Package ‘geosptdb’

June 23, 2015

Type  Package
Title  Spatio-Temporal; Inverse Distance Weighting and Radial Basis Functions with Distance-Based Regression
Version  0.5-0
Date  2015-06-22
Author  Carlos Melo, Oscar Melo <oomelom@unal.edu.co>
Maintainer  Carlos Melo <cmelo@udistrital.edu.co>
Depends  R (>= 2.15.0), FD, StatMatch, fields, sp
Imports  minqa, limSolve, gsl, geospt
Description  Spatio-temporal: Inverse Distance Weighting (IDW) and radial basis functions; optimization, prediction, summary statistics from leave-one-out cross-validation, adjusting distance-based linear regression model and generation of the principal coordinates of a new individual from Gower’s distance.
License  GPL (>= 2)
Encoding  latin1
NeedsCompilation  no
Repository  CRAN
Date/Publication  2015-06-23 15:06:59

R topics documented:

geosptdb-package .................................................. 2
cp.xnews .......................................................... 3
criterioST.cv ...................................................... 4
croatia ............................................................ 5
croatia.grid7cp ................................................... 6
croatia.temp ....................................................... 7
croatiadb .......................................................... 8
dblm ................................................................. 9
extractFormula .................................................... 10
graph.rbfST ....................................................... 11
Spatio-Temporal: Inverse Distance Weighting and Radial Basis Functions with Distance-Based Regression

Description

Spatio-temporal: Inverse Distance Weighting (IDW) and radial basis functions; optimization, prediction, summary statistics from leave-one-out cross-validation, adjusting distance-based linear regression model and generation of the principal coordinates of a new individual from Gower’s distance.

Details

Package: geosptdb
Type: Package
Version: 0.5-0
Date: 2015-06-22
License: GPL (>= 2)
LazyLoad: yes

Author(s)

Carlos Melo <cmelo@udistrital.edu.co>, Oscar Melo <oomelom@unal.edu.co>
Maintainer: Carlos Melo <cmelo@udistrital.edu.co>

References


See Also

`rbfST, graph.rbfST, cp.xnews, croatiadb`

---

**Description**

Function for generates a numeric matrix with principal coordinates of a new individual then you could obtain distances from this matrix and you can do a prediction using a Gower’s result (1971) and Cuadras & Arenas (1990) which relates the squared distances vector with the principal coordinates vector associated to the new individual.

**Usage**

`cp.xnews(newdata, eigenvalues, data, trend, ...)`

**Arguments**

- `newdata`: data frame values of new individual.
- `eigenvalues`: the $n$ eigenvalues computed during the scaling process (see `cmdscale`).
- `data`: matrix or data frame containing the explanatory variables. These variables can be numeric, ordered, or factor, the symmetric or asymmetric binary variables should be numeric and only contain 0 and 1 character variables will be converted to factor. NAs are tolerated. With these variables the principal coordinates are built which become the regressors in the linear model.
- `trend`: matrix $n \times k$ of the $k$ most statistically significant principal coordinates (5%) with the response variable, obtained from the matrix or data frame containing explanatory variables.
- `...`: further parameters to be passed to the `gower.dist` function (see `gower.dist`).
Value

Returns a numeric matrix with principal coordinates of the new individual.

References


See Also
dblm, rbfST

Examples

```r
## Not run:
data(croatia.temp)
data(croatiadb)
# prediction case: one point
point <- data.frame(670863,5043464,5,170,200,15.7,3)
names(point) <- c("x", "y", "t", "dem", "dsea", "twi", "est")
croatia.temp[,7] <- as.factor(croatia.temp[,7])
dblm1 <- dblm(data=croatia.temp, y=croatiadb$TEMP)
newdata1 <- t(cp.xnews(newdata=point, eigenvalues=dblm1$ev, data=croatia.temp,
trend=dblm1$cp))
colnames(newdata1) <- c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10")

## End(Not run)
```

<table>
<thead>
<tr>
<th>criterioST.cv</th>
<th>Cross-validation summaries</th>
</tr>
</thead>
</table>

Description

Generate a data frame of statistical values associated with cross-validation

Usage

criterioST.cv(m.cv)
Arguments

\texttt{m.cv} data frame containing: prediction columns, prediction variance of cross-validation data points, observed values, residuals, zscore (residual divided by kriging standard error), and fold. If the \texttt{rbfST.tcv} function is used, the prediction variance, zscore (residual divided by standard error) will have NA's, coordinates data and time.

Value

data frame containing: mean prediction errors (MPE), average kriging standard error (ASEPE), root-mean-square prediction errors (RMSPE), mean standardized prediction errors (MSPE), root-mean-square standardized prediction errors (RMSSPE), mean absolute percentage prediction errors (MAPPE), coefficient of correlation of the prediction errors (CCPE), coefficient of determination (R2) and squared coefficient of correlation of correlation of the prediction errors (pseudoR2)

Examples

\begin{verbatim}
# leave-one-out cross validation:
data(croatia)
coordinates(croatia) <- ~x+y

criterioST.cv(tempm)
\end{verbatim}

\begin{verbatim}
croatia Map Croatia.
\end{verbatim}

Description

Map Croatia. Spatial reference system: UTM 33N

Usage

data(croatia)

Format

The format is: Formal class 'SpatialPolygonsDataFrame' [package "sp"]

References

Examples

```r
data(croatia)
pts <- spsample(croatia, n=25000, type="regular")
plot(pts)
```

**croatia.grid7cp**  
Principal coordinates of a pixelated size 4994 in Croatia.

Description

data frame 4994 × 13 of spatio-temporal coordinates and principal coordinates associated with a pixelated size 4994 in Croatia. Spatial reference system: UTM 33N.

Usage

```r
data(croatia.grid7cp)
```

References


See Also

```
croatia.temp
```

Examples

```r
data(croatia.grid7cp)
plot(croatia.grid7cp[,1:2])
```
Data climatic stations in Croatia.

Description

Information of 142 climatic stations in Croatia in 2008, with topographical static predictors (Digital Elevation Model, (DEM, in meters), topographically weighted distances from the coast line (DSEA, in km), topographic wetness index (TWI))

Usage

data(croatia.temp)

Format

A data frame with 1752 observations on the following 7 variables:

x  a numeric vector; x-coordinate; Spatial reference system: UTM 33N
y  a numeric vector; y-coordinate; Spatial reference system: UTM 33N
t  a numeric vector; t-coordinate (1-12 for the months from January to December)
dem a numeric vector, Digital Elevation Model (DEM, in meters)
dsea a numeric vector with topographically weighted distances from the coast line (DSEA, in km)
twi a numeric vector with topographic wetness index
est a numeric vector with seasons (1 for January, February and March, 2 for April, May and June, 3 for July, August and September and 4 for October, November and December)

References


Examples

data(croatia.temp)
summary(croatia.temp)
data frame 1752 × 14 of spatio-temporal coordinates, earth’s average temperature monthly and 10 principal coordinates associated with data climatic stations in Croatia 2008.

Usage

data(croatiadb)

Format

The format is: Formal class 'data.frame' [package "base"]

References


See Also

croatia.grid7cp, croatia.temp

Examples

data(croatiadb)
str(croatiadb)
names(croatiadb)
Description

dblm is a linear model variety where explanatory information is coded as distances among individuals so these distances can also be computed from observed explanatory variables (a mix of continuous, qualitative explanatory variables or from more general quantities). The response is a continuous variable as in the classic linear model.

lm is used internally to adjust a distance-based linear regression model. The method considers the Gower's distance for mixed covariates (numeric, ordered, or factor), for explanation on the meaning of distance-based linear regression model and distance of Gower see the bibliography references below.

Usage

dblm(data, y, sc, ev.min, ...)

Arguments

data matrix or data frame containing the explanatory variables. These variables can be numeric, ordered, or factor. Symmetric or asymmetric binary variables should be numeric and only contain 0 and 1. character variables will be converted to factor. NAs are tolerated. With these variables are built, the principal coordinates which later become the regressors in the linear model.

y the response variable used to fit the model

sc the value of the correlation squared to select the principal coordinates more related to the response variable. The default value is 0.003.

ev.min the minimum value to select the eigenvalues. These eigenvalues must be positive, the default value is 0.007

... further parameters to be passed to the gowdis function (see gowdis) of low level.

Details

The dblm model builds; principal coordinates matrix, eigenvalues, and a linear regression model. gowdis function used in dblm compute the Gower (1971) similarity coefficient exactly as described by Podani (1999), then converts it to a dissimilarity coefficient by using $D = 1 - S$. It integrates variable weights as described by Legendre and Legendre (1998).

Value

A list containing the following components:

table table with eigenvalues, correlations squared, and percentages of inertia associated with the most statistically significant principal coordinates (5%) with the response variable.
the $n$ eigenvalues computed during the scaling process (see `cmdscale`).

- **cp**: the $k$ most statistically significant principal coordinates (5%) with the response variable.

- **dbmodel**: returns a list of summary statistics of the fitted linear model.

**References**


**See Also**

See function `gowdis` in the FD package.

**Examples**

```r
# considering 10 principal coordinates (constructed from a distance-based linear
# regression model)
## Not run:
data(croatia.temp)
data(croatiaadb)
croatia.temp[,,7] <- as.factor(croatia.temp[,,7])
dblm1 <- dblm(data=croatia.temp,y=croatiaadb$MTEMP)
str(dblm1)

## End(Not run)
```

**Description**

geospt internal function

**Note**

This function is not meant to be called by users directly.
graph.rbfST

Graph that describes the behavior of the optimized \( \eta \) and \( \rho \) parameters, associated with a spatio-temporal radial basis function.

Description

Function for plotting the RMSPE for several values of the smoothing parameter \( \eta \) with the same dataset. A curve is fitted to the points, and then the optimal \( \eta \) that provides the smallest RMSPE is determined from the curve, by the optimize function from the stats package.

Usage

\[
\text{graph.rbfST}(\text{formula}, \text{data}, \text{eta.opt}, \text{rho.opt}, \text{n.neigh}, \text{func}, \text{np}, \text{xo}, \text{eta.dmax}, \text{rho.dmax}, \text{P.T}, \text{iter}, \ldots)
\]

Arguments

- **formula**: formula that defines the dependent variable as a linear model of independent variables (covariates or the principal coordinates); suppose the dependent variable has name \( z_{st} \), for a rbfST detrended use \( z_{st} \sim 1 \), for a rbfST with trend, suppose \( z_{st} \) is linearly dependent on \( x \) and \( y \), use the formula \( z_{st} \sim x+y \) (linear trend).
- **data**: SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.
- **eta.opt**: logical, indicating whether the parameter \( \eta \) should be regarded as fixed (\( \eta_{\text{opt}} = \text{FALSE} \)) or should be estimated (\( \eta_{\text{opt}} = \text{TRUE} \)).
- **rho.opt**: logical, indicating whether the parameter \( \rho \) should be regarded as fixed (\( \rho_{\text{opt}} = \text{FALSE} \)) or should be estimated (\( \rho_{\text{opt}} = \text{TRUE} \)).
- **n.neigh**: number of nearest observations that should be used for a rbfST prediction, where nearest is defined in terms of the spatio-temporal locations.
- **func**: function to be optimized. The following radial basis function spatio-temporal model types are currently available: gaussian "GAU", exponential "EXPON", trigonometric "TRI", thin plate spline "TPS", completely regularized spline "CRS", spline with tension "ST", inverse multiquadratic "IM", and multiquadratic "M", are currently available.
- **np**: number of points, where the radial basis function spatio-temporal is calculated.
- **xo**: starting point for searching the optimum. Defaults to c(0.5, 0.5), \( \eta \) and \( \rho \) respectively. Use this statement only if \( \eta \) and \( \rho \) are equal to TRUE.
- **eta.dmax**: maximum value of the range of the \( \eta \) parameter that will be evaluated by the optimize function.
- **rho.dmax**: maximum value of the range of the \( \rho \) parameter that will be evaluated by the optimize function.
- **P.T**: logical. Print table (TRUE) or not (FALSE). Default P.T=NULL.
The maximum allowed number of function evaluations.

further parameters to be passed to the minimization functions optimize or bobyqa, typically arguments of the type control() which control the behavior of the minimization algorithm. See documentation about the selected minimization function for further details.

Returns a graph that describes the behavior of the optimized eta or rho parameters and a table of values associated with the graph including optimal smoothing eta or rho parameters. If both eta and rho are FALSE simultaneously then the function returns a list with the best value obtained from the combinations smoothing eta and rho parameters and a lattice plot of class "trellis" with RMSPE pixel values associated with combinations of eta and rho parameters. Finally, if both eta and rho are TRUE, the function will return a list with the best combination of values of the smoothing eta or rho parameters and the RMSPE associated with these.


See Also
rbfST, rbfST.cv

Examples

```r
# Not run:
data(croatiadb)
coordinates(croatiadb)<-x+y
# optimizing eta
graph.rbfST(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=croatiadb, eta.opt=TRUE, rho.opt=FALSE, n.neigh=30, func="TPS", np=40, eta.dmax=2, P.T=TRUE)
# optimizing rho
graph.rbfST(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=croatiadb, eta.opt=FALSE, rho.opt=TRUE, n.neigh=30, func="M", np=20, rho.dmax=2, P.T=TRUE)
# optimizing eta and rho
tps.lo <- graph.rbfST(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=croatiadb, eta.opt=TRUE, rho.opt=TRUE, n.neigh=25, func="TPS", eta.dmax=0.2, rho.dmax=0.2, xo=c(0.1,0.1), iter=50)
tps.lo # best combination of eta and rho obtained
# lattice of RMSPE values associated with a range of eta and rho, without optimization
tps.la <- graph.rbfST(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=croatiadb, eta.opt=FALSE, rho.opt=FALSE, n.neigh=30, func="TPS", np=10, eta.dmax=0.2, rho.dmax=0.2)
tps.l$table # best combination of eta and rho obtained
tps.l$splot # lattice of RMSPE
```

## End(Not run)
**Description**

This function performs spatio-temporal interpolation. Here `$idwST$` is in a local neighborhood. This interpolation method considers the value of a point can be obtained from the weighted sum of values of the regionalized variable of closest neighbors. The general formula for the IDW is given by:

$$
\hat{z}_0(st) = \sum_{i=1}^{n} \lambda_i z_i(st)
$$

The expression for determining the weights is:

$$
\lambda_i = \frac{d_{i0}^{-p}}{\sum_{i=1}^{n} d_{i0}^{-p}}
$$

The weight is controlled by a factor `$p$` with each increment of the distance, `$d_{i0}$` is the distance between the prediction position and each of the measured positions.

The expression `$d_{i0}$` can be obtained by:

$$
d_{i0} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + C \cdot (t_i - t_0)^2}
$$

$x$, $y$ and $t$ correspond to the spatio-temporal coordinates, `$p$` (factor.p) and $C$ factors defined below.

**Usage**

```r
idwST(formula, data, newdata, n.neigh, C, factor.p, progress)
```

**Arguments**

- `formula` : formula that defines a detrended linear model, use `$z_{st} \sim 1$`.
- `data` : `SpatialPointsDataFrame`: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.
- `newdata` : data frame or spatial object with prediction/simulation spatio-temporal locations; should contain attribute columns with the independent variables (if present) and (if locations is a formula) the coordinates and time with names, as defined in locations where you want to generate new predictions
- `n.neigh` : number of nearest observations that should be used for a `$idwST$` prediction, where nearest is defined in terms of the spatio-temporal locations
- `C` : numeric; associated to time factor, we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation. Using `$idwST.cv$` and `optimize`
- `factor.p` : numeric; specify the inverse distance weighting power ($p$ is the exponent that influences the weighting or optimal smoothing parameter)
- `progress` : whether a progress bar shall be printed for spatio-temporal inverse-distance weighted function; default=TRUE
Details

idwST function generates individual spatio-temporal predictions from IDW spatio-temporal interpolation. IDW is a type of deterministic method for interpolation, the assigned values to unknown points are calculated with a weighted average of the values available at the known points.

Value

Attributes columns contain coordinates, time, predictions, and the variance column contains NA's

References


Examples

```r
# Loading Croatia data
data(croatiadb)
coordinates(croatiadb) <- ~x+y

# prediction case: one point
point <- data.frame(670863, 5043464, 5)
names(point) <- c("x","y","t")
coordinates(point) <- ~x+y
idwST(MTEMP~1, data=croatiadb, newdata=point, n.neigh=60, C=1, factor.p=2)

## Not run:
# prediction case: a grid of points Croatia (year 2008)
data(croatia)
points <- spsample(croatia, n=5000, type="regular")
data(croatiadb)
coordinates(croatiadb) <- ~x+y
GridsT <- vector(mode = "list", length = 12)

for(i in 1:12){
 GridsT[[i]] <- data.frame(points@coords,i)
names(GridsT[[i]]) <- c("x","y","t")
}

idw.croatia <- data.frame(matrix(NA, ncol = 14, nrow=nrow(GridsT[[1]])))
pb <- txtProgressBar(min = 0, max = 12, char = "=", style = 3)
for(i in 1:12){
  coordinates(GridsT[[i]]) <- c("x", "y")
idw.croatia[,i+2] <- idwST(MTEMP~1, croatiadb, newdata=GridsT[[i]], n.neigh=10, C=1, factor.p=2, progress=FALSE)[,4]
  setTxtProgressBar(pb, i)
```
idwST.cv

IDW spatio-temporal leave-one-out cross validation

Description

Generate the RMSPE value which is given by Inverse Distance Weighting (IDW) interpolation.

Usage

idwST.cv(formula, data, n.neigh, C, factor.p, progress)

Arguments

formula formula that defines a detrended linear model, use \( z_{st} \sim 1 \).

data SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.
idwST.cv1

n.neigh  
number of nearest observations that should be used for a rbf.st prediction, where nearest is defined in terms of the spatio-temporal locations

C  
umeric; associated to time factor, we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation. Using idwST.cv and optimize

factor.p  
umeric; specify the inverse distance weighting power (p is the exponent that influences the weighting or optimal smoothing parameter)

progress  
whether a progress bar shall be printed for spatio-temporal inverse-distance weighted function; default=TRUE

Value

returns the RMSPE value

References


See Also

idwST.rbfST

Examples

```r
## Not run:
data(croatiadb)
coordinates(croatiadb) <- ~x+y
idwST.cv(MTEMP~1, croatiadb[,1:2], n.neigh=10, C=1, factor.p=2)
## End(Not run)
```

idwST.cv1  
Generate a RMSPE value, result of leave-one-out cross validation

Description

Generate the RMSPE value which is given by the radial basis function spatio-temporal with number of nearest observations n.neigh associated to time factor C and optimal smoothing parameter factor.p.

Usage

```r
idwST.cv1(param, formula, data, n.neigh, progress)
```
Arguments

param  vector starting points \( (C \text{ and } factor.p \text{ respectively}) \) for searching the RMSPE optimum.

formula  formula that defines a detrended linear model, use \( z_{st} \sim 1 \).

data  SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.

n.neigh  number of nearest observations that should be used for a \( rbf.st \) prediction where nearest is defined in terms of the spatio-temporal locations

progress  whether a progress bar shall be printed for spatio-temporal inverse-distance weighted function; default=FALSE

Value

returns the RMSPE value

See Also

\( idwST, idwST.cv, idwST.tcv \)

Examples

```r
require(minqa)
data(croatia$db)
coordinates(croatia$db) <- ~x+y

## Not run:
idwST.opt <- bobyqa(c(1, 2), idwST.cv1, lower=c(0, 0.1), upper=c(2, 4), formula=MTEMP~1, 
data=croatia$db[,1:2], n.neigh=10, progress=F, control=list(maxfun=50))

# obtained with optimal values previously estimated (33 iterations)
idwST.cv1(c(1.09538675066736, 1.95853920335545), MTEMP~1, data=croatia$db[,1:2], n.neigh=10, 
progress=T)

## End(Not run)
```

idwST.tcv  \( \text{table of } idw \text{ spatio-temporal leave-one-out cross validation} \)

Description

Generates a table with the results of inverse distance weighting spatio-temporal interpolation (\( idwST \)) from leave-one-out cross validation method.

Usage

\( idwST.tcv(formula, data, n.neigh, C, factor.p, progress) \)
Arguments

- **formula**: formula that defines a detrended linear model, use $z_{st} \sim 1$.
- **data**: SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.
- **n.neigh**: number of nearest observations that should be used for an \texttt{idwST} prediction where nearest is defined in terms of the spatio-temporal locations.
- **C**: numeric; associated to time factor, we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation. Using \texttt{idwST.cv} and \texttt{optimize}.
- **factor.p**: numeric; specify the inverse distance weighting power ($p$ is the exponent that influences the weighting or optimal smoothing parameter).
- **progress**: whether a progress bar shall be printed for spatio-temporal inverse-distance weighted function; default=TRUE

Details

Leave-one-out cross validation (LOOCV) consists of removing data, one at a time, and then trying to predict it. Next, the predicted value can be compared to the actual (observed) value to assess how well the prediction is working. The observed value is left out because \texttt{idwST} would otherwise predict the value itself.

Value

data frame contain prediction columns, observed values, residuals, the prediction variance, $z$score (residual divided by standard error) which left with NA's, the fold column which is associated to cross-validation count, coordinates data and time. Prediction columns and residuals are obtained from cross-validation data points.

See Also

- \texttt{idwST}

Examples

```r
## Not run:
data(croatia.db)
coordinates(croatia.db) <- ~x+y
idw.t <- \texttt{idwST.tcv(MTEMP=1, croatia.db, n.neigh=10, C=1.0054, factor.p=1.9585)}
criterioST.cv(idw.t)
## End(Not run)
```
rbfST

gaussian, exponential, trigonometric, thin plate spline, inverse multiquadratic, and multiquadratic radial basis function for spatio-temporal prediction

Description

Function for spatio-temporal interpolation from radial basis function (rbfST), where rbfST is in a local neighbourhood.

exponential (EXPON)

\[
\phi(\delta) = e^{-\eta \delta}, \eta > 0
\]

gaussian (GAU)

\[
\phi(\delta) = e^{-\eta \delta^2}, \eta \neq 0
\]

multiquadratic (M)

\[
\phi(\delta) = \sqrt{\eta^2 + \delta^2}, \eta \neq 0
\]

inverse multiquadratic (IM)

\[
\phi(\delta) = \frac{1}{\sqrt{\eta^2 + \delta^2}}, \eta \neq 0
\]

thin plate spline (TPS)

\[
\phi(\delta) = (\eta \cdot \delta)^2 \log(\eta \cdot \delta), \text{if } \delta > 0, \eta > 0 \\
\phi(\delta) = 0, \text{otherwise}
\]

completely regularized spline (CRS)

\[
\phi(\delta) = \ln(\eta \cdot \delta/2)^2 + E_1(\eta \cdot \delta/2)^2 + C_E, \text{if } \delta > 0, \eta > 0 \\
\phi(\delta) = 0, \text{otherwise}
\]

where \(\ln\) is natural logarithm, \(E_1(x)\) is the exponential integral function, and \(C_E\) is the Euler constant.

spline with tension (ST)

\[
\phi(\delta) = \ln(\eta \cdot \delta/2) + K_0(\eta \cdot \delta) + C_E, \text{if } \delta > 0 \\
\phi(\delta) = 0, \text{otherwise}
\]

where \(K_0(x)\) is the modified Bessel function and \(C_E\) is the Euler constant.

Usage

rbfST(formula, data, eta, rho, newdata, n.neigh, func, progress)
Arguments

formula formula that defines the dependent variable as a linear model of independent variables (covariates or principal coordinates); suppose the dependent variable has name \( z_{st} \) for a \( rbfST \) detrended use \( z_{st} - 1 \); for a \( rbfST \) with trend suppose \( z_{st} \) is linearly dependent on \( x \) and \( y \), use the formula \( z_{st} - x + y \) (linear trend).

data SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.

eta the optimal smoothing parameter, we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation

rho optimal robustness parameter, we recommend using the value obtained by minimizing the root-mean-square prediction errors with cross-validation. \( \eta \) and \( \rho \) parameters can be optimized simultaneously, through the \texttt{bobyqa} function from \texttt{nloptr} or \texttt{minqa} packages

newdata data frame or spatial object with prediction/simulation spatio-temporal locations; should contain attribute columns with the independent variables (if present) and (if locations is a formula) the coordinates and time with names, as defined in locations where you want to generate new predictions

n.neigh number of nearest observations that should be used for a \( rbfST \) prediction, where nearest is defined in terms of the spatio-temporal locations

func spatio-temporal radial basis function; model type: “GAU”, “EXPON”, “TRI”, “TPS”, “CRS”, “ST”, “IM” and “M”, are currently available

progress whether a progress bar shall be printed for spatio-temporal radial basis functions; default=TRUE

Details

\( rbf.st \) function generates individual spatio-temporal predictions from gaussian (GAU), exponential (EXPON), trigonometric (TRI) thin plate spline (TPS), completely regularized spline (CRS), spline with tension (ST), inverse multiquadratic (IM), and multiquadratic (M) functions

Value

Attributes columns contain coordinates, time, predictions, and the variance column contains NA’s

References


Examples

```r
## Not run:
# considering 10 principal coordinates (constructed from a distance-based regression model)
data(croatia.temp)
data(croatia.db)
```
# prediction case: one point
point <- data.frame(670863, 5043464, 5, 170, 200, 15, 7, 3)
names(point) <- c("x", "y", "t", "dem", "dsea", "twi", "est")
croatia.temp[,7] <- as.factor(croatia.temp[,7])
dblm1 <- dblm(data=croatia.temp, y=croatia$MTEMP)
newdata1 <- t(cp.xnew(newdata=point, eigenvalues=dblm1$ev, data=croatia.temp, trend=dblm1$cp))
colnames(newdata1) <- c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10")
newdata1 <- data.frame(point[,1:3], newdata1)
data(croatiadb)
coordinates(croatiadb) <- ~x+y
coordinates(newdata1) <- ~x+y
rbfST(MTEMP~x1+xR+xS+xT+xU+xV+xW+xX+xY+x1PL data=croatiadb, eta=0.01076, rho=0.00004, newdata=newdata1, n.neigh=60, func="TPS")

data(croatia.gridWcp)
coordinates(croatia.gridWcp) <- ~x+y
rbf.t <- rbfST(MTEMP~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10, data=croatiadb, eta=0.01076, rho=0.00004, newdata=croatia.grid7cp, n.neigh=30, func="TPS")
coordinates(rbf.t) <- c("x", "y")
gridded(rbf.t) <- TRUE

# show prediction map
spplot(rbf.t["var1.pred"], cuts=30, col.regions=bpy.colors(40), main = "Earth's average temperature TPS map\n(july month)", key.space=list(space="right", cex=0.8))

## End(Not run)

---

**rbfST.cv**  
**Leave-one-out cross validation for spatio-temporal radial basis function**

---

**Description**

It generates the RMSPE value, which is given by the radial basis function with smoothing eta and robustness rho parameters.

**Usage**

`rbfST.cv(formula, data, eta, rho, n.neigh, func)`

**Arguments**

- `formula` : formula that defines the dependent variable as a linear model of independent variables (covariates or the principal coordinates); suppose the dependent variable has name `z_{st}`, for a `rbfST` detrended use `z_{st}^{-1}`, for a `rbfST` with trend, suppose `z_{st}` is linearly dependent on `x` and `y`, use the formula `z_{st} \sim x+y` (linear trend).
data SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.

eta the optimal smoothing parameter, we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation.

rho optimal robustness parameter, we recommend using the value obtained by minimizing the root-mean-square prediction errors with cross-validation. \( \eta \) and \( \rho \) parameters can be optimized simultaneously, through the \texttt{bobyqa} function from \texttt{nloptr} or \texttt{minqa} packages.

n.neigh number of nearest observations that should be used for a \texttt{rbfST} prediction, where nearest is defined in terms of the spatio-temporal locations.

func spatio-temporal radial basis function; model type: "GAU", "EXPON", "TRI", "TPS", "CRS", "ST", "IM" and "M", are currently available

Value

returns the RMSPE value

References


See Also

\texttt{rbfST}, \texttt{graph.rbfST}

Examples

data(croatiadb)
coordinates(croatiadb) \leftarrow ~x+y
rbfST.cv1(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, croatiadb, eta=0.0108, rho=0.00004, n.neigh=25, func="TPS")

\texttt{rbfST.cv1} \hspace{1cm} \textit{RMSPE value result of leave-one-out cross validation for rbfST}

Description

It generates the RMSPE value which is given by the spatio-temporal radial basis function with smoothing \( \eta \) and robustness \( \rho \) parameters.

Usage

\texttt{rbfST.cv1(param, formula, data, n.neigh, func)}
Arguments

**param**  
vector starting points (eta and rho respectively) for searching the RMSPE optimum.

**formula**  
formula that defines the dependent variable as a linear model of independent variables (covariates or the principal coordinates); suppose the dependent variable has name \( z_{st} \), for a \( rbfST \) detrended use \( z_{st} \sim 1 \), for a \( rbfST \) with trend, suppose \( z_{st} \) is linearly dependent on \( x \) and \( y \), use the formula \( z_{st} \sim x+y \) (linear trend).

**data**  
SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.

**n.neigh**  
number of nearest observations that should be used for a \( rbfST \) prediction, where nearest is defined in terms of the spatio-temporal locations.

**func**  
spatio-temporal radial basis function; model type: "GAU", "EXPON", "TRI", "TPS", "CRS", "ST", "IM" and "M", are currently available

**Value**

returns the RMSPE value

**See Also**

\( rbfST, rbfST.cv, graph.rbfST \)

**Examples**

```r
require(minqa)
data(croatiadb)
coordinates(croatiadb) <- ~x+y

## Not run:
rbf.im <- bobyqa(c(0.5, 0.5), rbfST.cv1, lower=c(1e-05, 0), upper=c(2,2),
                  formula=MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=croatiadb, n.neigh=25,
                  func="IM", control=list(maxfun=50))

## End(Not run)

# obtained with the optimal values previously estimated
rbfST.cv1(c(0.847050095690357, 0.104157855356128), MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10,
          croatiadb, n.neigh=25, func="IM")
```

---

\( rbfST.tcv \)  
*table of rbf spatio-temporal cross validation, leave-one-out*
Description

It generates a table with the results of the evaluation of radial basis functions spatio-temporal (rbfST): gaussian (GAU), exponential (EXPON), trigonometric (TRI), thin plate spline (TPS), completely regularized spline (CRS), spline with tension (ST), inverse multiquadratic (IM), and multiquadratic (M) from the leave-one-out cross validation method.

Usage

rbfST.tcv(formula, data, eta, rho, n.neigh, func, progress)

Arguments

formula formula that defines the dependent variable as a linear model of independent variables (covariates or the principal coordinates); suppose the dependent variable has name z_st, for a rbf.st detrended use z_st~1, for a rbf.st with trend, suppose z_st is linearly dependent on x and y, use the formula z_st~x*y (linear trend).

data SpatialPointsDataFrame: should contain the spatio-temporal dependent variable, independent variables (statics and/or dynamics), spatial coordinates and the time as an integer or numerical variable.

eta the optimal smoothing parameter; we recommend using the parameter found by minimizing the root-mean-square prediction errors using cross-validation

rho optimal robustness parameter, we recommend using the value obtained by minimizing the root-mean-square prediction errors with cross-validation. eta and rho parameters can be optimized simultaneously, through the bobyqa function from nloptr or minqa packages

n.neigh number of nearest observations that should be used for a rbfST prediction, where nearest is defined in terms of the spatio-temporal locations.

func spatio-temporal radial basis function; model type: "GAU", "EXPON", "TRI", "TPS", "CRS", "ST", "IM" and "M", are currently available

progress whether a progress bar shall be printed for spatio-temporal radial basis functions; default=TRUE

Details

Leave-one-out cross validation (LOOCV) visits a data point, predicts the value at that location by leaving out the observed value, and proceeds with the next data point. The observed value is left out because rbf.st would otherwise predict the value itself.

Value

data frame contain prediction columns, observed values, residuals, the prediction variance, zscore (residual divided by standard error) which left with NA’s, the fold column which is associated to cross-validation count, coordinates data and time. Prediction columns and residuals are obtained from cross-validation data points.

See Also

rbfST
Examples

```r
data(croatiadb)
coordinates(croatiadb) <- ~x+y
rbfST.tcv(MTEMP~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, croatiadb, eta=0.0108, rho=0.00004,
n.neigh=30, func="TPS")
```

Description

standardize internal function

Note

This function is not meant to be called by users directly
Index

*Topic datasets
croatia, 5
croatia.grid7cp, 6
croatia.temp, 7
croatiadb, 8

*Topic package
geosptdb-package, 2

*Topic principal coordinates
dblm, 9

*Topic spatial
cp.xnews, 3
criterioST.cv, 4
extractFormula, 10
graph.rbfST, 11
idwST, 13
idwST.cv, 15
idwST.cv1, 16
idwST.tcv, 17
rbfST, 19
rbfST.cv, 21
rbfST.cv1, 22
rbfST.tcv, 23
standardize, 25

*Topic spatio-temporal
geosptdb-package, 2

bobyqa, 12, 20, 22, 24
cmdscale, 3, 10
cp.xnews, 3, 3
criterioST.cv, 4
croatia, 5
croatia.grid7cp, 6, 8
croatia.temp, 6, 7, 8
croatiadb, 3, 8
dblm, 4, 9, 9
extractFormula, 10
geosptdb (geosptdb-package), 2
gowdis, 9, 10
gower.dist, 3
graph.rbfST, 3, 11, 22, 23
idwST, 13, 16–18
idwST.cv, 15, 17
idwST.cv1, 16
idwST.tcv, 17, 17
lm, 9
optimize, 11–13, 16, 18
rbfST, 3, 4, 12, 16, 19, 22–24
rbfST.cv, 12, 21, 23
rbfST.cv1, 22
rbfST.tcv, 5, 23
standardize, 25