

Package ‘chopper’

July 22, 2025

Type Package

Title Changepoint-Aware Ensemble for Probabilistic Modeling

Version 1.0

Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Description Implements a changepoint-aware ensemble forecasting algorithm that combines Theta, TBATS (Trigonometric, Box-Cox transformation, ARMA errors, Trend, Seasonal components), and ARFIMA (AutoRegressive, Fractionally Integrated, Moving Average) using a product-of-experts approach for robust probabilistic prediction.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports normalp (>= 0.7.2), purrr (>= 1.0.1), ald (>= 1.3.1), evd (>= 2.3-6.1), GeneralizedHyperbolic (>= 0.8-6), fGarch (>= 4022.89), forecast (>= 8.21), imputeTS (>= 3.3), changepoint (>= 2.3), lubridate (>= 1.9.2), ggplot2 (>= 3.5.1), scales (>= 1.3.0)

URL https://rpubs.com/giancarlo_vercellino/chopper

Suggests knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

NeedsCompilation no

Author Giancarlo Vercellino [aut, cre, cph]

Repository CRAN

Date/Publication 2025-05-27 08:00:06 UTC

Contents

chopper	2
ts_set	4

Index	5
-------	---

chopper

*chopper: Change Point Product-of-Experts Ensemble***Description**

Implements a change point-aware ensemble forecasting algorithm that combines Theta, TBATS, and ARFIMA models using a product-of-experts approach for robust probabilistic prediction

A changepoint-aware probabilistic forecasting algorithm for univariate time series. This function decomposes the input series into structurally homogeneous segments to detect changepoints in both mean and variance. On each segment, three distinct probabilistic forecasting models are applied—TBATS, Theta and ARFIMA to estimate future trajectories.

Usage

```
chopper(
  ts,
  horizon,
  decay = 0.5,
  n_samples = 1000,
  alt_mode = FALSE,
  reverse = FALSE
)
```

Arguments

ts	A numeric vector representing the input time series.
horizon	An integer specifying the number of future time steps to forecast.
decay	A numeric value controlling the exponential decay rate used to the softmax weight forecasts from each segment (default is 0.5).
n_samples	An integer specifying the number of samples drawn for estimating the forecast distributions (default is 1000).
alt_mode	Logical flag: when TRUE, fits models on independent segments between changepoints instead of cumulative segments up to each changepoint (default: FALSE).
reverse	Logical flag: when TRUE, inverts the weight vector (default: FALSE).

Details

Instead of relying on a single model, ‘chopper()’ embraces the variety of probabilistic forecasts by merging their output using a product-of-experts composition rule. This method allows each model to contribute to the final distribution proportionally to the confidence it assigns to different outcomes.

The final output is a collection of forecast distributions—one for each horizon step—that captures both epistemic uncertainty (from model disagreement) and aleatoric uncertainty (from residual variability).

This function uses the ‘changepoint’ package to perform offline changepoint detection on the historical series. For each segment identified, four forecasting models are fit:

- **Theta**: decomposed trend extrapolation with adjusted curvature.
- **TBATS**: based on exponential smoothing, designed for time series data with multiple seasonalities.
- **ARFIMA**: a long-memory extension of ARIMA that captures persistence in volatility and mean.

The output of each model is converted to a forecast density for each horizon step. These are multiplied together (product of experts) and normalized over a shared evaluation grid. A weighted average of these densities is constructed across segments using segment-specific weights based on softmax and exponential decay.

Value

A named list of forecast distributions, one for each time step in the forecast horizon. Weights for each segment are assigned based on the softmax of exponential decay calculated at the end position of each segment.

Each distribution is represented as a list containing:

rfun A random sampling function for drawing synthetic future values.

dfun A density function over the predicted values.

pfun A cumulative distribution function.

qfun A quantile function.

Author(s)

Maintainer: Giancarlo Vercellino <giancarlo.vercellino@gmail.com> [copyright holder]

See Also

Useful links:

- https://rpubs.com/giancarlo_vercellino/chopper

Examples

```
# Fast example with the tail of a time series
forecasted <- chopper(ts = tail(ts_set$AMZN.Close, 70), horizon = 5, n_samples = 100)

# Draw samples from forecast for t1 and t5
forecasted$pred_funs$t1$rfun(10)
forecasted$pred_funs$t5$rfun(10)
```

`ts_set`*Tech Stock Time Series Dataset*

Description

A multivariate dataset for closing prices for several major tech stocks over time. Source: YahooFinance.

Usage

```
data(ts_set)
```

Format

A data frame with 2095 observations of 5 variables:

dates Character vector of dates in "YYYY-MM-DD" format.

TSLA.Close Numeric. Closing prices for Tesla (TSLA).

AMZN.Close Numeric. Closing prices for Amazon (AMZN).

IBM.Close Numeric. Closing prices for IBM.

META.Close Numeric. Closing prices for Meta (Facebook).

Examples

```
data(ts_set)
plot(as.Date(ts_set$dates), ts_set$TSLA.Close, type = "l")
```

Index

* **datasets**

ts_set, [4](#)

chopper, [2](#)

chopper-package (chopper), [2](#)

ts_set, [4](#)