Package 'mtarm'

July 22, 2024

Type Package

Title Bayesian Estimation of Multivariate Threshold Autoregressive Models

Version 0.1.2

Author Luis Hernando Vanegas [aut, cre],
Sergio Alejandro Calderón [aut],
Luz Marina Rondón [aut]

Maintainer Luis Hernando Vanegas <lhvanegasp@unal.edu.co>

Description Estimation, inference and forecasting using the Bayesian approach for multivariate threshold autoregressive (TAR) models in which the distribution used to describe the noise process belongs to the class of Gaussian variance mixtures.

Imports methods, stats, utils, graphics, Formula, grDevices, GIGrvg, coda

BugReports https://github.com/lhvanegasp/mtar/issues

Encoding UTF-8

LazyData false

RoxygenNote 7.2.3

NeedsCompilation no

License GPL-2 | GPL-3

Repository CRAN

Date/Publication 2024-07-22 18:20:02 UTC

Contents

`convert` *Converts chains from the Bayesian estimation of a multivariate TAR model to a mcmc object.*

Description

This function converts the chains obtained from the Bayesian estimation of a multivariate TAR model to a `mcmc` object to be analyzed with the **coda** package.

Usage

```
convert(object, type = c("location", "scale", "extra"), regime = 1)
```

Arguments

`object`	an object of the class *mtar*.
`type`	an (optional) character string that allows the user to specify the parameter that corresponds to the chains to convert. The available options are: "location", "scale" and "extra". As default, `type` is set to "location".
`regime`	an (optional) integer value that allows the user to specify the regime that corresponds to the chains to plot. As default, `regime` is set to 1.

Value

a `mcmc`-type object.

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, row.names=Date, dist="Slash",
             data=returns, ars=list(p=c(1,1,2)), n.burnin=100, n.sim=3000)
location.chains.1 <- convert(fit1,type="location",regime=2)
summary(location.chains.1)
plot(location.chains.1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, row.names=Date, dist="Laplace",
             data=riverflows, ars=list(p=c(5,5,5)), n.burnin=100, n.sim=3000)
location.chains.2 <- convert(fit2,type="location",regime=3)
summary(location.chains.2)
plot(location.chains.2)
```

DIC *Deviance information criterion (DIC)*

Description

This function computes the Deviance information criterion (DIC) for objects of class `mtar`.

Usage

```
DIC(..., verbose = TRUE, digits = max(3, getOption("digits") - 2))
```

Arguments

`...`	one or several objects of the class *mtar*.
`verbose`	an (optional) logical switch indicating if should the report of results be printed. As default, `verbose` is set to TRUE.
`digits`	an (optional) integer indicating the number of digits to print. As default, `digits` is set to `max(3, getOption("digits") - 2)`.

Value

A `data.frame` with the values of the DIC for each *mtar* object in the input.

References

Spiegelhalter D.J., Best N.G., Carlin B.P. and Van Der Linde A. (2002) Bayesian Measures of Model Complexity and Fit. Journal of the Royal Statistical Society Series B (Statistical Methodology), 64(4), 583–639.

Spiegelhalter D.J., Best N.G., Carlin B.P. and Van der Linde A. (2014). The deviance information criterion: 12 years on. Journal of the Royal Statistical Society Series B (Statistical Methodology), 76(3), 485–493.

See Also

WAIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1a <- mtar(~ COLCAP + BOVESPA | SP500, row.names=Date, dist="Gaussian",
              data=returns, ars=list(p=c(1,1,2)), n.burnin=100, n.sim=3000)
fit1b <- update(fit1a,dist="Slash")
fit1c <- update(fit1a,dist="Student-t")
DIC(fit1a,fit1b,fit1c)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
```

```
fit2a <- mtar(~ Bedon + LaPlata | Rainfall, row.names=Date, dist="Gaussian",
              data=riverflows, ars=list(p=c(5,5,5)), n.burnin=100, n.sim=3000)
fit2b <- update(fit2a,dist="Slash")
fit2c <- update(fit2a,dist="Student-t")
DIC(fit2a,fit2b,fit2c)
```

`forecasting` *Forecasting of a multivariate TAR model.*

Description

This function computes forecasting from a fitted multivariate TAR model.

Usage

```
forecasting(object, data, credible = 0.95, row.names)
```

Arguments

`object`	an object of the class *mtar*.
`data`	an (optional) data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the future values of the threshold series as well as the exogenous series in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `mtar` is called.
`credible`	an (optional) value for the level of the credible intervals. As default, `credible` is set to 0.95.
`row.names`	an (optional) vector that allows the user to name the time point to which each row in the data set `data` corresponds.

Value

a list with the following component

`ypred`	a matrix with the results of the forecasting,
`summary`	a matrix with the mean, standard deviation, and the HDP credible intervals of the forecasting,

References

Nieto, F.H. (2005) Modeling Bivariate Threshold Autoregressive Processes in the Presence of Missing Data. Communications in Statistics - Theory and Methods, 34, 905-930.

Romero, L.V. and Calderon, S.A. (2021) Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. Communications in Statistics - Theory and Methods, 50, 2508-2530.

Calderon, S.A. and Nieto, F.H. (2017) Bayesian analysis of multivariate threshold autoregressive models with missing data. Communications in Statistics - Theory and Methods, 46, 296-318.

Karlsson, S. (2013) Chapter 15-Forecasting with Bayesian Vector Autoregression. In Elliott, G. and Timmermann, A. Handbook of Economic Forecasting, Volume 2, 791–89, Elsevier.

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, row.names=Date, dist="Slash",
             data=subset(returns,Date < "2016-03-20"), ars=list(p=c(1,1,2)),
             n.burnin=100, n.sim=3000)
out1 <- forecasting(fit1,data=subset(returns,Date >= "2016-03-20"),row.names=Date)
out1$summary

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, row.names=Date, dist="Laplace",
             data=subset(riverflows,Date < "2009-04-09"), ars=list(p=c(5,5,5)),
             n.burnin=100, n.sim=3000)
out2 <- forecasting(fit2,data=subset(riverflows,Date >= "2009-04-09"),row.names=Date)
out2$summary
```

`mtar` *Bayesian estimation of a multivariate threshold autoregressive (TAR) model.*

Description

This function uses Gibbs sampling to generate a sample from the posterior distribution of the parameters of a multivariate TAR model when the noise process follows Gaussian, Student-t, Slash, Symmetric Hyperbolic, Contaminated normal, or Laplace distribution.

Usage

```
mtar(
  formula,
  data,
  subset,
  Intercept = TRUE,
  ars,
  row.names,
  dist = "Gaussian",
  prior = list(),
  n.sim = 500,
  n.burnin = 100,
```

```
  n.thin = 1,
  log = FALSE,
  ...
)
```

Arguments

formula	a three-part expression of type Formula describing the TAR model to be fitted to the data. In the first part, the variables in the multivariate output series are listed; in the second part, the threshold series is specified, and in the third part, the variables in the multivariate exogenous series are specified.
data	an (optional) data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which mtar is called.
subset	an (optional) vector specifying a subset of observations to be used in the fitting process.
Intercept	an (optional) logical variable. If TRUE, then the model includes an intercept.
ars	a list composed of three objects, namely: p, q and d, each of which corresponds to a vector of non-negative integers with as many elements as there are regimes in the TAR model.
row.names	an (optional) vector that allows the user to name the time point to which each row in the data set corresponds.
dist	an (optional) character string that allows the user to specify the multivariate distribution to be used to describe the behavior of the noise process. The available options are: Gaussian ("Gaussian"), Student-t ("Student-t"), Slash ("Slash"), Symmetric Hyperbolic ("Hyperbolic"), Laplace ("Laplace"), and contaminated normal ("Contaminated normal"). As default, dist is set to "Gaussian".
prior	an (optional) list that allows the user to specify the values of the hyperparameters, that is, allows to specify the values of the parameters of the prior distributions.
n.sim	an (optional) positive integer specifying the required number of iterations for the simulation after the burn-in period. As default, n.sim is set to 500.
n.burnin	an (optional) positive integer specifying the required number of burn-in iterations for the simulation. As default, n.burnin is set to 100.
n.thin	an (optional) positive integer specifying the required thinning interval for the simulation. As default, n.thin is set to 1.
log	an (optional) logical variable. If TRUE, then the behaviour of the output series is described using the exponentiated version of dist.
...	further arguments passed to or from other methods.

Value

an object of class *mtar* in which the main results of the model fitted to the data are stored, i.e., a list with components including

chains	list with several arrays, which store the values of each model parameter in each iteration of the simulat
n.sim	number of iterations of the simulation after the burn-in period,
n.burnin	number of burn-in iterations in the simulation,
n.thin	thinning interval in the simulation,
regim	number of regimes,
ars	list composed of three objects, namely: p, q and d, each of which corresponds to a vector of non-negat
dist	name of the multivariate distribution used to describe the behavior of the noise process,
threshold.series	vector with the values of the threshold series,
response.series	matrix with the values of the output series,
covariable.series	matrix with the values of the exogenous series,
Intercept	If TRUE, then the model included an intercept term,
formula	the formula,
call	the original function call.

References

Nieto, F.H. (2005) Modeling Bivariate Threshold Autoregressive Processes in the Presence of Missing Data. Communications in Statistics - Theory and Methods, 34, 905-930.

Romero, L.V. and Calderon, S.A. (2021) Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. Communications in Statistics - Theory and Methods, 50, 2508-2530.

Calderon, S.A. and Nieto, F.H. (2017) Bayesian analysis of multivariate threshold autoregressive models with missing data. Communications in Statistics - Theory and Methods, 46, 296-318.

See Also

DIC, WAIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, row.names=Date, dist="Slash",
             data=returns, ars=list(p=c(1,1,2)), n.burnin=100, n.sim=3000)
summary(fit1)
```

```
###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, row.names=Date, dist="Laplace",
             data=riverflows, ars=list(p=c(5,5,5)), n.burnin=100, n.sim=3000)
summary(fit2)
```

`returns` *Returns of the closing prices of three financial indexes*

Description

These data correspond to the returns of closing prices of the Colcap, Bovespa, and S&P 500 indexes from 2010-02-01 to 2016-03-31 (1505 time points). Colcap is a leading indicator of the price dynamics of the 20 most liquid shares on the Colombian Stock Market. Bovespa is the Brazilian stock market index, the world's thirteenth largest and most important stock exchange, and the first in Latin America. Finally, the Standard & Poor's 500 (S&P 500) index is a stock index based on the 500 largest companies in the United States.

Usage

```
data(returns)
```

Format

A data frame with 1505 rows and 4 variables:

Date a vector indicating the dates of the measurements.

COLCAP a numeric vector indicating the returns of COLCAP.

SP500 a numeric vector indicating the returns of SP500.

BOVESPA a numeric vector indicating the returns of BOVESPA.

References

Romero, L.V. and Calderon, S.A. (2021) Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. Communications in Statistics - Theory and Methods, 50, 2508-2530.

Examples

```
data(returns)
dev.new()
plot(ts(as.matrix(returns[,-1])), main="Returns")
```

`riverflows` *Rainfall and two river flows in Colombia*

Description

The data represent daily rainfall (in mm) and two river flows (in m^3/s) in southern Colombia. A meteorological station located with an altitude of 2400 meters was used to measure rainfall. In the first case, the El Trebol hydrological station was used to measure the flow in the Bedon river at an altitude of 1720 meters. In the second case, the Villalosada hydrological station measured the flow in the La Plata river at an altitude of 1300 meters. Geographically, the stations are located near the equator. The last characteristic allows for control over hydrological and meteorological factors that might distort a dynamic relationship. January 1, 2006, to April 14, 2009, is the sample period.

Usage

```
data(riverflows)
```

Format

A data frame with 1200 rows and 4 variables:

Date a vector indicating the dates of the measurements.

Bedon a numeric vector indicating the Bedon river flow.

LaPlata a numeric vector indicating the La Plata river flow.

Rainfall a numeric vector indicating the rainfall.

References

Calderon, S.A. and Nieto, F.H. (2017) Bayesian analysis of multivariate threshold autoregressive models with missing data. Communications in Statistics - Theory and Methods, 46, 296-318.

Examples

```
data(riverflows)
dev.new()
plot(ts(as.matrix(riverflows[,-1])), main="Rainfall and river flows")
```

`simtar` *Simulation of multivariate time series according to a TAR model*

Description

This function simulates multivariate time series according to a user-specified TAR model.

Usage

```
simtar(
  n,
  k = 2,
  ars = list(p = 1),
  Intercept = TRUE,
  parms,
  delay = 0,
  thresholds = 0,
  t.series,
  ex.series,
  dist = "gaussian",
  extra
)
```

Arguments

`n`	a positive integer value indicating the length of the desired output series.
`k`	a positive integer value indicating the dimension of the desired output series.
`ars`	a list composed of three objects, namely: `p`, `q` and `d`, each of which corresponds to a vector of l non-negative integers, where l represents the number of regimes in the TAR model.
`Intercept`	an (optional) logical variable. If `TRUE`, then the model includes an intercept.
`parms`	a list with as many sublists as regimes in the user-specified TAR model. Each sublist is composed of two matrices. The first corresponds to location parameters, while the second corresponds to scale parameters.
`delay`	an (optional) non-negative integer value indicating the delay in the threshold series.
`thresholds`	a vector with $l - 1$ real values sorted ascendingly.
`t.series`	a matrix with the values of the threshold series.
`ex.series`	a matrix with the values of the multivariate exogenous series.
`dist`	an (optional) character string which allows the user to specify the multivariate distribution to be used to describe the behavior of the noise process. The available options are: Gaussian ("Gaussian"), Student-t ("Student-t"), Slash ("Slash"), Symmetric Hyperbolic ("Hyperbolic"), Laplace ("Laplace"), and contaminated normal ("Contaminated normal"). As default, `dist` is set to "Gaussian".

extra — a value indicating the value of the extra parameter of the noise process distribution, if any.

Value

a data.frame containing the output series, threshold series (if any), and multivariate exogenous series (if any).

Examples

```
###### Simulation of a trivariate TAR model with two regimes
n <- 2000
k <- 3
ars <- list(p=c(1,2))
Z <- as.matrix(arima.sim(n=n+max(ars$p),list(ar=c(0.5))))
Intercept <- TRUE
parms <- list()
for(i in 1:length(ars$p)){
   np <- Intercept + ars$p[i]*k
   parms[[i]] <- list()
  parms[[i]]$location <- c(ifelse(runif(np*k)<=0.5,1,-1)*rbeta(np*k,shape1=4,shape2=16))
   parms[[i]]$location <- matrix(parms[[i]]$location,np,k)
   parms[[i]]$scale <- rgamma(k,shape=1,scale=1)*diag(k)
}
thresholds <- quantile(Z,probs=seq(1,length(ars$p)-1)/length(ars$p))
out1 <- simtar(n=n,k=k,ars=ars,Intercept=Intercept,parms=parms,
               thresholds=thresholds,t.series=Z,dist="Student-t",extra=6)
str(out1)

###### Simulation of a trivariate VAR model
n <- 2000
k <- 3
ars <- list(p=2)
Intercept <- TRUE
parms <- list()
for(i in 1:length(ars$p)){
   np <- Intercept + ars$p[i]*k
   parms[[i]] <- list()
  parms[[i]]$location <- c(ifelse(runif(np*k)<=0.5,1,-1)*rbeta(np*k,shape1=4,shape2=16))
   parms[[i]]$location <- matrix(parms[[i]]$location,np,k)
   parms[[i]]$scale <- rgamma(k,shape=1,scale=1)*diag(k)
}
out2 <- simtar(n=n,k=k,ars=ars,Intercept=Intercept,parms=parms,
               dist="Slash",extra=2)
str(out2)
```

WAIC *Watanabe-Akaike or Widely Available Information Criterion (WAIC)*

Description

This function computes the Watanabe-Akaike or Widely Available Information criterion (WAIC) for objects of class `mtar`.

Usage

```
WAIC(..., verbose = TRUE, digits = max(3, getOption("digits") - 2))
```

Arguments

`...`	one or several objects of the class *mtar*.
`verbose`	an (optional) logical switch indicating if should the report of results be printed. As default, `verbose` is set to TRUE.
`digits`	an (optional) integer indicating the number of digits to print. As default, `digits` is set to `max(3, getOption("digits") - 2)`.

Value

A `data.frame` with the values of the WAIC for each *mtar* object in the input.

References

Watanabe S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. The Journal of Machine Learning Research, 11, 3571–3594.

See Also

DIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1a <- mtar(~ COLCAP + BOVESPA | SP500, row.names=Date, dist="Gaussian",
              data=returns, ars=list(p=c(1,1,2)), n.burnin=100, n.sim=3000)
fit1b <- update(fit1a,dist="Slash")
fit1c <- update(fit1a,dist="Student-t")
WAIC(fit1a,fit1b,fit1c)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2a <- mtar(~ Bedon + LaPlata | Rainfall, row.names=Date, dist="Gaussian",
              data=riverflows, ars=list(p=c(5,5,5)), n.burnin=100, n.sim=3000)
fit2b <- update(fit2a,dist="Slash")
fit2c <- update(fit2a,dist="Student-t")
WAIC(fit2a,fit2b,fit2c)
```

Index