Package 'VARMER'

February 8, 2021

Type Package

Title Variational Merging

Version 1.0.0

Description A new mathematical formulation to merge observed data with gridded images of environmental variables using partial differential equations in a variational setting. The original method was created, developed and published by Ulloa, Samaniego, Campozano and Ballari (2018) <doi:10.1002/2017JD027982>.

License CC0

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports stats, methods, raster, pracma, rgl, deldir, grDevices, zoo, sf, gstat, Metrics, hydroGOF, tictoc, cluster, parallel

Suggests knitr, rmarkdown, rgdal, RFmerge

VignetteBuilder knitr

RoxygenNote 7.1.1

Language en-US

NeedsCompilation no

Author Jacinto Ulloa [aut],
Esteban Samaniego [aut],
Lenin Campozano [aut],
Daniela Ballari [aut],
Leandro Robaina [cre, com, trl],
Ronald Gualan [com]

Maintainer Leandro Robaina <metclimaepn.libs@gmail.com>

Repository CRAN

Date/Publication 2021-02-07 23:00:03 UTC

R topics documented:

`coordinates` *coordinates data*

Description

A data containing the nodes coordinates of uhat.raster and v.raster.

Usage

```
data(coordinates,"VARMER")
```

Format

A data frame with 4340 rows and 3 columns (x y and index).

ecuador.tmax.stations.df

Spatial location of meteorological stations in Ecuador

Description

Spatial location of the 34 stations with daily maximum temperature for Ecuador (dataset ecuador.tmax.zoo).

Usage

```
data(ecuador.tmax.stations.df)
```

Format

A data.frame with eight fields: *) CODIGO : identifier of each station *) NOMBRE : station name *) CUENCA_INAMHI : basin name *) PROVINCIA : province name *) CANTON : canton name *) PARROQUIA : parish name *) LAT : northing coordinate of the station, EPSG:4326 *) LON : easting coordinate of the station, EPSG:4326

ecuador.tmax.wrf.out *WRF (model-based) output for maximum temperature*

Description

Model-based maximum temperature datasets providing global spatial structure

Usage

```
data(ecuador.tmax.wrf.out)
```

Format

A RasterBrick frame with 10 layers (1 for each day), and a geographical area of 44x47 for a 10Km spatial resolution

ecuador.tmax.zoo *Ground-based observations of maximum temperature for Ecuador*

Description

A zoo data frame containing time series for january 2004 on 34 meteorological stations located in continental Ecuador.

Usage

```
data(ecuador.tmax.zoo)
```

Format

A zoo object with 34 columns (one for each station) and 10 rows (one for each day in jan-2004).

elements *elements data*

Description

A table of elements

Usage

```
data(elements,"VARMER")
```

Format

A data frame with 8418 rows and 3 columns index1 index2 index3.

fit.varmer *Training eta parameter for the varmer function*

Description

Training eta parameter for the varmer function evaluating a vector of etas using Cross-validation. The best eta is the one yielding the highest KGE metric.

Usage

```
fit.varmer(
  stations.sf,
  v,
  etas = c(10, 100, 500, 1000, 5000),
  idw_formula = Variable ~ 1,
  factor_agg = 2,
  drty.out = tempdir(),
  apply_varmer = T
)
```

Arguments

stations.sf	data.frame with the observations metadata
v	grided image
etas	(optional) vector of eta values to evaluate in a CV exercise
idw_formula	formula for the idw interpolation
factor_agg	scalar which defines the aggregation factor to apply to the raster images in order to reduce computation requirements for solving varmer
drty.out	(optional) output folder for the CV metrics
apply_varmer	(optional) boolean which determines if a merging image is produced with the best eta

---

idw.uhat.from.sf *IDW interpolation for varmer*

Description

It produces an IDW-interpolated image from observations

Usage

```
idw.uhat.from.sf(stations.sf, reference.raster, idw.formula)
```

Arguments

stations.sf	data.frame with metadata for the observations
reference.raster	raster image which provides the base structure for the resulting IDW interpolated image
idw.formula	The formula to be passed for the IDW interpolation method

img2varmer *Compare two images*

Description

The function compare two raster or bricks with different z values(uhat and v images) It check if the images has the same extension, resolution, coordinate reference system, columns and rows.

Usage

```
img2varmer(uhat.img, v.img)
```

Arguments

uhat.img	The interpolated image as a raster or brick.
v.img	The satellite image as a raster or brick.

Value

If the images have the same extension, resolution, coordinate reference system, columns and rows.

A list containing two data frames (uhat and v) in the necessary format to varmer function (x,y,layer1,layer2).

If the images has not the same values an error message is returned.

Examples

```
library(raster)
data(uhat_raster,"VARMER")
force(uhat.raster)
data(v_raster,"VARMER")
force(v.raster)
img2varmer(uhat.raster,v.raster)
```

mesh_3 *Generates triangular finite element mesh.*

Description

The function mesh_3 takes as input data the length of the domain in x (Lx), the length of the domain in y (Ly), the number of divisions in x (Nx) and the number of divisions in y (Ny).

These values are obtained with the function prev_data, specifically from coordinates inside the variable uhat (which must be equal to those of v).

Usage

```
mesh_3(Lx, Ly, Nx, Ny)
```

Arguments

Lx, Ly, Nx, Ny	a group of vectors obtained with the function prev_data. The coordinates of the nodes (coordinates) and the table of elements (elements). Following the next structure in columns coordinates: x, y, index elements: index1, index2, index3 A plot with triangles from the coordinates

Examples

```
data(uhat,"VARMER")
invisible(force(uhat))
pre_results=pre_data(uhat)
coordsx=pre_results[[1]][[1]]
coordsx=pre_results[[1]][[2]]
Lx=pre_results[[2]]
Ly=pre_results[[3]]
Nx=pre_results[[4]]
Ny=pre_results[[5]]
mesh_results=mesh_3(Lx,Ly,Nx,Ny)
```

---

plot_mesh *Visualization of the curves*

Description

Using the data of previous functions (mesh_3 and var_merge), three 3D graphs are realized. Each graph represent in order the variation through the space (longitude and latitude) U,uhat and v.

Usage

```
plot_mesh(U, uhat, v, color_plot, eta_0)
```

Arguments

U	A data frame representing resulting images in relative coordinates.
uhat	A matrix containing values from the interpolated image with x y and data and the next column structure x y date1 date2 date3
v	A matrix containing values from the satellite/model based image. The values of U, uhat and v should correspond to the same pixels in the coordinates matrix.
color_plot	A palette of colors from grDevices.
eta_0	A scalar representing length scale constant.

Value

Three 3D plots representing the variation

Examples

```
data(U,"VARMER")
invisible(force(U))
data(uhat,"VARMER")
invisible(force(uhat))
data(v,"VARMER")
invisible(force(v))
eta_0=0.5
plot_mesh(U,uhat,v,topo.colors,eta_0)
```

---

pre_data *Obtain necessary data to the package.*

Description

Using a variable in the needed format the unique coordinates.

The function calculates necessary variables for mesh_3.

Usage

```
pre_data(uhat)
```

Arguments

uhat	A matrix containing values from the interpolated image with x y and data and the next column structure x y date1 date2 date3

Value

A list with vector data to use later with other functions of the package (Lx,Ly,Nx,Ny).

Examples

```
data(uhat,"VARMER")
invisible(force(uhat))
pre_results=pre_data(uhat)
```

U *U data*

Description

A data frame containing the data to make a corrected raster by the Variational Merging method.

Usage

```
data(U,"VARMER")
```

Format

A data frame with 4320 rows and 3 columns x y and data

uhat *uhat data*

Description

A data frame containing the data present in the raster uhat.raster.

Usage

```
data(uhat,"VARMER")
```

Format

A data frame with 4340 rows and 3 columns x and y are column 1 and 2 and data is the next column.

uhat.raster *uhat.raster data*

Description

A raster layer containing temperature values of interpolated stations

Usage

```
data(uhat_raster,"VARMER")
```

Format

A raster with 70 rows and 62 columns with UTM 17S as CRS.

v *v data*

Description

A data frame containing the data present in the raster v.raster.

Usage

```
data(v,"VARMER")
```

Format

A data frame with 4340 rows and 3 columns x and y are column 1 and 2 and data is the next column.

v.raster *v.raster data*

Description

A raster layer containing temperature values of a satellite image

Usage

```
data(uhat_raster,"VARMER")
```

Format

A raster with 70 rows and 62 columns with UTM 17S as CRS.

varmer *Variational merging*

Description

Apply a function to the interpolated and satellite image (These image are raster, see raster). The function is based on many functions, these are applied in the following order:

img2varmer Compare two raster or bricks with different z values(uhat and v images) It check if the images has the same extension, resolution, coordinate reference system, columns and rows.

pre_data Using a variable in the needed format the unique coordinates. The function calculates necessary variables for mesh_3.

mesh_3 The function mesh_3 takes as input data the length of the domain in x (Lx), the length of the domain in y (Ly), the number of divisions in x (Nx) and the number of divisions in y (Ny). These

values are obtained with the function prev_data, specifically from coordinates inside the variable uhat (which must be equal to those of v).

var_merge It takes as input the coordinates of the nodes (variable coordinates defined in the mesh_3 function). The table of elements (variable elements defined in the mesh_3 function). The fields uhat and v, and the factor eta_0.

plot_mesh Using the data of previous functions (mesh_3 and var_merge), three 3D graphs are realized. Each graph represent in order the variation through the space (longitude and latitude) U,uhat and v.

Usage

```
varmer(
  uhat.img,
  v.img,
  eta_0,
  color_plot = grDevices::heat.colors,
  create_plots = F
)
```

Arguments

uhat.img	The interpolated image as a raster.
v.img	The satellite image as a raster.
eta_0	A scalar
color_plot	The name of a palette of color from grDevices
create_plots	Flag defining whether plots are to be created

Value

Three 3D plots corresponding to the variation of U, uhat and v in the space. A list containing initial data, preparation data and U. Inside of U there are two files U_varmer is a data frame and U_img a raster.

Examples

```
library(raster)
data(uhat_raster,"VARMER")
force(uhat.raster)
data(v_raster,"VARMER")
force(v.raster)
eta_0=0.5
color_plot=topo.colors
varmer_results=varmer(uhat.raster,v.raster,eta_0,color_plot)
```

varmer.ts *Variational merging for time series*

Description

It allows merging satellite-based or model-based grided images (in raster format) with ground-based observed time-series (in zoo format). This function is a wrapper for the function which allows to combine a product-based image with an interpolated image from observations for a single time step.

Usage

```
varmer.ts(
  x,
  x.metadata,
  v,
  lat = "lat",
  lon = "lon",
  drty.out = tempdir(),
  verbose = T
)
```

Arguments

x	data.frame with the ground-based observation in time series format. Every column must represent one ground-based station and the codes of the stations must be provided as colnames. class(data) must be zoo.
x.metadata	data.frame with the ground-based stations' metadata. At least, it MUST have the following 3 columns: id: This column stores the unique identifier (ID) of each ground observation. Default value is "id". lat: This column stores the latitude of each ground observation. Default value is "lat". lon: This column stores the longitude of each ground observation. Default value is "lon".
v	The satellite-based or model-based gridded images (in raster format, see raster). It can be a RasterStack or RasterBrick object since it is supposed to containe multiple layers (time series).
lat	A character string with the name of the column in x.metadata where the latitude of the stations is stored.
lon	A character string with the name of the column in x.metadata where the longitude of the stations is stored.
drty.out	A character string with the full path to the directory where the final merged products will be exported
verbose	A logical which indicates if information messages are to be printed. By default verbose=TRUE

var_merge *Solve the variational problem by the finite element method.*

Description

It takes as input the coordinates of the nodes (variable coordinates defined in the mesh_3 function). The table of elements (variable elements defined in the mesh_3 function). The fields uhat and v, and the factor eta_0

Note that columns of uhat and v are arranged as follows:

date1 date2 date3

And performs the variational merging technique to uhat and v using the finite.

uhat and v have to be introduced like in the example, only with data

Usage

```
var_merge(coordinates, elements, uhat, v, eta_0)
```

Arguments

coordinates	A data frame with Node definitions in relative coordinates.
elements	A data frame containing element definitions and nodes should be ordered counterclockwise.
uhat	A matrix containing values from the interpolated image with x y and data and the next column structure x y date1 date2 date3
v	A matrix containing values from the satellite/model based image. The values of uhat and v should correspond to the same pixels in the coordinates matrix.
eta_0	A scalar representing length scale constant.

Value

A data frame representing resulting images in relative coordinates.

Examples

```
data(uhat,"VARMER")
invisible(force(uhat))
data(v,"VARMER")
invisible(force(v))
data(coordinates,"VARMER")
invisible(force(coordinates))
data(elements,"VARMER")
invisible(force(elements))
```

```
eta_0=0.5
U=var_merge(coordinates, elements, uhat[,3:ncol(uhat)], v[,3:ncol(v)], eta_0)
```

Index