Package 'tbma'

April 6, 2020

Type Package

Title Tree-Based Moving Average Forecasting Model

Version 0.1.0

Date 2020-03-16

Author Burakhan Sel [aut, cre],
Mustafa Gokce Baydogan [aut]

Description We provide a forecasting model for time series forecasting problems with predictors. The offered model, which is based on a submitted research and called tree-based moving average (TBMA), is based on the integration of the moving average approach to tree-based ensemble approach. The tree-based ensemble models can capture the complex correlations between the predictors and response variable but lack in modelling time series components. The integration of the moving average approach to the tree-based ensemble approach helps the TBMA model to handle both correlations and autocorrelations in time series data. This package provides a tbma() forecasting function that utilizes the ranger() function from the 'ranger' package. With the help of the ranger() function, various types of tree-based ensemble models, such as extremely randomized trees and random forests, can be used in the TBMA model.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests knitr, rmarkdown

Imports data.table, ranger, zoo, RcppRoll

NeedsCompilation no

Maintainer Burakhan Sel <burakhan.sel@boun.edu.tr>

Repository CRAN

Date/Publication 2020-04-06 14:20:03 UTC

R topics documented:

---

tbma *Tree-based Moving Average (tbma)*

Description

The tbma() function is used for forecasting problems with predictors. With the help of integrating the moving average approach to tree-based ensemble approach, the function handles the correlations and autocorrelations in time series data. The tree-based ensemble models in the tbma() function is provided by the ranger() function from the 'ranger' package (Marvin N. Wright & Andreas Ziegler, 2017).

Usage

```
tbma(
  formula,
  train,
  test,
  prediction_type = "point",
  percentile = c(0.25, 0.5, 0.75),
  group_id = NULL,
  horizon = nrow(train),
  splitrule = "extratrees",
  always_split_variables = NULL,
  min_node_size = 5,
  max_depth = NULL,
  num_trees = 100,
  ma_order = 2,
  mtry = round(sqrt(ncol(train)))
)
```

Arguments

formula	Object of class formula
train	A data.table object
test	A data.table object
prediction_type	Prediction type can be either "point" or "probabilistic". In case of "probabilistic", percentiel parameter is required.
percentile	Percentile of the probabilistic forecasts if the prediction type is "probabilistic". Percentile paramater can take multiple values between 0 and 1 in a vector.
group_id	Gorup identity parameter is required to filter the data that is going to be used for prediction of a test observations. Group identity parameter is optional to use and usually one of the categorical variables has significant effect on the response variable.

horizon	Horizon parameter filters the train data that is going to be used for forecasting a test observations. The last n train observation is used for forecasting in case of horizon is n. Default value is number of observations in the train set which means no filtering.
splitrule	Splitrule determines the process of splitting. It can be "extratrees","variance", or "maxstat". See the documentation of the 'ranger' package for details.
always_split_variables	Vector of column names indicating the colums that should be selected as candidate variables for splitting. See the documentation of the 'ranger' package for details.
min_node_size	Minimum node size allowed in terminal nodes of decesion trees.
max_depth	Maximum depth of decision trees. See the documentation of the 'ranger' package for details.
num_trees	Number of trees
ma_order	Order of the movinh average part of the TBMA model. Default is 2. High order parameter can lead NA forecasts.
mtry	Number of variables selected as candidate varibles for splitting. See the documentation of the 'ranger' package for details.

Value

A data.table object. In case of point forecasting, a column called "prediction" is added to the data table that contains the columns mentioned in the formula. In case of probabilistic forecasting, columns named with the percentile values are added to thr data table that contains the columns mentioned in the formula.

References

- Wright, M. N. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw 77:1-17. https://doi.org/10.18637/jss.v077.i01.
- Matt Dowle and Arun Srinivasan (2019). data.table: Extension of 'data.frame'. R package version 1.12.8. https://CRAN.R-project.org/package=data.table

Examples

```
library(datasets)
library(data.table)
data(airquality)
summary(airquality)
airquality<-as.data.table(airquality)
airquality[complete.cases(airquality)]
train <- airquality[1:102,]
test <- airquality[103:nrow(airquality), ]
test_data_with_predictions<-tbma(Temp ~ .,train = train,test = test,
prediction_type = "point",horizon=100,ma_order = 2)
```

Index