `scaleWMRR`

Examples

```
data(carlinadata)

coords <- carlinadata[ ,4:5]
pv <- covar.plot(carlina.horrida ~ aridity + land.use,
                 data = carlinadata,
                 coord = coords,
                 wavelet = 'd4',
                 wtrafo = 'modwt',
                 plot = 'var')

pv$plot
```

WRM *Wavelet-revised models (WRMs)*

Description

A wavelet-based method to remove spatial autocorrelation in multiple linear regressions. Wavelet transforms are implemented using **waveslim** (Whitcher, 2005).

Usage

```
WRM(
  formula,
  family,
  data,
  coord,
  level = 1,
  wavelet = "haar",
  wtrafo = "dwt",
  b.ini = NULL,
  pad = list(),
  control = list(),
  moran.params = list(),
  plot = FALSE,
  customize_plot = NULL
)

## S3 method for class 'WRM'
plot(x, ...)

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

## S3 method for class 'WRM'
predict(object, newdata, sm = FALSE, newcoord = NA, ...)
```

Arguments

`formula`	Model formula. Variable names must match variables in `data`.
`family`	`gaussian`, `binomial`, or `poisson` are supported.
`data`	A data frame with variable names that match the variables specified in `formula`.
`coord`	A matrix of two columns with corresponding cartesian coordinates. Currently only supports integer coordinates.
`level`	An integer specifying the degree of wavelet decomposition • 0 - Without autocorrelation removal (equivalent to a GLM) • 1 - For best autocorrelation removal • ... - Higher integers possible. The limit depends on sample size
`wavelet`	Name of wavelet family. `haar`, `d4`, and `la8`. are possible. `haar` is the default.
`wtrafo`	Type of wavelet transform. Either `dwt` or `modwt`. `dwt` is the default.
`b.ini`	Initial parameter values. Default is NULL.

`pad`	A list of parameters for padding wavelet coefficients.
	• padform - 0, 1, and 2 are possible. `padform` is automatically set to 0 when either `level`=0 or a `formula` including an intercept and a non-gaussian family
	  – 0 - Padding with 0s.
	  – 1 - Padding with mean values.
	  – 2 - Padding with mirror values.
	• padzone - Factor for expanding the padding zone

`control`	a list of parameters for controlling the fitting process. • `eps` - Positive convergence tolerance. Smaller values of `eps` provide better parameter estimates, but also reduce the probability of the iterations converging. In case of issues with convergence, test larger values of `eps`. Default is 10^-5. • `denom.eps` - Default is 10^-20. • `itmax` - Integer giving the maximum number of iterations. Default is 200.
`moran.params`	A list of parameters for calculating Moran's I. • `lim1` - Lower limit for first bin. Default is 0. • `increment` - Step size for calculating Moran's I. Default is 1.
`plot`	A logical value indicating whether to plot autocorrelation of residuals by distance bin. NOW DEPRECATED in favor of `plot.WRM` method.
`customize_plot`	Additional plotting parameters passed to `ggplot`. NOW DEPRECATED in favor of `plot.WRM` method.
`x`	An object of class `GEE` or `WRM`
`...`	Not used
`object`	An object of class `WRM`
`newdata`	A data frame containing variables used to make predictions.
`sm`	Logical. Should part of smooth components be included?
`newcoord`	New coordinates corresponding to observations in `newdata`.

Details

WRM can be used to fit linear models for response vectors of different distributions: `gaussian`, `binomial`, or `poisson`. As a spatial model, it is a generalized linear model in which the residuals may be autocorrelated. It corrects for 2-dimensional residual autocorrelation for regular gridded data sets using the wavelet decomposition technique. The grid cells are assumed to be square. Furthermore, this function requires that **all predictor variables be continuous**.

Value

An object of class `WRM`. This consists of a list with the following elements:

`call` Call

`formula` Model formula

`family` Family

`coord` Coordinates used in the model

`b` Estimate of regression parameters

`s.e.` Standard errors

`z` Depending on the `family`, either a z or t value

`p` p-values

`fitted` Fitted values from the model

`resid` Pearson residuals

`b.sm` Parameter estimates of neglected smooth part

`fitted.sm` Fitted values of neglected smooth part

`level` Selected level of wavelet decomposition

`wavelet` Selected wavelet

`wtrafo` Selected wavelet transformation

`padzone` Selected padding zone expansion factor

`padform` Selected matrix padding type

`n.eff` Effective number of observations

`AIC` Akaike information criterion

`AICc` AIC score corrected for small sample sizes

`LogLik` Log likelihood of the model

`ac.glm` Autocorrelation of GLM residuals

`ac.wrm` Autocorrelation of WRM residuals

`b.ini` Initial parameter values

`control` Control parameters for the fitting process

`moran.params` Parameters for calculating Moran's I

`pad` List of parameters for padding wavelet coefficients

`plot` An object of class `ggplot` containing information on the autocorrelation of residuals from the fitted `WRM` and a `GLM`

Note

For those interested in multimodel inference approaches, `WRM` with `level = 1` is identical to `mmiWMRR` with `scale = 1`.

Author(s)

Gudrun Carl, Sam Levin

References

Carl, G., Kuehn, I. (2010): A wavelet-based extension of generalized linear models to remove the effect of spatial autocorrelation. Geographical Analysis 42 (3), 323 - 337

Whitcher, B. (2005) Waveslim: basic wavelet routines for one-, two- and three-dimensional signal processing. R package version 1.5.

See Also

`mmiWMRR`, `predict.WRM`, `summary.WRM`, `aic.calc`

Examples

```
data(musdata)
coords <- musdata[,4:5]

## Not run:
mwrm <- WRM(musculus ~ pollution + exposure,
            family = "poisson",
            data = musdata,
            coord = coords,
            level = 1)

pred <- predict(mwrm, newdata = musdata)

summary(mwrm)

plot(mwrm)

library(ggplot2)

my_wrm_plot <- mwrm$plot

# increase axis text size
print(my_wrm_plot + ggplot2::theme(axis.text = element_text(size = 15)))


## End(Not run)
```

Index