Package 'setartree'

August 24, 2023

Title SETAR-Tree - A Novel and Accurate Tree Algorithm for Global Time Series Forecasting

Version 0.2.1

Maintainer Rakshitha Godahewa <rakshithagw@gmail.com>

Description The implementation of a forecasting-specific tree-based model that is in particular suitable for global time series forecasting, as proposed in Godahewa et al. (2022) <arXiv:2211.08661v1>. The model uses the concept of Self Exciting Threshold Autoregressive (SETAR) models to define the node splits and thus, the model is named SETAR-Tree. The SETAR-Tree uses some time-series-specific splitting and stopping procedures. It trains global pooled regression models in the leaves allowing the models to learn cross-series information. The depth of the tree is controlled by conducting a statistical linearity test as well as measuring the error reduction percentage at each node split. Thus, the SETAR-Tree requires minimal external hyperparameter tuning and provides competitive results under its default configuration. A forest is developed by extending the SETAR-Tree. The SETAR-Forest combines the forecasts provided by a collection of diverse SETAR-Trees during the forecasting process.

License MIT + file LICENSE

URL https://github.com/rakshitha123/setartree

BugReports https://github.com/rakshitha123/setartree/issues

Depends R (>= 3.5.0)

Imports stats, utils, methods, parallel, generics (>= 0.1.2)

Suggests forecast

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Rakshitha Godahewa [cre, aut, cph],
Christoph Bergmeir [aut],
Daniel Schmidt [aut],
Geoffrey Webb [ctb]

Repository CRAN

Date/Publication 2023-08-24 11:00:02 UTC

R topics documented:

setartree-package *Getting started with the setartree package*

Description

The setartree is a library containing the implementations of SETAR-Tree and SETAR-Forest which are forecasting-specific tree-based models that are in particular suitable for global time series forecasting.

Details

If you have problems using setartree, find a bug, or have suggestions, please file an issue on github (bugs/suggestions). If that fails, then you can contact the maintainer directly by email.

If you use the package, please cite the following work in your publications:

Godahewa, R., Webb, G. I., Schmidt, D., & Bergmeir, C. (2023). SETAR-Tree: A novel and accurate tree algorithm for global time series forecasting. Machine Learning, 112, 2555-2591. doi: 10.1007/s1099402306316x

Demos for using SETAR-Tree and SETAR-Forest are available. To get a list of them, type:

```
library(setartree)
```

```
demo()
```

To execute the SETAR-Tree demo, type:

```
demo(tree_demo)
```

To execute the SETAR-Forest demo, type:

```
demo(forest_demo)
```

To fit a SETAR-Tree model either using a list of time series or an embedded input matrix and labels, use the function `setartree`. To fit a SETAR-Forest model either using a list of time series or an embedded input matrix and labels, use the function `setarforest`. To obtain forecasts from a SETAR-Tree or a SETAR-Forest, use the functions `forecast.setartree` and `forecast.setarforest`, respectively.

The setartree package also contains three datasets that can be used to train/test the SETAR-Tree and SETAR-Forest models: `chaotic_logistic_series`, `web_traffic_train` and `web_traffic_test`.

See the setartree user manual for detailed explanations about the datasets and the parameters taken by each function.

Another nice tool is the `forecast` package, that can be used to plot the time series together with the forecasts generated by SETAR-Tree or SETAR-Forest.

Author(s)

Rakshitha Godahewa <`rakshithagw@gmail.com`>

Christoph Bergmeir <`christoph.bergmeir@monash.edu`>

Daniel Schmidt <`daniel.schmidt@monash.edu`>

and Geoffrey Webb <`geoff.webb@monash.edu`>

Department of Data Science and AI, Faculty of Information Technology, Monash University, Australia.

`https://www.monash.edu/it/dsai`

References

Godahewa, R., Webb, G. I., Schmidt, D., & Bergmeir, C. (2023). SETAR-Tree: A novel and accurate tree algorithm for global time series forecasting. Machine Learning, 112, 2555-2591. doi: 10.1007/s1099402306316x

---

`chaotic_logistic_series`

Chaotic logistic map example time series

Description

A list of 20 time series constructed by using the Chaotic Logistic Map (May, 1976) data generation process. This is a part of a simulated dataset used in Hewamalage et al.(2021). These series can be used to train the SETAR-Tree and SETAR-Forest models.

Format

A list containing 20 numerical vectors.

References

May, R. M. (1976). Simple mathematical models with very complicated dynamics. Nature, 261, 459–467.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Global models for time series forecasting: A simulation study. Pattern Recognition, 108441.

Examples

```
chaotic_logistic_series
```

forecast.setarforest *Forecast method for SETAR-Forest fits*

Description

Obtains forecasts for a given set of time series or a dataframe/matrix of new instances from a fitted SETAR-Forest model.

Usage

```
## S3 method for class 'setarforest'
forecast(object, newdata, h = 5, level = c(80, 95), ...)
```

Arguments

`object`	An object of class `setarforest` which is a trained SETAR-Forest model.
`newdata`	A list of time series which need forecasts or a dataframe/matrix of new instances which need predictions.
`h`	The required number of forecasts (forecast horizon). This parameter is only required when `newdata` is a list of time series. Default value is 5.
`level`	Confidence level for prediction intervals. Default value is c(80, 95).
`...`	Other arguments.

Value

If `newdata` is a list of time series, then an object of class `mforecast` is returned. The `plot` or `autoplot` functions in the R `forecast` package can then be used to produce a plot of any time series in the returned object which contains the following properties.

`method`	A vector containing the name of the forecasting method ("SETAR-Forest").
`forecast`	A list of objects of class `forecast`. Each list object is corresponding with a time series and its forecasts. Each list object contains 7 properties: method (the name of the forecasting method, SETAR-Forest, as a character string), x (the original time series), mean (point forecasts as a time series), series (the name of the series as a character string), upper (upper bound of confidence intervals), lower (lower bound of confidence intervals) and level (confidence level of prediction intervals).

If `newdata` is a dataframe/matrix, then a list containing the prediction and prediction intervals (upper and lower bounds) of each instance is returned.

Examples

```
# Obtaining forecasts for a list of time series
forest1 <- setarforest(chaotic_logistic_series, bagging_freq = 2, num_cores = 1)
forecast(forest1, chaotic_logistic_series)

# Obtaining forecasts for a set of test instances
forest2 <- setarforest(data = web_traffic_train[,-1],
                       label = web_traffic_train[,1],
                       bagging_freq = 2,
                       num_cores = 1,
                       categorical_covariates = "Project")
forecast(forest2, web_traffic_test)
```

forecast.setartree *Forecast method for SETAR-Tree fits*

Description

Obtains forecasts for a given set of time series or a dataframe/matrix of new instances from a fitted SETAR-Tree model.

Usage

```
## S3 method for class 'setartree'
forecast(object, newdata, h = 5, level = c(80, 95), ...)
```

Arguments

`object` An object of class `setartree` which is a trained SETAR-Tree model.

`newdata` A list of time series which needs forecasts or a dataframe/matrix of new instances which need predictions.

`h` The required number of forecasts (forecast horizon). This parameter is only required when `newdata` is a list of time series. Default value is 5.

`level` Confidence level for prediction intervals. Default value is c(80, 95).

`...` Other arguments.

Value

If `newdata` is a list of time series, then an object of class `mforecast` is returned. The `plot` or `autoplot` functions in the R `forecast` package can then be used to produce a plot of any time series in the returned object which contains the following properties.

`method` A vector containing the name of the forecasting method ("SETAR-Tree").

`forecast` A list of objects of class `forecast`. Each list object is corresponding with a time series and its forecasts. Each list object contains 7 properties: method (the name of the forecasting method, SETAR-Tree, as a character string), x (the original time series), mean (point forecasts as a time series), series (the name of the series as a character string), upper (upper bound of confidence intervals), lower (lower bound of confidence intervals) and level (confidence level of prediction intervals).

If `newdata` is a dataframe/matrix, then a list containing the predictions, prediction intervals (upper and lower bounds), the size and standard deviations of the residuals of the models used to get each prediction is returned.

Examples

```
# Obtaining forecasts for a list of time series
tree1 <- setartree(chaotic_logistic_series)
forecast(tree1, chaotic_logistic_series)

# Obtaining forecasts for a set of test instances
tree2 <- setartree(data = web_traffic_train[,-1],
                   label = web_traffic_train[,1],
                   stopping_criteria = "both",
                   categorical_covariates = "Project")
forecast(tree2, web_traffic_test)
```

setarforest *Fitting SETAR-Forest models*

Description

Fits a SETAR-Forest model either using a list of time series or an embedded input matrix and labels.

Usage

```
setarforest(
  data,
  label = NULL,
  lag = 10,
  bagging_fraction = 0.8,
  bagging_freq = 10,
  random_tree_significance = TRUE,
  random_tree_significance_divider = TRUE,
  random_tree_error_threshold = TRUE,
  depth = 1000,
  significance = 0.05,
  significance_divider = 2,
  error_threshold = 0.03,
  stopping_criteria = "both",
  mean_normalisation = FALSE,
  window_normalisation = FALSE,
  verbose = 2,
  num_cores = NULL,
  categorical_covariates = NULL
)
```

Arguments

`data` A list of time series (each list element is a separate time series) or a dataframe/matrix containing model inputs (the columns can contain past time series lags and/or external numerical/categorical covariates).

`label` A vector of true outputs. This parameter is only required when `data` is a dataframe/matrix containing the model inputs.

`lag` The number of past time series lags that should be used when fitting each SETAR-Tree in the forest. This parameter is only required when `data` is a list of time series. Default value is 10.

`bagging_fraction` The percentage of instances that should be used to train each SETAR-Tree in the forest. Default value is 0.8.

`bagging_freq` The number of SETAR-Trees in the forest. Default value is 10.

`random_tree_significance` Whether a random significance should be considered for splitting per each tree. Each node split within the tree considers the same significance level. When this parameter is set to TRUE, the "significance" parameter will be ignored. Default value is TRUE.

`random_tree_significance_divider` Whether a random significance divider should be considered for splitting per each tree. When this parameter is set to TRUE, the "significance_divider" parameter will be ignored. Default value is TRUE.

`random_tree_error_threshold` Whether a random error threshold should be considered for splitting per each tree. Each node split within the tree considers the same error threshold. When this parameter is set to TRUE, the "error_threshold" parameter will be ignored. Default value is TRUE.

`depth` Maximum depth of each SETAR-Tree in the forest. Default value is 1000. Thus, unless specify a lower value, the depth of a SETAR-Tree is actually controlled by the stopping criterion.

`significance` In each SETAR-Tree in the forest, the initial significance used by the linearity test (alpha_0). Default value is 0.05.

`significance_divider` In each SETAR-Tree in the forest, the corresponding significance in a tree level is divided by this value. Default value is 2.

`error_threshold`
In each SETAR-Tree in the forest, the minimum error reduction percentage between parent and child nodes to make a split. Default value is 0.03.

`stopping_criteria`
The required stopping criteria for each SETAR-Tree in the forest: linearity test (lin_test), error reduction percentage (error_imp) or linearity test and error reduction percentage (both). Default value is `"both"`.

`mean_normalisation`
Whether each series should be normalised by deducting its mean value before building the forest. This parameter is only required when `data` is a list of time series. Default value is FALSE.

`window_normalisation`
Whether the window-wise normalisation should be applied before building the forest. This parameter is only required when `data` is a list of time series. When this is TRUE, each row of the training embedded matrix is normalised by deducting its mean value before building the forest. Default value is FALSE.

`verbose` Controls the level of the verbosity of SETAR-Forest: 0 (errors/warnings), 1 (limited amount of information including the depth of the currently processing tree), 2 (full training information including the depth of the currently processing tree and stopping criterion related details in each tree). Default value is 2.

`num_cores` The number of cores to be used. `num_cores` > 1 means parallel processing. When not provided, it will find the available number of cores and use those to run the SETAR-Trees in the forest in parallel.

`categorical_covariates`
Names of the categorical covariates in the input data. This parameter is only required when `data` is a dataframe/matrix and it contains categorical variables.

Value

An object of class `setarforest` which contains the following properties.

`trees` A list of objects of class `setartree` which represents the trained SETAR-Tree models in the forest.

`lag` The number of features used to train each SEATR-Tree in the forest.

`feature_names` Names of the input features.

`coefficients` Names of the coefficients of leaf node regresion models in each SETAR-Tree in the forest.

`categorical_covariate_values`
Information about the categorical covarites used during training (only if applicable).

`mean_normalisation`
Whether mean normalisation was applied for the training data.

`window_normalisation`
Whether window normalisation was applied for the training data.

`input_type` Type of input data used to train the SETAR-Forest. This is `list` if `data` is a list of time series, and `df` if `data` is a dataframe/matrix containing model inputs.

`execution_time` Execution time of SETAR-Forest.

Examples

```
# Training SETAR-Forest with a list of time series
setarforest(chaotic_logistic_series, bagging_freq = 2, num_cores = 1)

# Training SETAR-Forest with a dataframe containing model inputs where the model inputs may contain
# past time series lags and numerical/categorical covariates
setarforest(data = web_traffic_train[,-1],
            label = web_traffic_train[,1],
            bagging_freq = 2,
            num_cores = 1,
            categorical_covariates = "Project")
```

setartree *Fitting SETAR-Tree models*

Description

Fits a SETAR-Tree model either using a list of time series or an embedded input matrix and labels.

Usage

```
setartree(
  data,
  label = NULL,
  lag = 10,
  depth = 1000,
  significance = 0.05,
  significance_divider = 2,
  error_threshold = 0.03,
  stopping_criteria = "both",
  mean_normalisation = FALSE,
  window_normalisation = FALSE,
  verbose = 2,
  categorical_covariates = NULL
)
```

Arguments

`data` A list of time series (each list element is a separate time series) or a dataframe/matrix containing model inputs (the columns can contain past time series lags and/or external numerical/categorical covariates).

`label` A vector of true outputs. This parameter is only required when `data` is a dataframe/matrix containing the model inputs.

`lag` The number of past time series lags that should be used when fitting the SETAR-Tree. This parameter is only required when `data` is a list of time series. Default value is 10.

`depth` Maximum tree depth. Default value is 1000. Thus, unless specify a lower value, the depth is actually controlled by the stopping criterion.

`significance` Initial significance used by the linearity test (alpha_0). Default value is 0.05.

`significance_divider` The corresponding significance in each tree level is divided by this value. Default value is 2.

`error_threshold` The minimum error reduction percentage between parent and child nodes to make a split. Default value is 0.03.

`stopping_criteria` The required stopping criteria: linearity test (lin_test), error reduction percentage (error_imp) or linearity test and error reduction percentage (both). Default value is `"both"`.

`mean_normalisation` Whether each series should be normalised by deducting its mean value before building the tree. This parameter is only required when `data` is a list of time series. Default value is FALSE.

`window_normalisation` Whether the window-wise normalisation should be applied before building the tree. This parameter is only required when `data` is a list of time series. When this is TRUE, each row of the training embedded matrix is normalised by deducting its mean value before building the tree. Default value is FALSE.

`verbose` Controls the level of the verbosity of SETAR-Tree: 0 (errors/warnings), 1 (limited amount of information including the current tree depth), 2 (full training information including the current tree depth and stopping criterion results in each tree node). Default value is 2.

`categorical_covariates` Names of the categorical covariates in the input data. This parameter is only required when `data` is a dataframe/matrix and it contains categorical variables.

Value

An object of class `setartree` which contains the following properties.

`leaf_models` Trained global pooled regression models in each leaf node.

`opt_lags` Optimal features used to split each node.

`opt_thresholds` Optimal threshold values used to split each node.

`lag` The number of features used to train the SETAR-Tree.

`feature_names` Names of the input features.

`coefficients` Names of the coefficients of leaf node regresion models.

`num_leaves` Number of leaf nodes in the SETAR-Tree.

`depth` Depth of the SETAR-Tree which was determined based on the specified stopping criterion.

`leaf_instance_dis`
Number of instances used to train the regression models at each leaf node.

`stds` The standard deviations of the residuals of each leaf node.

`categorical_covariate_values`
Information about the categorical covarites used during training (only if applicable).

`mean_normalisation`
Whether mean normalisation was applied for the training data.

`window_normalisation`
Whether window normalisation was applied for the training data.

`input_type` Type of input data used to train the SETAR-Tree. This is `list` if `data` is a list of time series, and `df` if `data` is a dataframe/matrix containing model inputs.

`execution_time` Execution time of SETAR-Tree.

Examples

```
# Training SETAR-Tree with a list of time series
setartree(chaotic_logistic_series)

# Training SETAR-Tree with a dataframe containing model inputs where the model inputs may contain
# past time series lags and numerical/categorical covariates
setartree(data = web_traffic_train[,-1],
          label = web_traffic_train[,1],
          stopping_criteria = "both",
          categorical_covariates = "Project")
```

`web_traffic_test` *A dataframe of test instances*

Description

A dataframe containing 5 instances that can be used to test the SETAR-Tree and SETAR-Forest models. The data are related to the number of hits or web traffic of a set of Wikipedia pages. Each intance consists of 10 time series lags (Lag1 to Lag10) and a categorical covariate (Project). The data were downloaded from the Wikimedia REST API (Wikimedia, 2022).

Format

A dataframe containing 5 test instances.

References

Wikimedia Analytics Team (2022). Wikistats: Pageview complete dumps.
URL https://dumps.wikimedia.org/other/pageview_complete

Examples

```
web_traffic_test
```

web_traffic_train *A dataframe of training instances*

Description

A dataframe containing 120 instances that can be used to train the SETAR-Tree and SETAR-Forest models. The data are related to the number of hits or web traffic of a set of Wikipedia pages. Each instance in the dataframe consists of 10 time series lags (Lag1 to Lag10), a categorical covariate (Project) and the corresponding true outputs (y). The data were downloaded from the Wikimedia REST API (Wikimedia, 2022).

Format

A dataframe containing 120 training instances.

References

Wikimedia Analytics Team (2022). Wikistats: Pageview complete dumps.
URL https://dumps.wikimedia.org/other/pageview_complete

Examples

```
web_traffic_train
```

Index