Package ‘audrex’

April 28, 2021

Type Package
Title Automatic Dynamic Regression using Extreme Gradient Boosting
Version 1.0.0
Author Giancarlo Vercellino
Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>
Description Dynamic regression for time series using Extreme Gradient Boosting with hyper-parameter tuning via Bayesian Optimization.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Depends R (>= 3.6)
Imports rBayesianOptimization, xgboost, purrr, abind, ggplot2, readr, stringr, lubridate, narray, fANCOVA, imputeTS, scales, tictoc, bizdays
NeedsCompilation no
Repository CRAN
Date/Publication 2021-04-28 13:30:02 UTC

R topics documented:

audrex ................................................................. 2
bitcoin_gold_oil .................................................. 5

Index 6
**Description**

audrex

**Usage**

```r
audrex(
  data,
  targets,
  past,
  deriv,
  future,
  shift = 0,
  smoother = FALSE,
  ci = 0.8,
  holdout = 0.3,
  nrounds = 100,
  patience = 10,
  booster = "gbtree",
  max_depth = NULL,
  eta = NULL,
  gamma = NULL,
  min_child_weight = NULL,
  subsample = NULL,
  colsample_bytree = NULL,
  lambda = NULL,
  alpha = NULL,
  verbose = FALSE,
  reg = "squarederror",
  eval_metric = "rmse",
  starting_date = NULL,
  dbreak = NULL,
  days_off = NULL,
  min_set = 30,
  seed = 42,
  opt_metric = "mae",
  n_samp = 15,
  n_search = 15,
  acq = "ucb",
  kappa = 2.576,
  eps = 0,
  kernel = list(type = "exponential", power = 2)
)
```
Arguments

data: A data frame with time series on columns and possibly a date column (not mandatory).

targets: String. Names of ts features to be jointly analyzed: for each feature a distinct model is built using the others as regressors.

past: Positive integer. The past dimension with number of time-steps in the past used for the prediction.

deriv: Positive integer. Number of differentiation operations to perform on the original series. 0 = no change; 1: one diff; 2: two diff, and so on.

future: Positive integer. The future dimension with number of time-steps to be predicted.

shift: Vector of positive integers. Allow for each ts feature to shift ahead of time. Zero means no shift. Length must be equal to the number of targets. Default: 0.

smoother: Logical. Perform optimal smoothing using standard loess. Default: FALSE

ci: Confidence interval. Default: 0.8

holdout: Positive numeric. Percentage of time series for holdout validation. Default: 0.5.


patience: Positive integer. Waiting rounds without improvement before xgboost stops. Default: 10

booster: String. Optimization methods available are: "gbtree", "gblinear". Default: "gbtree".

max_depth: Positive integer. Look to xgboost documentation for description. A vector with one or two positive integer for the search boundaries. The default value (NULL) sets automatic the values in c(1, 10).

eta: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatic the values in c(0.001, 1).

gamma: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatic the values in c(0.001, 100).

min_child_weight: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatic the values in c(1, 100).

subsample: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatic the values in c(0.1, 1).

colsample_bytree: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatic the values in c(0.1, 1).

lambda: Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatic the values in c(0.1, 100).
alpha Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatic the values in c(0.1, 100).

verbose Logical. Default: TRUE

reg String. Learning objective function. Options are: "squarederror", "pseudohubererror". Default: "squarederror".

eval_metric String. Evaluation metric for the boosting algorithm. Options are: "rmse", "mae", "mape". Default: "mae".

starting_date Date. Initial date to assign temporal values to the series. Default: NULL (progressive numbers).

dbreak String. Minimum time marker for x-axis, in liberal form: i.e., "3 months", "1 week", "20 days". Default: NULL.

days_off String. Weekdays to exclude (i.e., c("saturday", "sunday")). Default: NULL.


seed Random seed. Default: 42.

opt_metric String. Parameter for selecting the best model, averaging one-step error across all ts features. Default: "mae".

n_samp Positive integer. Number of samples for the Bayesian Optimization. Default: 15.

n_search Positive integer. Number of search steps for the Bayesian Optimization. Default: 15.

acq String. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: "ucb".


eps Positive numeric. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: 0.

kernel List. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: list(type = "exponential", power = 2).

Value

This function returns a list including:

- best_par: the parameter of the best model selected through Bayesian Optimization
- history: a table with the sampled models (n_samp + n_search), their parameters and optimization metric
- best_model: results for the best selected model, including:
  - errors: training and testing errors for one-step and sequence for each ts feature (rmse, mae, mdae, mpe, mape, smape)
  - predictions: min, max, q25, q50, q75, quantiles at selected ci, mean, sd for each ts feature
  - pred_stats: some stats for each ts feature (iqr to min-max range, last-to-first iqr ratio, upside probability from ecdf)
- time_log
Author(s)
Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Examples

audrex(bitcoin_gold_oil, c("gold_close", "oil_Close"), past = 30, deriv = 1, future = 10)

bitcoin_gold_oil  bitcoin_gold_oil data set

Description
A data frame with different time series (prices and volumes) for bitcoin, gold and oil.

Usage
bitcoin_gold_oil

Format
A data frame with 18 columns and 1827 rows.

Source
Yahoo Finance
Index

* datasets
  bitcoin_gold_oil, 5

audrex, 2

bitcoin_gold_oil, 5