Package ‘bayesDccGarch’

October 5, 2021

Type Package

Title Methods and Tools for Bayesian Dynamic Conditional Correlation GARCH(1,1) Model

Version 3.0.3

Author Jose Augusto Fiorucci [aut, cre, cph]
  (<https://orcid.org/0000-0002-1201-9089>),
  Ricardo Sanders Ehlers [aut, cph]
  (<https://orcid.org/0000-0001-9034-5173>),
  Francisco Louzada [aut, cph] (<https://orcid.org/0000-0001-7815-9554>)

Maintainer Jose Augusto Fiorucci <jafiorucci@gmail.com>


License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 2.0), numDeriv, coda

BugReports https://github.com/jafiorucci/bayesDccGarch/issues


NeedsCompilation yes

RoxygenNote 7.1.2

Repository CRAN

Date/Publication 2021-10-05 14:20:02 UTC

R topics documented:

bayesDccGarch-package .................................................. 2
bayesDccGarch .......................................................... 3
DaxCacNik ............................................................... 6
densityFunctions ......................................................... 7
bayesDccGarch-package

bayesDccGARCH: Methods and tools for Bayesian analysis of DCC-GARCH(1,1) Model.

Description

In this package we implemented functions for Bayesian analysis of DCC-GARCH(1,1) Model using the same modelling of Fioruci et al (2014a). Several probabilities distributions are available for the errors which can model both skewness and heavy tails. See Fioruci et al (2014b) for more details about the package.

Details

Package: bayesDccGarch
Type: Package
Version: 3.0.3
Date: 2021-10-02
License: GPL (>=2.14)

bayesDccGarch(mY,n_sim = 10000)

Author(s)

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada. Maintainer: Jose Augusto Fiorucci <jafiorucci@gmail.com>

References


See Also

Available functions: bayesDccGarch, update, predict, plot, logLikDccGarch, dssnorm, dsst, dssged, plotVol
Examples

```r
data(DaxCacNik)
out = bayesDccGarch(DaxCacNik)
summary(out)
plot(out)
```

bayesDccGarch  

Bayesian Estimation of the DCC-GARCH(1,1) Model.

Description

Performs a Markov Chain for all parameters of the DCC-GARCH(1,1) Model.

Usage

```r
bayesDccGarch(mY, nSim = 10000, tail_ini = 8, omega_ini = 0.1*diag(var(mY)),
               alpha_ini = rep(0.05, ncol(mY)), beta_ini = rep(0.85, ncol(mY)),
               a_ini = 0.04, b_ini = 0.8, gamma_ini = rep(1, ncol(mY)),
               errorDist = 2, control = list())
```

```r
increaseSim(x, nSim = 10000)
```

## S3 method for class 'bayesDccGarch'
update(object, ..., mY_new)

## S3 method for class 'bayesDccGarch'
window(x, start = NULL, end = NULL, thin = NULL, ...)

Arguments

- **mY**: a matrix of the data \((n \times k)\).
- **nSim**: length of Markov chain. Default: 10000.
- **tail_ini**: initial value of \(\nu\) parameter if errorDist = 2 or initial value of \(\delta\) parameter if errorDist = 3. If errorDist = 1 this arguments is not used.
- **omega_ini**: a numeric vector \((k \times 1)\) with the initial values of \(\omega_i\) parameters. Default: rep(0.03, ncol(mY)).
- **alpha_ini**: a numeric vector \((k \times 1)\) with the initial values of \(\alpha_i\) parameters. Default: rep(0.03, ncol(mY)).
**bayesDccGarch**

- **beta_ini**: a numeric vector \((k \times 1)\) with the initial values of \(\beta_i\) parameters. Default: `rep(0.8, ncol(mY))`.
- **a_ini**: a numeric value of the initial values of \(a\) parameter. Default: 0.03.
- **b_ini**: a numeric value of the initial values of \(b\) parameter. Default: 0.8.
- **gamma_ini**: a numeric vector \((k \times 1)\) with the initial values of \(\gamma_i\) parameters. Default: `rep(1.0, ncol(mY))`.
- **errorDist**: a probability distribution for errors. Use `errorDist=1` for `SSNorm`, `errorDist=2` for `SST` or `errorDist=3` for `SSGED`. Default: 2.
- **control**: list of control arguments (See *Details*).
- **x, object**: an object of `bayesDccGarch` class.
- **mY_new**: a matrix of new data \((n_{new} \times k)\).
- **start**: the first iteration of interest from Markov chain.
- **end**: the last iteration of interest from Markov chain.
- **thin**: the required interval between successive samples.
- **...**: additional arguments for S3 generic `window` function

### Details

The `bayesDccGarch()` function performs a Markov Chain for all parameters of the model DCC-GARCH(1,1) (or GARCH(1,1) in the univariate case). There are three options of probability distributions for the error component. These are the standardized skew versions of normal, t-student and ged distributions. See Fioruci et al (2014a) and Fioruci et al (2014b) for any detail. The `control` argument can be used for define the prior hyper-parameters and the simulation algorithm parameters. It is a list that can supply any of the following components:

- **$mu_tail**: the value of hyper-parameter \(\mu_{\nu}\) if `errorDist=2` or the hyper-parameter \(\mu_{\delta}\) if `errorDist=3`. Default: 8
- **$mu_gamma**: a vector with the hyper-parameters \(\mu_{\gamma_i}\). Default: `rep(0, ncol(mY))`
- **$mu_omega**: a vector with the hyper-parameters \(\mu_{\omega_i}\). Default: `rep(0, ncol(mY))`
- **$mu_alpha**: a vector with the hyper-parameters \(\mu_{\alpha_i}\). Default: `rep(0, ncol(mY))`
- **$mu_beta**: a vector with the hyper-parameters \(\mu_{\beta_i}\). Default: `rep(0, ncol(mY))`
- **$mu_a**: the value of the hyper-parameter \(\mu_a\). Default: 0
- **$mu_b**: the value of the hyper-parameter \(\mu_b\). Default: 0
- **$sigma_tail**: the value of hyper-parameter \(\sigma_{\nu}\) if `errorDist=2` or the hyper-parameter \(\sigma_{\delta}\) if `errorDist=3`. Default: 10
- **$sigma_gamma**: a vector with the hyper-parameters \(\sigma_{\gamma_i}\). Default: `rep(1.25, ncol(mY))`
- **$sigma_omega**: a vector with the hyper-parameters \(\sigma_{\omega_i}\). Default: `rep(10, ncol(mY))`
- **$sigma_alpha**: a vector with the hyper-parameters \(\sigma_{\alpha_i}\). Default: `rep(10, ncol(mY))`
- **$sigma_beta**: a vector with the hyper-parameters \(\sigma_{\beta_i}\). Default: `rep(10, ncol(mY))`
- **$sigma_a**: the value of the hyper-parameter \(\sigma_a\). Default: 10
- **$sigma_b**: the value of the hyper-parameter \(\sigma_b\). Default: 10
$\textbf{simAlg}$ the random walk Metropolis-Hasting algorithm update. Use 1 for update all parameters as one block, use 2 for update one parameter for each time and use 3 for an automatic choice.

$\textbf{nPilotSim}$ number of simulation for pilot sample if control$\text{simAlg}=3$. Default:1000

$\textbf{ScholCov}$ the cholesky decomposition matrix of the covariance matrix for simulation by one-block Metropolis-Hasting. It must to be passed if control$\text{simAlg}=1$.

$\textbf{sdSim}$ a vector with the standard deviations for simulation by one-dimensional Metropolis-Hasting. It must to be passed if control$\text{simAlg}=2$.

$\textbf{print}$ a logical variable for if the function should report the number of interactions in each 100 interactions or not. Default: TRUE

The function increaseSim() can be used to increase the length of Markov chain simulation.

The function window() can be used to filter the Markov chain simulation. In this case, all statistics are recomputed.

**Value**

An object of bayesDccGarch class, which contains a list with elements:

$\textbf{control}$ a list with the used control argument.

$\textbf{MC}$ an objetic of mcmc class with the Markov Chain simulation for all parameters. (R package coda)

$\textbf{H}$ a matrix with the Bayesian estimates of volatilities and co-volatilities.

$\textbf{R}$ a matrix with the estimates of the dynamic coditional correlation.

$\textbf{H_n1}$ Bayesian prediction of volatilities and co-volatilities for y_n+1.

$\textbf{R_n1}$ Bayesian prediction of coditional correlation for y_n+1.

$\textbf{IC}$ the Bayesian estimate of Akaike Information Criterion, Bayesian Information Criterion and Deviance Information Criterion.

$\textbf{elapsedTime}$ an object of class proc.time which is a numeric vector of length 5, containing the user, system, and total elapsed times of the process.

**Author(s)**

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada

**References**


**See Also**

bayesDccGarch-package, logLikDccGarch, plot, plotVol
Examples

data(DaxCacNik)

### Bayes DCC-GARCH(1,1) ###

mY = head(DaxCacNik, 1500)
out1 = bayesDccGarch(mY)
  # more 50000 simulations
out2 = increaseSim(out1, 50000)
  # remove first 10000 simulations and take at intervals of 20
out3 = window(out2, start=10000, thin = 20)
summary(out3)

# Plotting volatilities
plot(out3)

# Plotting Markov Chain
plot(out3$MC)

# Forecast volatility
H_pred = predict(out3, n_ahead=200)$H
plot.ts(rbind(out3$H, H_pred), main="volatility: historical and forecast")

# New data
out4 = update(out3, mY_new=DaxCacNik[1501:1628,])
plot(out4)

### Bayes univariate GARCH(1,1) ###

Dax = DaxCacNik[,1]
out = bayesDccGarch(Dax)
summary(out)
plot(out)

---

**DaxCacNik**

*Log-returns of daily indices of stock markets in Frankfurt, Paris and Tokio*

**Description**

The matrix DaxCacNik contains daily observations of the hundredfold log-returns of daily indices of stock markets in Frankfurt (DAX), Paris (CAC40) and Tokyo (NIKKEI), from 10 October 1991 until 30 December 1997 (a total of 1627 days). The stock market data is freely available at [https://robjhyndman.com/tsdldata/data/FVD1.dat](https://robjhyndman.com/tsdldata/data/FVD1.dat).
**Usage**

data(DaxCacNik)

**Author(s)**

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada

**References**


**densityFunctions**

*Density functions of multivariate Standard Skew Norm, t-Student and GED distributions*

**Description**

Compute the density function of Standard Skew Normal distribution (SSNORM) or density function of Standard Skew t-Student distribution (SST) or density function of Standard Skew GED distribution (SSGED)

**Usage**

dssnorm(x, gamma=rep(1,length(x)), log=FALSE)
dsst(x, gamma=rep(1,length(x)), nu=10, log=FALSE)
dssged(x, gamma=rep(1,length(x)), delta=2, log=FALSE)

**Arguments**

- **x**: a numeric vector for the point which the density will be computed.
- **gamma**: a numeric vector for skew parameters. Must be positive.
- **nu**: a numeric value of shape parameter of the multivariate Standard Skew t-Student distribution. Must be greater than 2.
- **delta**: a numeric value of shape parameter of GED distribution. Must be positive.
- **log**: logical; if TRUE, densities \( p \) are returned as \( \log(p) \).

**Value**

Returns the computed value of the density.

**Author(s)**

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada
logLikDccGarch

The logarithm of likelihood function of DCC-GARCH(1,1) Model.

Description

Compute the logarithm of likelihood function of DCC-GARCH(1,1) Model if mY is a matrix or the logarithm of likelihood function of GARCH(1,1) Model if mY is numeric vector.

Usage

logLikDccGarch(mY, omega = rep(0.03, ncol(mY)), alpha = rep(0.03, ncol(mY)),
beta = rep(0.8, ncol(mY)), a = 0.03, b = 0.8, gamma = rep(1, ncol(mY)),
tail = 10, errorDist = 2)

Arguments

mY a matrix of the data \((n \times k)\).
omega a numeric vector \((k \times 1)\) with the the values of \(\omega_i\) parameters. Default: rep(0.03, ncol(mY)).
alpha a numeric vector \((k \times 1)\) with the the values of \(\alpha_i\) parameters. Default: rep(0.03, ncol(mY)).
beta a numeric vector \((k \times 1)\) with the the values of \(\beta_i\) parameters. Default: rep(0.80, ncol(mY)).
a a numeric value of the \(a\) parameter. Default: 0.03.
logLikDccGarch

b

da numeric value of the b parameter. Default: 0.8.
gamma

da numeric vector (k x 1) with the values of $\gamma_i$ parameters. Default: \text{rep(1.0, ncol(mY))}.
tail

da numeric value of $\nu$ parameter if errorDist = 2 or of $\delta$ parameter if errorDist = 3. If errorDist = 1 so this arguments is no used.
errorDist

da probability distribution for errors. Use errorDist=1 for SSNorm, errorDist=2 for SST or errorDist=3 for SSGED. Default: 2.

Details

The log-likelihood of the model GARCH(1,1) is computed if mY has just one column. The arguments a and b are not consider in this case.

Value

Return a list with the elements:

$H$

da matrix where the lines are the $H_t$ values for t=1,...,n.
$value$

da the value of the logarithm of likelihood function.

Author(s)

Jose Augusto Fiorucci, Ricardo Sandes Ehlers and Francisco Louzada

References


See Also

bayesDccGarch-package, bayesDccGarch

Examples

data(DaxCacNik)

Dax = DaxCacNik[,1]

######## log-likelihood function of GARCH(1,1) model with SST innovations #####
logLikDccGarch(Dax, omega=0.03, alpha=0.03, beta=0.8, gamma=0.7)$value

######## log-likelihood function of DCC-GARCH(1,1) model with SST innovations #####
logLikDccGarch(DaxCacNik, beta=c(0.82,0.91,0.85), gamma=c(0.7, 1.3, 1.7), tail=10)$value
plot.bayesDccGarch  

Plotting volatilities for Bayesian DCC-GARCH model

Description

Produces a plot of time series and the volatilities. This is a particular case of plotVol function.

Usage

## S3 method for class 'bayesDccGarch'
plot(x, ts.names=NULL, colors = c("grey","red"), ...)

Arguments

x Object of class “bayesDccGarch”.

ts.names a vector of length \( k \) with the names of the time series.

colors a vector with the colors for plotting the returns and volatilities.

... additional arguments for plot function

Value

No return value

Author(s)

Ricardo Sandes Ehlers, Jose Augusto Fiorucci and Francisco Louzada

References


See Also

bayesDccGarch-package, bayesDccGarch, plotVol
Examples

```r
data(DaxCacNik)
mY = DaxCacNik
out = bayesDccGarch(mY, nSim=1000)
plot(out)
```

## plotVol

### Plotting volatilities of time series

#### Description
Plotting method for volatilities of time series.

#### Usage

```r
plotVol(mY, vol, ts.names=paste("TS_", 1:ncol(mY), sep=""), colors = c("grey","red"), ...)
```

#### Arguments

- `mY`: a matrix of the data \((n \times k)\).
- `vol`: a matrix \((n \times k)\) with the volatility estimates.
- `ts.names`: a vector of length \(k\) with the names of the time series.
- `colors`: a vector with name of the colors for plotting the returns and volatilities.
- `...`: additional arguments for `plot` function

#### Value
No return value

#### Author(s)
Ricardo Sandes Ehlers, Jose Augusto Fiorucci and Francisco Louzada

#### References

predict.bayesDccGarch

See Also
bayesDccGarch-package, bayesDccGarch, plot.bayesDccGarch

Examples

```r
data(DaxCacNik)
mY = DaxCacNik
out = bayesDccGarch(mY)

## The code
plotVol(mY, out$H[,c("H_1,1","H_2,2","H_3,3")], c("DAX","CAC40","NIKKEI"))

## gives the result of ##
plot(out)
```

predict.bayesDccGarch Bayesian forecast for volatilities and conditional correlations

Description
Bayesian forecast for volatilities and conditional correlations

Usage
```
## S3 method for class 'bayesDccGarch'
predict(object, ..., n_ahead = 5, bayes = F)
```

Arguments
- **object**: a bayesDccGarch object
- **...**: default argument of predict function, not used
- **n_ahead**: number of steps ahead forecast
- **bayes**: a boolean. If True, then the forecast is calculated as being the average of the forecasts across all states in the Markov chain (much slower). If False then predictions are calculated using estimation parameters (much faster).

Value
A list with elements H and R
References


Examples

```python
out = bayesDccGarch(DaxCacNik)
predict.bayesDccGarch(out, n_ahead=5)
```
Index

* Bayes
  bayesDccGarch-package, 2
* DCC-GARCH
  bayesDccGarch-package, 2
  logLikDccGarch, 8
* GARCH
  bayesDccGarch-package, 2
  logLikDccGarch, 8
* datasets
  DaxCacNik, 6
* dcc-garch
  bayesDccGarch, 3
* density-function
  densityFunctions, 7
* distribution
  densityFunctions, 7
* garch
  bayesDccGarch, 3
* models
  bayesDccGarch, 3
* multivariate-distribution
  densityFunctions, 7
* multivariate
  bayesDccGarch, 3
  bayesDccGarch-package, 2
* package
  bayesDccGarch-package, 2
* plot
  plot.bayesDccGarch, 10
  plotVol, 11
* skew-distribution
  densityFunctions, 7
* ts
  bayesDccGarch, 3
* volatility
  plot.bayesDccGarch, 10
  plotVol, 11

DaxCacNik, 6
densityFunctions, 7
dssged, 2
dssged (densityFunctions), 7
dssnorm, 2
dssnorm (densityFunctions), 7
dsst, 2
dsst (densityFunctions), 7
increaseSim (bayesDccGarch), 3
logLikDccGarch, 2, 5, 8
plot, 2, 5
plot (plot.bayesDccGarch), 10
plotVol, 2, 5, 10, 11
predict, 2
predict (predict.bayesDccGarch), 12
predict.bayesDccGarch, 12
update, 2
update (bayesDccGarch), 3
window (bayesDccGarch), 3

bayesDccGarch, 2, 3, 9, 10, 12
bayesDccGarch-package, 2