

Package ‘EXPAR’

July 21, 2025

Type Package

Title Fitting of Exponential Autoregressive (EXPAR) Model

Version 0.1.0

Description

The amplitude-dependent exponential autoregressive (EXPAR) time series model, initially proposed by Haggan and Ozaki (1981) <[doi:10.2307/2335819](https://doi.org/10.2307/2335819)> has been implemented in this package. Throughout various studies, the model has been found to adequately capture the cyclical nature of datasets. Parameter estimation of such family of models has been tackled by the approach of minimizing the residual sum of squares (RSS). Model selection among various candidate orders has been implemented using various information criteria, viz., Akaike information criteria (AIC), corrected Akaike information criteria (AICc) and Bayesian information criteria (BIC). An illustration utilizing data of egg price indices has also been provided.

License GPL-3

Imports forecast, stats

Encoding UTF-8

LazyData true

NeedsCompilation no

Author Saikath Das [aut, cre],
Bishal Gurung [aut],
Achal Lama [aut],
KN Singh [aut]

Maintainer Saikath Das <saikathdas007@gmail.com>

Repository CRAN

Date/Publication 2024-05-10 13:50:11 UTC

Contents

best_EXPAR	2
egg_price_index	4
fit_EXPAR	4
forecast_EXPAR	5
inital_val	6
optimise_EXPAR	7

best_EXPAR	<i>Fitting of EXPAR model</i>
------------	-------------------------------

Description

Searches for the best EXPAR model among many orders (defaults upto 5), compares them using information criterion and returns the best fit.

Usage

```
best_EXPAR(ts_data, max.p = 5, ic = "AIC", opt_method = "BFGS")
```

Arguments

ts_data	A univariate time series data, to which an EXPAR model is to be fitted.
max.p	The maximum order upto which models are to be searched for comparison.
ic	Information criterion to be used for model selection: Akaike information criterion ("AIC"), corrected Akaike information criterion ("AIC_c") or Bayesian information criterion ("BIC").
opt_method	The optimization algorithm to be used for RSS minimization. Corresponds to arguments from <code>optim()</code> in stats. Defaults to the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

Details

Fits `max.p` number of EXPAR models to the given dataset by minimisation of RSS using `optimise_EXPAR()` and returns the best model among the evaluated ones. Model selection is based on the information criteria given in `ic`.

The various information criterion are calculated (estimated) from RSS as,

$$AIC = 2k + n \log\left(\frac{RSS}{n}\right)$$

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$$

$$BIC = k \log(n) + n \log\left(\frac{RSS}{n}\right)$$

where, n, k are the number of observations and the number of parameters, respectively.

Value

Returns the fitted EXPAR model as a list with the following components,

series	The data used for fitting the model.
order	Order p of the fitted EXPAR model.
n	Number of observations in series.
k	Number of parameters in the model.
par	Parameters of the fitted model.
Fitted	Fitted values obtained from the model.
Residuals	Residuals of the fitted model.
RSS	The residual sum of squares.
AIC	Akaike information criterion, evaluated from RSS.
AIC_c	Corrected Akaike information criterion, evaluated from RSS.
BIC	Bayesian information criterion, evaluated from RSS.
counts	counts returned by <code>optim()</code>
convergence	convergence returned by <code>optim()</code>
message	message returned by <code>optim()</code>

Author(s)

Saikath Das, Bishal Gurung, Achal Lama and KN Singh

References

Haggan and Ozaki (1981). Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model. *Biometrika*, 68(1):189-199. <doi:10.1093/biomet/68.1.189>.

Gurung (2015). An exponential autoregressive (EXPAR) model for the forecasting of all India annual rainfall. *Mausam*, 66(4):847-849. <doi:10.54302/mausam.v66i4.594>.

Examples

```
datats <- ts(egg_price_index[,3], start = c(2013, 1), frequency = 12)
best_EXPAR(datats)
```

egg_price_index	<i>Price Index of Eggs in Urban Ares of India</i>
-----------------	---

Description

This data set contains observations of Consumer Price Index (CPI) of eggs for urban areas in India. It consists of 86 monthly observations, ranging from January, 2013 till February, 2020. Some observations after this point of time were also available, but were discarded due to their questionable reliability and missingness pertaining to the onset of COVID-19 pandemic from March, 2020.

Usage

```
egg_price_index
```

Format

A data frame containing 86 observations. Observations are contained in the "egg_price_index" column, while the first two columns pertain to year and month of observation, respectively.

fit_EXPAR	<i>Fitting of EXPAR model with known parameters</i>
-----------	---

Description

Fits an EXPAR model to the dataset for given parameter values, without any optimisation.

Usage

```
fit_EXPAR(ts_data, par)
```

Arguments

ts_data	A univariate data to which the model is to be fitted, preferably (but not limited to) a ts object.
par	The parameter vector. Any value from the parametric space of EXPAR model is permissible. Number of required parameters is $2p + 1$, where p is order of the model. Entries are arranged consecutively, for example, for order 2, the form of input is <code>c(Phi1,Phi2,Pi1,Pi2,Gamma)</code> .

Details

This function uses the form of EXPAR model to generate consecutive fitted observations as theoretical values obtained from the model. The form of EXPAR model employed is given by,

$$y_{t+1} = \sum_{i=1}^p [(\phi_i + \pi_i e^{-\gamma y_t^2}) y_{t-i+1}] + \epsilon_{t+1}$$

where, $\gamma > 0$ and $\{\epsilon_t\}$ is white noise process with zero mean and constant variance σ_e^2

The process of fitting involves using actual observations in the dataset in the RHS of above equation and obtaining the fitted values y_{t+1} . No optimization is done as parameter estimation is omitted. It is useful for simulation of data with given order and parameters.

Value

Returns the fitted EXPAR model as a list with the following components,

Fitted	Fitted values obtained from LHS of the aforementioned model. Due to obvious reasons, fits of the first p observations are not obtained.
Residuals	The residuals ϵ_{t+1} , computed by iterative fitting.
RSS	The residual sum of squares.
AIC	Akaike information criterion, evaluated from RSS.
AIC_c	Corrected Akaike information criterion, evaluated from RSS.
BIC	Bayesian information criterion, evaluated from RSS.

Examples

```
datats <- ts(egg_price_index[,3], start = c(2013, 1), frequency = 12)
fit_EXPAR(datats, par = c(0.45, 0.68, 0.48))
```

forecast_EXPAR	<i>Forecasting from EXPAR model</i>
----------------	-------------------------------------

Description

Generates out-of-sample forecasts from any fitted EXPAR model.

Usage

```
forecast_EXPAR(EXPAR_model, h = 1)
```

Arguments

EXPAR_model	A fitted EXPAR model.
h	Number of periods for forecasting. Defaults to one-step ahead forecast.

Details

Forecasts from EXPAR model are generated iteratively using mathematical form of EXPAR model, considering fitted values and residuals while putting future error terms as zero.

Value

Vector of forecasted values

Examples

```
datats <- ts(egg_price_index[,3], start = c(2013, 1), frequency = 12)
best_model = best_EXPAR(datats)
forecast_EXPAR(best_model, h = 5)
```

initial_val	<i>Initial values for fitting EXPAR model</i>
-------------	---

Description

Determines the initial values to be used during parameter estimation.

Usage

```
initial_val(ts_data, order)
```

Arguments

ts_data	A univariate time series data, to which an EXPAR model is to be fitted.
order	Order p of the EXPAR model to be considered.

Details

This function determines the initial values required for optimising the parameters of EXPAR model. The initial values are extracted from an $ARIMA(p, d, 0)$ model with the same order and appropriate differencing. These will act as default initial values in the `optimise_EXPAR()` and `best_EXPAR()` functions, unless specified by user. Without these values, the optimisation process may fail.

Value

It returns the initial values of the EXPAR model over which optimisation is to be done parameter estimation.

Examples

```
datats <- ts(egg_price_index[,3], start = c(2013, 1), frequency = 12)
initial_val(datats, 2)
```

optimise_EXPAR	<i>Fitting of EXPAR model with given order</i>
----------------	--

Description

Fits an EXPAR model of given order to the data by RSS minimisation

Usage

```
optimise_EXPAR(ts_data, order, init, opt_method = "BFGS")
```

Arguments

ts_data	A univariate time series data, to which an EXPAR model of given order is to be fitted.
order	Order p of the EXPAR model to be considered.
init	The initial values for optimisation. If omitted, initial values are decided using <code>initial_val()</code> .
opt_method	The optimization algorithm to be used for RSS minimization. Corresponds to arguments from <code>optim()</code> in stats library. Defaults to the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

Details

This function estimates the parameters an $\text{EXPAR}(p)$ model by minimizing RSS and returns a fit with the optimised parameters. The fit is returned using the function `fit_EXPAR()`, with the parameters inputted being the optimized ones. Available optimisation techniques include possible arguments from `method` in `optim()`.

Value

Returns the fitted EXPAR model as a list with the following components,

series	The data used for fitting the model.
order	Order p of the fitted EXPAR model.
n	Number of observations in series.
k	Number of parameters in the model.
par	Parameters of the fitted model.
Fitted	Fitted values obtained from the model.
Residuals	Residuals of the fitted model.
RSS	The residual sum of squares.
AIC	Akaike information criterion, evaluated from RSS.
AIC_c	Corrected Akaike information criterion, evaluated from RSS.
BIC	Bayesian information criterion, evaluated from RSS.

counts	counts returned by optim()
convergence	convergence returned by optim()
message	message returned by optim()

Examples

```
datats <- ts(egg_price_index[,3], start = c(2013, 1), frequency = 12)
optimise_EXPAR(datats, 2, opt_method = "BFGS")
```


Index

* **datasets**

egg_price_index, [4](#)

best_EXPAR, [2](#)

egg_price_index, [4](#)

fit_EXPAR, [4](#)

forecast_EXPAR, [5](#)

inital_val, [6](#)

optimise_EXPAR, [7](#)