Package 'narfima'

September 21, 2025

Title Neural AutoRegressive Fractionally Integrated Moving Average Model

Version 0.1.0

Description Methods and tools for forecasting univariate time series using the NARFIMA (Neural AutoRegressive Fractionally Integrated Moving Average) model. It combines neural networks with fractional differencing to capture both nonlinear patterns and long-term dependencies. The NARFIMA model supports seasonal adjustment, Box-Cox transformations, optional exogenous variables, and the computation of prediction intervals. In addition to the NARFIMA model, this package provides alternative forecasting models including NARIMA (Neural ARIMA), NBSTS (Neural Bayesian Structural Time Series), and NNaive (Neural Naive) for performance comparison across different modeling approaches. The methods are based on algorithms introduced by Chakraborty et al. (2025) <doi:10.48550/arXiv.2509.06697>.

Maintainer Donia Besher <donia.a.besher@gmail.com>

Imports forecast, nnet, bsts, stats, utils, withr

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Tanujit Chakraborty [aut] (ORCID: <https://orcid.org/0000-0002-3479-2187>), Donia Besher [aut, cre, cph] (ORCID: <https://orcid.org/0009-0008-8314-1576>), Madhurima Panja [aut] (ORCID: <https://orcid.org/0009-0004-7467-2456>), Shovon Sengupta [aut] (ORCID: <https://orcid.org/0000-0003-2169-7364>)

Repository CRAN

Date/Publication 2025-09-21 14:00:02 UTC

Contents

`auto_narfima` *Fitting a NARFIMA Model*

Description

This function fits a Neural AutoRegressive Fractionally Integrated Moving Average (NARFIMA) model to univariate time series. The model uses `p` Autoregressive terms, `q` moving average terms, and a single hidden layer with `size` nodes. If `p`, `q`, or `size` are not specified, they are automatically determined:

- `p` is set to the number of significant Autoregressive coefficients estimated from an AR model.
- `q` is set to the order of the moving average component from an ARFIMA model fitted to the residuals.
- `size` is set to the floor of half the sum of `p` and `q`:

$$size = \lfloor \frac{p+q}{2} \rfloor.$$

Exogenous variables can be included via `xreg`. If `er` is not provided, it is computed from an ARFIMA model fitted to `y`. When `y` or `xreg` have missing values, the corresponding rows and any dependent lagged rows are removed before fitting the model. Multiple neural network models are fitted, each initialized with random weights, and the final model is obtained by averaging their outputs. Optionally, Box-Cox transformations can be applied to stabilize variance, and inputs can be scaled to improve model convergence.

Usage

```
auto_narfima(
  y,
  er,
  xreg = NULL,
  p,
  q,
  P = 1,
  size,
  skip,
  repeats = 1000,
  lambda = 0.5,
  lambdae = 0.5,
  scale.inputs = TRUE,
  ...
)
```

Arguments

y	A numeric vector or time series of class ts.
er	A numeric vector or time series object representing the series of residuals. If missing, it will be calculated from an ARFIMA model with exogenous variable (if provided).
xreg	An optional numeric matrix of exogenous variables to be included in the model (default is NULL).
p	Integer indicating the number of lags of the input series y.
q	Integer indicating the number of lags of the errors er.
P	Integer indicating the number of seasonal lags of the input series y (default is 1).
size	Integer specifying the number of nodes of the feed-forward neural networks with a single hidden layer.
skip	Logical value indicating whether to use the direct connections in the neural network (default is TRUE).
repeats	Integer specifying the number of times to fit the neural network model (default is 1000).
lambda	Numeric value for the Box-Cox transformation parameter of y (default is 0.5).
lambdae	Numeric value for the Box-Cox transformation parameter of er (default is 0.5).
scale.inputs	Logical value indicating whether to standardize the inputs before fitting the model (default is TRUE).
...	Additional arguments passed to auto_narfima.

Value

Returns an object of class "narfima", containing the following components:

series	The name of the input series.
method	A string describing the model parameters.
model	The fitted NARFIMA model.
fitted	The fitted values from the model.
residuals	The residuals from the model.
m	The seasonal frequency of the input series.
p	The number of Autoregressive terms used.
q	The number of moving average terms used.
P	The number of seasonal lags used.
size	The number of nodes in the hidden layer used in the neural network.
skip	Indicates if the direct connections were used in the neural network.
lambda	The Box-Cox transformation parameter for the input series.
scaley	Scaling parameters for the input series.
lags	The lags used for the input series.

lambdae	The Box-Cox transformation parameter for the error term.
scalee	Scaling parameters for the error term.
lagse	The lags used for the error term.
scalexreg	Scaling parameters for the exogenous variables, if provided.
y	The input time series.
e	The transformed residual series
xreg	The exogenous variables used in the model, if provided.
nnetargs	Additional arguments passed to the neural network function.

Examples

```
h <- 3

data <- EuStockMarkets[(nrow(EuStockMarkets) - 53):nrow(EuStockMarkets),4]

train <- data[1:(length(data) - h)]
test <- data[(length(data) - h + 1):length(data)]

narfima_model <- auto_narfima(train)
```

---

auto_narima *Fitting a NARIMA Model*

Description

This function fits a Neural AutoRegressive Integrated Moving Average (NARIMA) model to a univariate time series. It follows the same framework as narfima, but replaces the ARFIMA component with an ARIMA model. This function is intended for ablation studies to benchmark the performance of narfima against alternatives.

Usage

```
auto_narima(
  y,
  er,
  xreg = NULL,
  p,
  q,
  P = 1,
  size,
  skip,
  repeats = 1000,
  lambda = 0.5,
  lambdae = 0.5,
  scale.inputs = TRUE,
  ...
)
```

Arguments

y	A numeric vector or time series of class ts.
er	A numeric vector or time series object representing the series of residuals. If missing, it will be calculated from an ARIMA model.
xreg	An optional numeric matrix of exogenous variables to be included in the model (default is NULL).
p	Integer indicating the number of lags of the input series y.
q	Integer indicating the number of lags of the errors er.
P	Integer indicating the number of seasonal lags of the input series y (default is 1).
size	Integer specifying the number of nodes of the feed-forward neural networks with a single hidden layer.
skip	Logical value indicating whether to use the direct connections in the neural network (default is TRUE).
repeats	Integer specifying the number of times to fit the neural network model (default is 1000).
lambda	Numeric value for the Box-Cox transformation parameter of y (default is 0.5).
lambdae	Numeric value for the Box-Cox transformation parameter of er (default is 0.5).
scale.inputs	Logical value indicating whether to standardize the inputs before fitting the model (default is TRUE).
...	Additional arguments passed to auto_narima.

Value

Returns an object of class "narfima" with the same structure as narfima, but using ARIMA residuals instead of ARFIMA residuals.

---

auto_nbsts *Fitting a NBSTS Model*

Description

This function fits Neural Bayesian Structural Time Series (NBSTS) model to a univariate time series. It follows the same architecture as narfima, but replaces the ARFIMA component with a BSTS framework. By default, the state specification includes a semilocal linear trend component and a seasonal component with 12 periods. The model is estimated using MCMC sampling with 1000 iterations by default. This function is intended for ablation studies to benchmark the performance of narfima against alternatives.

Usage

```
auto_nbsts(
  y,
  er,
  ss,
  xreg = NULL,
  p,
  q,
  P = 1,
  size,
  skip,
  niter = 1000,
  repeats = 1000,
  lambda = 0.5,
  lambdae = 0.5,
  scale.inputs = TRUE,
  ...
)
```

Arguments

y	A numeric vector or time series of class ts.
er	A numeric vector or time series object representing the series of residuals. If missing, it will be calculated from a BSTS model.
ss	A BSTS state specification list. By default, a semilocal linear trend and seasonal components are added using y.
xreg	An optional numeric matrix of exogenous variables to be included in the model (default is NULL).
p	Integer indicating the number of lags of the input series y.
q	Integer indicating the number of lags of the errors er.
P	Integer indicating the number of seasonal lags of the input series y (default is 1).
size	Integer specifying the number of nodes of the feed-forward neural networks with a single hidden layer.
skip	Logical value indicating whether to use the direct connections in the neural network (default is TRUE).
niter	Integer specifying the number of MCMC iterations for fitting the BSTS model (default is 1000).
repeats	Integer specifying the number of times to fit the neural network model (default is 1000).
lambda	Numeric value for the Box-Cox transformation parameter of y (default is 0.5).
lambdae	Numeric value for the Box-Cox transformation parameter of er (default is 0.5).
scale.inputs	Logical value indicating whether to standardize the inputs before fitting the model (default is TRUE).
...	Additional arguments passed to auto_nbsts.

Value

Returns an object of class "narfima" with the same structure as `narfima`, but using BSTS residuals instead of ARFIMA residuals.

`auto_nnaive` *Fitting a NNaive Model*

Description

This function fits Neural Naive (NNaive) model to a univariate time series. It follows the same framework as `narfima`, but replaces the ARFIMA component with a Naive model. This function is intended for ablation studies to benchmark the performance of `narfima` against alternatives.

Usage

```
auto_nnaive(
  y,
  er,
  xreg = NULL,
  p,
  q,
  P = 1,
  size,
  skip,
  repeats = 1000,
  lambda = 0.5,
  lambdae = 0.5,
  scale.inputs = TRUE,
  ...
)
```

Arguments

`y`	A numeric vector or time series of class `ts`.
`er`	A numeric vector or time series object representing the series of residuals. If missing, it will be calculated from a Naive model.
`xreg`	An optional numeric matrix of exogenous variables to be included in the model (default is `NULL`).
`p`	Integer indicating the number of lags of the input series `y`.
`q`	Integer indicating the number of lags of the errors `er`.
`P`	Integer indicating the number of seasonal lags of the input series `y` (default is `1`).
`size`	Integer specifying the number of nodes of the feed-forward neural networks with a single hidden layer.
`skip`	Logical value indicating whether to use the direct connections in the neural network (default is `TRUE`).

repeats	Integer specifying the number of times to fit the neural network model (default is 1000).
lambda	Numeric value for the Box-Cox transformation parameter of y (default is 0.5).
lambdae	Numeric value for the Box-Cox transformation parameter of er (default is 0.5).
scale.inputs	Logical value indicating whether to standardize the inputs before fitting the model (default is TRUE).
...	Additional arguments passed to auto_nnaive.

Value

Returns an object of class "narfima" with the same structure as narfima, but using Naive residuals instead of ARFIMA residuals.

---

forecast_narfima_class

Forecasting from NARFIMA-class Models

Description

This function produces forecasts from fitted models of class narfima, including NARFIMA, NARIMA, NBSTS, and NNaive. Multi-step forecasts are computed recursively. Prediction intervals can also be obtained through simulation.

Usage

```
forecast_narfima_class(
  object,
  h = ifelse(object$m > 1, 2 * object$m, 10),
  PI = FALSE,
  level = 80,
  fan = FALSE,
  bootstrap = FALSE,
  npaths = 1000,
  innov = NULL,
  xreg = NULL,
  lambda = object$lambda,
  lambdae = object$lambdae,
  ...
)
```

Arguments

object	An object of class "narfima".
h	Number of periods to forecast. If xreg is provided, h is ignored and the forecast horizon is determined by the number of rows in xreg.

PI	Logical value indicating whether to compute prediction intervals (default is FALSE).
level	Confidence level for prediction intervals (default is 80), ignored when PI = FALSE.
fan	If TRUE (default is FALSE), level is set to seq(51, 99, by = 3). This is suitable for fan plots.
bootstrap	Logical value indicating whether to use bootstrapping for prediction intervals (default is FALSE), ignored when PI = FALSE.
npaths	Number of simulation paths for prediction intervals (default is 1000), ignored when PI = FALSE.
innov	An optional vector of innovations to use for simulating future values (default is NULL). If NULL and PI = TRUE, random innovations are generated.
xreg	An optional matrix or data frame of exogenous variables to be used in forecasting (default is NULL).
lambda	Numeric value for the Box-Cox transformation parameter of y (default is taken from the fitted model, i.e., object$lambda).
lambdae	Numeric value for the Box-Cox transformation parameter of er (default is taken from the fitted model, i.e., object$lambdae).
...	Additional arguments passed to pred_interval_narfima.

Value

An object of class "forecast" containing:

mean	The point forecasts.
lower	The lower bounds of the prediction intervals.
upper	The upper bounds of the prediction intervals.
level	The confidence level for the prediction intervals.
model	The fitted NARFIMA model.
series	The name of the input series.
method	A string describing the model parameters.
y	The input time series.
e	The transformed residual series

Examples

```
h <- 3

data <- EuStockMarkets[(nrow(EuStockMarkets) - 83):nrow(EuStockMarkets), 3]

train <- data[1:(length(data) - h)]
test <- data[(length(data) - h + 1):length(data)]

narfima_model <- auto_narfima(train)
narfima_forecast <- forecast_narfima_class(narfima_model, h)
```

Index