Package 'bridgr'

December 13, 2024

Type Package

Title Bridging Data Frequencies for Timely Economic Forecasts

Version 0.1.1

Maintainer Marc Burri <marc.burri91@gmail.com>

Description Implements bridge models for nowcasting and forecasting macroeconomic variables by linking high-frequency indicator variables (e.g., monthly data) to low-frequency target variables (e.g., quarterly GDP). Simplifies forecasting and aggregating indicator variables to match the target frequency, enabling timely predictions ahead of official data releases. For more on bridge models, see Baffigi, A., Golinelli, R., & Parigi, G. (2004) <doi:10.1016/S0169-2070(03)00067-0>, Burri (2023) <https://www5.unine.ch/RePEc/ftp/irn/pdfs/WP23-02.pdf> or Schumacher (2016) <doi:10.1016/j.ijforecast.2015.07.004>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

Imports magrittr, dplyr, rlang, generics, tsbox, lubridate, forecast

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

Collate 'bridge.R' 'bridgr-package.R' 'data-indicators.R' 'forecast.R' 'summary.R' 'utils.R'

URL https://github.com/marcburri/bridgr, https://marcburri.github.io/bridgr/

BugReports https://github.com/marcburri/bridgr/issues

VignetteBuilder knitr

NeedsCompilation no

Author Marc Burri [aut, cre, cph] (<https://orcid.org/0000-0001-8974-9090>)

Repository CRAN

Date/Publication 2024-12-13 16:40:09 UTC

Contents

`bridge` *Estimate a Bridge Model*

Description

This function estimates a bridge model, aligning high-frequency indicator variables with a lower-frequency target variable to perform nowcasting or forecasting. The bridge model leverages time series alignment, lag structures, and forecasting methods to provide a comprehensive tool for time series analysis.

Usage

```
bridge(
  target,
  indic,
  indic_predict = NULL,
  indic_aggregators = NULL,
  indic_lags = 0,
  target_lags = 0,
  h = 1,
  frequency_conversions = c(dpw = 5, wpm = 4, mpq = 3, qpy = 4),
  ...
)
```

Arguments

`target`	A time series or data frame representing the target variable (dependent variable). Must be in a format compatible with the tsbox package (see `tsbox::ts_boxable()`).
`indic`	A time series, list of time series, or data frame containing the indicator variables (independent variables). Must be in a format compatible with the tsbox package (see `tsbox::ts_boxable()`).
`indic_predict`	A character string or vector specifying the forecasting method(s) for the indicator variables. Supported methods include `"mean"`, `"last"`, `"auto.arima"`, and `"ets"`. Defaults to `"auto.arima"`.
`indic_aggregators`	A character string or vector specifying the aggregation method(s) for aligning indicator variables with the target variable. Supported methods include `"mean"`, `"last"`, `"expalmon"`, `"sum"` or o custom vector of weights with the same length as the frequency ratio. Defaults to `"mean"`.

`indic_lags` An integer or vector of integers specifying the number of lags to include for the indicator variables. Defaults to 0 (no lags).

`target_lags` An integer specifying the number of lags to include for the target variable. Defaults to 0 (no lags).

`h` An integer specifying the forecast horizon in terms of the target variable's frequency. Defaults to 1 (next period).

`frequency_conversions` A named vector specifying the conversion factors between different time frequencies. Defaults to `c("dpw" = 5, "wpm" = 4, "mpq" = 3, "qpy" = 4)` for days per week, weeks per month, months per quarter, and quarters per year, respectively.

`...` Additional arguments for future extension, not used at the moment.

Details

The bridge model aligns time series of different frequencies by slicing and aggregating indicator variables to match the target variable's frequency. It uses predefined rules for frequency conversion and alignment. The function checks for mismatches in start dates and aligns the variables when necessary.

Forecasting methods for the indicator variables:

- `auto.arima`: Automatically selects the best ARIMA (AutoRegressive Integrated Moving Average) model for a given time series based on information criteria (e.g., AIC, AICc, BIC). The method identifies the orders of the AR (p), differencing (d), and MA (q) components and estimates the model parameters. It is particularly suitable for time series with seasonality, trends, or other non-stationary patterns.
- `ets`: Fits an exponential smoothing state-space model to the data. The ETS framework automatically includes Error (additive or multiplicative), Trend (none, additive, or damped), and Seasonal (none, additive, or multiplicative) components. This method is effective for capturing underlying patterns in the data such as level, trend, and seasonality, making it suitable for time series with these features.

Aggregation methods for the indicator variables:

- `mean`: Calculates the mean of the indicator variable values within each target period.
- `last`: Takes the last value of the indicator variable within each target period.
- `expalmon`: Estimates a nonlinear exponential almon lag polynomial for weighting the indicator.
- `sum`: Calculates the sum of the indicator variable values within each target period.
- **Custom weights**: Allows the user to specify custom weights for aggregating the indicator variables.

Value

An object of class `"bridge"` containing:

- **target**: The standardized target variable.
- **indic**: The standardized indicator variables.

- **indic_predict**: The prediction methods applied to the indicators.
- **indic_aggregators**: The aggregation methods used for the indicators.
- **estimation_set**: A data frame containing the aligned and processed time series used to estimate the bridge model. This set includes the target variable and all indicator variables transformed to match the target variable's frequency and alignment.
- **forecast_set**: A data frame containing the aligned and processed time series used for forecasting. This includes the forecasts for the indicator variables as inputs for the h-step ahead prediction of the target variable.
- **model**: The fitted bridge model object for the target variable.
- **indic_models**: A list of models used to forecast the indicator variables. Each element in this list corresponds to the forecasting method (e.g., `auto.arima` or `ets`) applied to an individual indicator variable.
- **Additional components**: Internal parameters, summary statistics, and alignment metadata.

Author(s)

Marc Burri

References

- Baffigi, A., Golinelli, R., & Parigi, G. (2004). Bridge models to forecast the euro area GDP. International Journal of Forecasting, 20(3), 447–460. doi:10.1016/S01692070(03)000670
- Burri, M. (2023). Do daily lead texts help nowcasting GDP growth? IRENE Working Papers 23-02. https://www5.unine.ch/RePEc/ftp/irn/pdfs/WP23-02.pdf
- Schumacher, C. (2016). A comparison of MIDAS and bridge equations. International Journal of Forecasting, 32(2), 257–270. doi:10.1016/j.ijforecast.2015.07.004

Examples

```
library(bridgr)

# Example usage
target_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2022-12-01"), by = "quarter"),
  value = rnorm(12)
)))

indic_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2023-01-01"), by = "month"),
  value = rnorm(37)
)))

bridge_model <- suppressMessages(bridge(
  target = target_series,
  indic = indic_series,
  indic_predict = "mean",
  indic_aggregators = "mean",
  indic_lags = 2,
```

```
  target_lags = 1,
  h = 1
))
```

`forecast.bridge` *Forecasting from a* `bridge` *object*

Description

This function is used to forecast the target variable using the fitted `bridge` model.

Usage

```
## S3 method for class 'bridge'
forecast(object, xreg = NULL, ...)
```

Arguments

`object`	A `bridge` object obtained from `bridge()`.
`xreg`	A `tsbox::ts_boxable()` series of external regressors. If not supplied by the user, the forecast set calculated by `bridge()` will be used. This is useful for made up scenarios.
`...`	Additional arguments to be passed to the forecast function. Ignored at the moment.

Value

An object of class `"forecast"`. An object of class `"forecast"` is a list usually containing at least the following elements:

`model`	A list containing information about the fitted model
`method`	The name of the forecasting method as a character string
`mean`	Point forecasts as a time series
`lower`	Lower limits for prediction intervals
`upper`	Upper limits for prediction intervals
`level`	The confidence values associated with the prediction intervals
`x`	The original time series (either `object` itself or the time series used to create the model stored as `object`).
`residuals`	Residuals from the fitted model. For models with additive errors, the residuals will be x minus the fitted values.
`fitted`	Fitted values (one-step forecasts)
`forecast_set`	The forecast set with the forecasted indicator variables used to calculate the forecast of the target variable.

Examples

```
library(bridgr)

# Example usage
target_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2022-12-01"), by = "quarter"),
  value = rnorm(12)
)))

indic_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2023-01-01"), by = "month"),
  value = rnorm(37)
)))

bridge_model <- suppressMessages(bridge(
  target = target_series,
  indic = indic_series,
  indic_predict = "mean",
  indic_aggregators = "mean",
  indic_lags = 2,
  target_lags = 1,
  h = 1
))

# Forecasting using the bridge model
fcst <- forecast(bridge_model)
```

---

gdp *Swiss Economic Indicators*

Description

A collection of datasets containing economic indicators for Switzerland.

Usage

```
gdp

baro

wea

fcurve
```

Details

- `baro`: The KOF barometer, a monthly business cycle indicator.
- `fcurve`: The F-curve, a daily business cycle indicator.
- `gdp`: Quarterly GDP data (real, seasonally adjusted).
- `wea`: The Weekly Economic Activity (WEA) indicator.

Datasets

- `baro`:
  - Source: KOF Swiss Economic Institute
  - Timeframe: January 2004 - December 2022
  - Frequency: Monthly
  - Format: A tibble with monthly observations and 2 variables:
    - `time`: Date, the month and year of the observation.
    - `values`: Numeric, the value of the KOF barometer.
- `fcurve`:
  - Source: Burri and Kaufmann GitHub
  - Timeframe: January 2004 - December 2022
  - Frequency: Daily
  - Format: A tibble with daily observations and 2 variables:
    - `time`: Date, the date of the observation.
    - `values`: Numeric, the value of the F-curve (inverted for compatibility).
- `gdp`:
  - Source: SECO
  - Timeframe: January 2004 - December 2022
  - Frequency: Quarterly
  - Format: A tibble with quarterly observations and 2 variables:
    - `time`: Date, the quarter and year of the observation.
    - `values`: Numeric, the real GDP (seasonally adjusted).
- `wea`:
  - Source: SECO WEA Indicator
  - Timeframe: January 2005 - December 2022
  - Frequency: Weekly
  - Format: A tibble with weekly observations and 2 variables:
    - `time`: Date, the week and year of the observation.
    - `values`: Numeric, the value of the WEA.

Examples

```
# Load and plot `baro`
data(baro)
library(tsbox)
suppressMessages(ts_plot(baro))
```

summary.bridge *Summarize a* bridge *object*

Description

This method is summarizes the bridge model.

Usage

```
## S3 method for class 'bridge'
summary(object, ...)
```

Arguments

object	A bridge object obtained from bridge().
...	Additional arguments to be passed to the summary function. Ignored at the moment.

Value

The function summary is used to obtain and print a summary of the results.

Examples

```
library(bridgr)

# Example usage
target_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2022-12-01"), by = "quarter"),
  value = rnorm(12)
)))

indic_series <- suppressMessages(tsbox::ts_tbl(data.frame(
  time = seq(as.Date("2020-01-01"), as.Date("2023-01-01"), by = "month"),
  value = rnorm(37)
)))

bridge_model <- suppressMessages(bridge(
  target = target_series,
  indic = indic_series,
  indic_predict = "mean",
  indic_aggregators = "mean",
  indic_lags = 2,
  target_lags = 1,
  h = 1
))

# Summarize the information in the bridge model
summary(bridge_model)
```

Index