Package 'SpatialML'

January 20, 2025

Version 0.1.7

Date 2024-04-02

Type Package

Title Spatial Machine Learning

Depends R (>= 4.3.0), ranger (>= 0.15.1), caret (>= 6.0), randomForest (>= 4.7)

Description Implements a spatial extension of the random forest algorithm (Georganos et al. (2019) <doi:10.1080/10106049.2019.1595177>). Allows for a geographically weighted random forest regression including a function to find the optical bandwidth. (Georganos and Kalogirou (2022) <https://www.mdpi.com/2220-9964/11/9/471>).

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://stamatisgeoai.eu/

NeedsCompilation no

Author Stamatis Kalogirou [aut, cre], Stefanos Georganos [aut, ctb]

Maintainer Stamatis Kalogirou <stamatis.science@gmail.com>

Repository CRAN

Date/Publication 2024-04-02 00:00:02 UTC

Contents

grf *Geographically Weighted Random Forest Model*

Description

Fit a local version of the Random Forest algorithm, accounting for spatial non-stationarity.

Usage

```
grf(formula, dframe, bw, kernel, coords, ntree=500, mtry=NULL,
           importance="impurity", nthreads = NULL, forests = TRUE,
           geo.weighted = TRUE, print.results=TRUE, ...)
```

Arguments

formula	a formula specifying the local model to be fitted, using the syntax of the ranger package's ranger function.
dframe	a numeric data frame with at least two suitable variables (one dependent and one independent).
bw	a positive number representing either the number of nearest neighbors (for "adaptive kernel") or bandwidth in meters (for "fixed kernel").
kernel	the type of kernel to use in the regression: "adaptive" or "fixed".
coords	a numeric matrix or data frame containing X and Y coordinates of observations.
ntree	an integer referring to the number of trees to grow for each local random forest.
mtry	the number of variables randomly sampled as candidates at each split. Default is p/3, where p is the number of variables in the formula.
importance	feature importance measure for the dependent variables used as input in the random forest. Default is "impurity", which refers to the Gini index for classification and the variance of the responses for regression.
nthreads	number of threads for parallel processing. Default is the number of available CPUs. The argument passes to both ranger and predict functions.
forests	a option to save and export (TRUE) or not (FALSE) all local forests.
geo.weighted	if TRUE, calculate Geographically Weighted Random Forest using case weights. If FALSE, calculate local random forests without weighting each observation.
print.results	a option to print the summary of the analysis (TRUE) or not (FALSE).
...	additional arguments passed to the ranger function.

Details

Geographically Weighted Random Forest (GRF) is a spatial analysis method using a local version of the famous Machine Learning algorithm. It allows for the investigation of the existence of spatial non-stationarity, in the relationship between a dependent and a set of independent variables. The latter is possible by fitting a sub-model for each observation in space, taking into account the neighbouring observations. This technique adopts the idea of the Geographically Weighted Regression,

Kalogirou (2003). The main difference between a tradition (linear) GWR and GRF is that we can model non-stationarity coupled with a flexible non-linear model which is very hard to overfit due to its bootstrapping nature, thus relaxing the assumptions of traditional Gaussian statistics. Essentially, it was designed to be a bridge between machine learning and geographical models, combining inferential and explanatory power. Additionally, it is suited for datasets with numerous predictors, due to the robust nature of the random forest algorithm in high dimensionality.

Geographically Weighted Random Forest (GRF) is a spatial analysis method that fits a local version of the Random Forest algorithm for investigating spatial non-stationarity, in the relationship between a dependent and a set of independent variables. The latter is possible by fitting a sub-model for each observation in space, taking into account the neighbouring observations. This technique adopts the idea of the Geographically Weighted Regression, Kalogirou (2003). It models non-stationarity with a flexible non-linear approach, bridging the gap between machine learning and geographical models. The main difference between a tradition (linear) GWR and GRF is that we can model non-stationarity coupled with a flexible non-linear model which is very hard to overfit due to its bootstrapping nature, thus relaxing the assumptions of traditional Gaussian statistics.GRF is suitable for datasets with numerous predictors due to the robustness of the random forest algorithm in high dimensionality.

Value

`Global.Model`	A ranger object of the global random forest model.
`Locations`	a numeric matrix or data frame with X and Y coordinates of observations.
`Local.Variable.Importance`	anumeric data frame with local feature importance for each predictor in each local random forest model.
`LGofFit`	a numeric data frame with residuals and local goodness of fit statistics.
`Forests`	all local forests.
`lModelSummary`	Local Model Summary and goodness of fit statistics.

Warning

Large datasets may take long to calibrate. A high number of observations may result in a voluminous forests output.

Note

This function is under development, and improvements are expected in future versions of the package `SpatialML`. Any suggestions are welcome!

Author(s)

Stamatis Kalogirou <stamatis.science@gmail.com>, Stefanos Georganos <sgeorgan@ulb.ac.be>

References

Stefanos Georganos, Tais Grippa, Assane Niang Gadiaga, Catherine Linard, Moritz Lennert, Sabine Vanhuysse, Nicholus Odhiambo Mboga, Eléonore Wolff & Stamatis Kalogirou (2019) Geographical Random Forests: A Spatial Extension of the Random Forest Algorithm to Address Spatial Heterogeneity in Remote Sensing and Population Modelling, Geocarto International, DOI: 10.1080/10106049.2019.1595177

Georganos, S. and Kalogirou, S. (2022) A Forest of Forests: A Spatially Weighted and Computationally Efficient Formulation of Geographical Random Forests. ISPRS, International Journal of Geo-Information, 2022, 11, 471. <https://www.mdpi.com/2220-9964/11/9/471>

See Also

predict.grf

Examples

```
  ## Not run:
      RDF <- random.test.data(10,10,3)
      Coords<-RDF[ ,4:5]
      grf <- grf(dep ~ X1 + X2, dframe=RDF, bw=10,
                 kernel="adaptive", coords=Coords)

## End(Not run)

      data(Income)
      Coords<-Income[ ,1:2]
      grf <- grf(Income01 ~ UnemrT01 + PrSect01, dframe=Income, bw=60,
                 kernel="adaptive", coords=Coords)
```

---

grf.bw *Geographically Weighted Random Forest optimal bandwidth selection*

Description

This function finds the optimal bandwidth for the Geographically Weighted Random Forest algorithm using an exhaustive approach.

Usage

```
grf.bw(formula, dataset, kernel="adaptive", coords, bw.min = NULL,
            bw.max = NULL, step = 1, trees=500, mtry=NULL, importance="impurity",
              nthreads = 1, forests = FALSE, geo.weighted = TRUE, ...)
```

Arguments

formula	the local model to be fitted using the same syntax used in the ranger function of the R package ranger. This is a string that is passed to the sub-models' ranger function. For more details look at the class formula.
dataset	a numeric data frame of at least two suitable variables (one dependent and one independent)
kernel	the kernel to be used in the regression. Options are "adaptive" (default) or "fixed".

coords	a numeric matrix or data frame of two columns giving the X,Y coordinates of the observations
bw.min	an integer referring to the minimum bandwidth that evaluation starts.
bw.max	an integer referring to the maximum bandwidth that evaluation ends.
step	an integer referring to the step for each iteration of the evaluation between the min and the max bandwidth. Default value is 1.
trees	an integer referring to the number of trees to grow for each of the local random forests.
mtry	the number of variables randomly sampled as candidates at each split. Note that the default values is p/3, where p is number of variables in the formula
importance	feature importance of the dependent variables used as input at the random forest. Default value is "impurity" which refers to the Gini index for classification and the variance of the responses for regression.
nthreads	Number of threads. Default is number of CPUs available. The argument passes to both ranger and predict functions.
forests	a option to save and export (TRUE) or not (FALSE) all the local forests. Default value is FALSE.
geo.weighted	if TRUE the algorithm calculates Geographically Weighted Random Forest using the case.weights option of the package ranger. If FALSE it will calculate local random forests without weighting each observation in the local data set.
...	further arguments passed to the grf and ranger functions

Details

Geographically Weighted Random Forest (GRF) is a spatial analysis method using a local version of the famous Machine Learning algorithm. It allows for the investigation of the existence of spatial non-stationarity, in the relationship between a dependent and a set of independent variables. The latter is possible by fitting a sub-model for each observation in space, taking into account the neighbouring observations. This technique adopts the idea of the Geographically Weighted Regression, Kalogirou (2003). The main difference between a tradition (linear) GWR and GRF is that we can model non-stationarity coupled with a flexible non-linear model which is very hard to over-fit due to its bootstrapping nature, thus relaxing the assumptions of traditional Gaussian statistics. Essentially, it was designed to be a bridge between machine learning and geographical models, combining inferential and explanatory power. Additionally, it is suited for datasets with numerous predictors, due to the robust nature of the random forest algorithm in high dimensionality.

This function is a first attempt to find the optimal bandwidth for the grf. It uses an exhaustive approach, i.e. it tests sequential nearest neighbour bandwidths within a range and with a user defined step, and returns a list of goodness of fit statistics. It chooses the best bandwidth based on the maximum R2 value of the local model. Future versions of this function will include heuristic methods to find the optimal bandwidth using algorithms such as optim.

Value

tested.bandwidths

A table with the tested bandwidths and the corresponding R2 of three model configurations: Local that refers to predictions based on the local (grf) model

only; Mixed that refers to predictions that equally combine local (grf) and global (rf) model predictors; and Low.Local that refers to a prediction based on the combination of the local model predictors with a weight of 0.25 and the global model predictors with a weight of 0.75).

`best.bw` Best bandwidth based on the local model predictions.

Warning

Large datasets may take long time to evaluate the optimal bandwidth.

Note

This function is under development. There should be improvements in future versions of the package `SpatialML`. Any suggestion is welcome!

Author(s)

Stamatis Kalogirou <stamatis.science@gmail.com>, Stefanos Georganos <sgeorgan@ulb.ac.be>

References

Stefanos Georganos, Tais Grippa, Assane Niang Gadiaga, Catherine Linard, Moritz Lennert, Sabine Vanhuysse, Nicholus Odhiambo Mboga, Eléonore Wolff and Stamatis Kalogirou (2019) Geographical Random Forests: A Spatial Extension of the Random Forest Algorithm to Address Spatial Heterogeneity in Remote Sensing and Population Modelling, Geocarto International, DOI: 10.1080/10106049.2019.1595177

Georganos, S. and Kalogirou, S. (2022) A Forest of Forests: A Spatially Weighted and Computationally Efficient Formulation of Geographical Random Forests. ISPRS, International Journal of Geo-Information, 2022, 11, 471. <https://www.mdpi.com/2220-9964/11/9/471>

See Also

`grf`

Examples

```
  ## Not run:
      RDF <- random.test.data(8,8,3)
      Coords<-RDF[ ,4:5]
      bw.test <- grf.bw(dep ~ X1 + X2, RDF, kernel="adaptive",
      coords=Coords, bw.min = 20, bw.max = 23, step = 1,
      forests = FALSE, weighted = TRUE)

## End(Not run)

      data(Income)
      Coords<-Income[ ,1:2]

      bwe <-grf.bw(Income01 ~ UnemrT01 + PrSect01, Income, kernel="adaptive",
                   coords=Coords, bw.min = 30, bw.max = 80, step = 1,
                   forests = FALSE, weighted = TRUE)
```

```
grf <- grf(Income01 ~ UnemrT01 + PrSect01, dframe=Income, bw=bwe$Best.BW,
           kernel="adaptive", coords=Coords)
```

Income *Mean household income at lcoal authorities in Greece in 2011*

Description

Municipality centroids and socioeconomic variables aggregated to the new local authority geography in Greece (Programme Kallikratis).

Usage

```
data(Income)
```

Format

A data frame with 325 observations on the following 5 variables.

`X` a numeric vector of x coordinates

`Y` a numeric vector of y coordinates

`UnemrT01` a numeric vector of total unemployment rate in 2001 (Census)

`PrSect01` a numeric vector of the proportion of economically active working in the primary financial sector (mainly agriculture; fishery; and forestry in 2001 (Census))

`Foreig01` a numeric vector of proportion of people who do not have the Greek citizenship in 2001 (Census)

`Income01` a numeric vector of mean recorded household income (in Euros) earned in 2001 and declared in 2002 tax forms

Details

The X,Y coordinates refer to the geometric centroids of the new 325 Municipalities in Greece (Programme Kallikratis) in 2011.

Source

The original shapefile of the corresponding polygons is available from the Hellenic Statistical Authority (EL.STAT.) at http://www.statistics.gr/el/digital-cartographical-data. The population, employment, citizenship and employment sector data is available from the Hellenic Statistical Authority (EL.STAT.) at http://www.statistics.gr/en/home but were aggregated to the new municipalities by the author. The income data are available from the General Secretariat of Information Systems in Greece at http://www.gsis.gr/ at the postcode level of geography and were aggregated to the new municipalities by the author.

References

Kalogirou, S., and Hatzichristos, T. (2007). A spatial modelling framework for income estimation. Spatial Economic Analysis, 2(3), 297-316. https://www.tandfonline.com/doi/full/10.1080/17421770701576921

Kalogirou, S. (2010). Spatial inequalities in income and post-graduate educational attainment in Greece. Journal of Maps, 6(1), 393-400.https://www.tandfonline.com/doi/abs/10.4113/jom.2010.1095

Kalogirou, S. (2013) Testing geographically weighted multicollinearity diagnostics, GISRUK 2013, Department of Geography and Planning, School of Environmental Sciences, University of Liverpool, Liverpool, UK, 3-5 April 2013.

Examples

```
data(Income)
boxplot(Income$Income01)
hist(Income$PrSect01)
```

`predict.grf` *Predict Method for Geographical Random Forest*

Description

Prediction of test data using the geographical random forest.

Usage

```
## S3 method for class 'grf'
predict(object, new.data, x.var.name, y.var.name, local.w=1, global.w=0,...)
```

Arguments

`object`	an object that created by the function grf that includes all local forests.
`new.data`	a data frame containing new data.
`x.var.name`	the name of the variable with X coordinates.
`y.var.name`	the name of the variable with Y coordinates.
`local.w`	weight of the local model predictor allowing semi-local predictions. Default value is 1.
`global.w`	weight of the global model predictor allowing semi-local predictions. Default value is 0.
`...`	for other arguments passed to the generic predict functions. For example you may pass here the number of threats

Details

A Geographical Random Forest prediction on unknown data. The nearest local random forest model in coordinate space is used to predict in each unknown y-variable location.

Value

vector of predicted values

Note

This function is under development. There should be improvements in future versions of the package SpatialML. Any suggestion is welcome!

Author(s)

Stamatis Kalogirou <stamatis.science@gmail.com>, Stefanos Georganos <sgeorgan@ulb.ac.be>

References

Stefanos Georganos, Tais Grippa, Assane Niang Gadiaga, Catherine Linard, Moritz Lennert, Sabine Vanhuysse, Nicholus Odhiambo Mboga, Eléonore Wolff & Stamatis Kalogirou (2019) Geographical Random Forests: A Spatial Extension of the Random Forest Algorithm to Address Spatial Heterogeneity in Remote Sensing and Population Modelling, Geocarto International, DOI: 10.1080/10106049.2019.1595177

See Also

grf

Examples

```
  ## Not run:
      RDF <- random.test.data(10,10,3)
      Coords<-RDF[ ,4:5]
      grf <- grf(dep ~ X1 + X2, dframe=RDF, bw=10,
                kernel="adaptive", coords=Coords)

      RDF.Test <- random.test.data(2,2,3)

      predict.grf(grf, RDF.Test, x.var.name="X", y.var.name="Y", local.w=1, global.w=0)

## End(Not run)

      #Load the sample data
      data(Income)

      #Create the vector of XY coordinates
      Coords<-Income[,1:2]

      #Fit local model
      grf <- grf(Income01 ~ UnemrT01 + PrSect01, dframe=Income, bw=60,
                kernel="adaptive", coords=Coords)

      #Create New Random Data - XY coordinates inside the sample data map extend
      x<-runif(20, min = 142498, max = 1001578)
      y<-runif(20, min = 3855768, max = 4606754)
      u<-runif(20, min = 5, max = 50)
```

```
p<-runif(20, min = 0, max = 100)
f<-runif(20, min = 2, max = 30)
df2<-data.frame(X=x, Y= y,  UnemrT01=u,  PrSect01=p,  Foreig01=f)

#Make predictions using the local model
predict.grf(grf, df2, x.var.name="X", y.var.name="Y", local.w=1, global.w=0)
```

`random.test.data` *Radmom data generator*

Description

Generates datasets with random data for modelling including a dependent variable, independent variables and X,Y coordinates.

Usage

```
random.test.data(nrows = 10, ncols = 10, vars.no = 3, dep.var.dis = "normal",
                 xycoords = TRUE)
```

Arguments

`nrows`	an integer referring to the number of rows for a regular grid
`ncols`	an integer referring to the number of columns for a regular grid
`vars.no`	an integer referring to the number of independent variables
`dep.var.dis`	a character referring to the distribution of the dependent variable. Options are "normal" (default) and "poisson"
`xycoords`	a logical value indicating whether X,Y coordinates will be created (default) or not.

Details

The creation of a random dataset was necessary here to provide examples to some functions. However, random datasets may be used in simulation studies.

Value

a dataframe

Author(s)

Stamatis Kalogirou <stamatis.science@gmail.com>

Examples

```
RDF <- random.test.data(12,12,3)
```

`rf.mtry.optim` *Optimal mtry*

Description

This function calculates the optimal mtry for a given Random Forest (RF) model in a specified range of values. The optimal mtry value can then be used in the grf model.

Usage

```
rf.mtry.optim(formula, dataset, min.mtry=NULL, max.mtry=NULL, mtry.step,
       cv.method="repeatedcv", cv.folds=10, ...)
```

Arguments

`formula`	the model to be fitted using the function `train` of the R package `caret`.
`dataset`	a numeric data frame of at least two suitable variables (one dependent and one independent)
`min.mtry`	the minimum mtry value for its optimisation (function `expand.grid`)
`max.mtry`	the maximum mtry value for its optimisation (function `expand.grid`)
`mtry.step`	the step in the sequence of mtry values for its optimisation (function `expand.grid`)
`cv.method`	the resampling method in the function `trainControl` of the R package `caret`. Default option is "repeatedcv" and alternative option is "cv".
`cv.folds`	the number of folds (argument "number" in the function `trainControl`). Default value is 10)
`...`	additional arguments affecting the function `trainControl`)

Details

Based on the `train` function of the `caret` package, this function sets up a grid of tuning parameters for a number of random forest routines, fits each model and calculates a resampling based performance measure to choose the best mtry value.

Value

A list is returned of class train as in the function `train` in the `caret` package.

Note

This function is under development.

Author(s)

Stamatis Kalogirou <stamatis.science@gmail.com>, Stefanos Georganos <sgeorgan@ulb.ac.be>

References

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1 - 26. doi: <http://dx.doi.org/10.18637/jss.v028.i05>

Georganos, S. and Kalogirou, S. (2022) A Forest of Forests: A Spatially Weighted and Computationally Efficient Formulation of Geographical Random Forests. ISPRS, International Journal of Geo-Information, 2022, 11, 471. <https://www.mdpi.com/2220-9964/11/9/471>

Examples

```
data(Income)
Coords <- Income[ ,1:2]
results <- rf.mtry.optim(Income01 ~ UnemrT01 + PrSect01, Income)
```

Index