Package ‘segMGarch’

October 14, 2022

Type Package

Title Multiple Change-Point Detection for High-Dimensional GARCH Processes

Version 1.2

Date 2018-12-10

Author Haeran Cho and Karolos Korkas

Maintainer Karolos Korkas <kkorkas@yahoo.co.uk>

Description Implements a segmentation algorithm for multiple change-point detection in high-dimensional GARCH processes. It simultaneously segments GARCH processes by identifying 'common' change-points, each of which can be shared by a subset or all of the component time series as a change-point in their within-series and/or cross-sectional correlation structure.

License GPL (>= 2)

Imports Rcpp (>= 0.12.12), foreach, iterators, doParallel, fGarch, corpcor, mvtnorm, methods

Suggests MASS

LinkingTo Rcpp,RcppArmadillo

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-17 22:30:03 UTC

R topics documented:

`segMGarch-package` *Multiple Change-Point Detection for High-Dimensional GARCH Processes*

Description

Implements a segmentation algorithm for multiple change-point detection in high-dimensional GARCH processes described in Cho and Korkas (2018) ("High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155). It simultaneously segments GARCH processes by identifying 'common' change-points, each of which can be shared by a subset or all of the component time series as a change-point in their within-series and/or cross-sectional correlation structure. We adopt the Double CUSUM Binary Segmentation procedure Cho (2016), which achieves consistency in estimating both the total number and locations of the multiple change-points while permitting within-series and cross-sectional correlations, for simultaneous segmentation of the panel data of transformed time series.

It also provides additional functions and methods that relate to risk management measures and backtests.

Details

We develop a segmentation algorithm for multiple change-point detection in high-dimensional GARCH processes. It simultaneously segments GARCH processes by identifying 'common' change-points, each of which can be shared by a subset or all of the component time series as a change-point in their within-series and/or cross-sectional correlation structure. The methodology first transforms the d-dimensional time series into $d(d+1)/2$-dimensional panel data consisting of empirical residual series and their cross-products, whereby change-points in the complex ((un)conditional variance and covariance) structure are made detectable as change-points in the simpler (mean) structure of the panel data at the price of the increased dimensionality. The main routine is `garch.seg`.

Author(s)

Haeran Cho and Karolos Korkas

Maintainer: Karolos Korkas <kkorkas@yahoo.co.uk>

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2018).

Cho, Haeran. "Change-point detection in panel data via double CUSUM statistic." Electronic Journal of Statistics 10, no. 2 (2016): 2000-2038.

Examples

```
  ## Not run:
#pw.CCC.obj <- new("simMGarch")
#pw.CCC.obj <- pc_cccsim(pw.CCC.obj)
#pw.CCC.obj@d=10
#pw.CCC.obj@n=1000
#pw.CCC.obj@changepoints=c(250,750)
#pw.CCC.obj <- pc_cccsim(pw.CCC.obj)
#dcs.obj=garch.seg(pw.CCC.obj@y)
#dcs.obj$est.cps
#ts.plot(t(pw.CCC.obj@y),col="grey");grid()
#abline(v=dcs.obj$est.cps,col="red" )
#abline(v=pw.CCC.obj@changepoints,col="blue" )
#legend("bottom", legend=c("Estimated change-points", "Real change-points"),
#col=c("red", "blue"), lty=1:2, cex=0.8)

## End(Not run)
```

DQtest   *A regression-based test to backtest VaR models proposed by Engle and Manganelli (2004)*

Description

Typical VaR tests cannot control for the dependence of violations, i.e., violations may cluster while the overall (unconditional) average of violations is not significantly different from $\alpha = 1 - VaR$. The conditional expectation should also be zero meaning that $Hit_t(\alpha)$ is uncorrelated with its own past and other lagged variables (such as r_t, r_t^2 or the one-step ahead forecast VaR). To test this assumption, the dynamic conditional quantile (DQ) test is used which involves the following statistic $DQ = Hit^T X(X^T X)^{-1} X^T Hit/\alpha(1-\alpha)$ where X is the matrix of explanatory variables (e.g., raw and squared past returns) and Hit the vector collecting $Hit_t(\alpha)$. Under the null hypothesis, Engle and Manganelli (2004) show that the proposed statistic DQ follows a χ^2_q where $q = rank(X)$.

Usage

```
DQtest(y, VaR, VaR_level, lag = 1, lag_hit = 1, lag_var = 1)

## S4 method for signature 'ANY'
DQtest(y, VaR, VaR_level, lag = 1, lag_hit = 1,
  lag_var = 1)
```

Arguments

y	The time series to apply a VaR model (a single asset rerurn or portfolio return).
VaR	The forecast VaR.
VaR_level	The VaR level, typically 95% or 99%.

`lag`	The chosen lag for y.Default is 1.
`lag_hit`	The chosen lag for hit. Default is 1.
`lag_var`	The chosen lag for VaR forecasts. Default is 1.

References

Engle, Robert F., and Simone Manganelli. "CAViaR: Conditional autoregressive value at risk by regression quantiles." Journal of Business & Economic Statistics 22, no. 4 (2004): 367-381.

Examples

```
#VaR_level=0.95
#y=rnorm(1000,0,4)
#VaR=rep(quantile(y,1-VaR_level),length(y))
#y[c(17,18,19,20,100,101,102,103,104)]=-8
#lag=5
#DQtest(y,VaR,VaR_level,lag)
```

`garch.seg-class` *An S4 method to detect the change-points in a high-dimensional GARCH process.*

Description

An S4 method to detect the change-points in a high-dimensional GARCH process using the DCBS methodology described in Cho and Korkas (2018). If a `tvMGarch` is specified then it returns a `tvMGarch` object is returned. Otherwise a list of features is returned.

Usage

```
garch.seg(object, x, p = 1, q = 0, f = NULL, sig.level = 0.05,
  Bsim = 200, off.diag = TRUE, dw = NULL, do.pp = TRUE,
  do.parallel = 4)

## S4 method for signature 'ANY'
garch.seg(object = NULL, x, p = 1, q = 0, f = NULL,
  sig.level = 0.05, Bsim = 200, off.diag = TRUE, dw = NULL,
  do.pp = TRUE, do.parallel = 4)

## S4 method for signature 'tvMGarch'
garch.seg(object, p = 1, q = 0, f = NULL,
  sig.level = 0.05, Bsim = 200, off.diag = TRUE, dw = NULL,
  do.pp = TRUE, do.parallel = 4)
```

Arguments

object	A tvMGarch object. Not necessary if x is used.
x	Input data matrix, with each row representing the component time series.
p	Choose the ARCH order. Default is 1.
q	Choose the GARCH order. Default is 0.
f	The dampening factor. If NULL then f is selected automatically. Default is NULL.
sig.level	Indicates the quantile of bootstrap test statistics to be used for threshold selection. Default is 0.05.
Bsim	Number of bootstrap samples for threshold selection. Default is 200.
off.diag	If TRUE allows to look at the cross-sectional correlation structure.
dw	The length of boundaries to be trimmed off.
do.pp	Allows further post processing of the estimated change-points to reduce the risk of undersegmentation.
do.parallel	Number of copies of R running in parallel, if do.parallel = 0, %do% operator is used, see also foreach.

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2018).

Examples

```
#pw.CCC.obj <- new("simMGarch")
#pw.CCC.obj@d=10
#pw.CCC.obj@n=1000
#pw.CCC.obj@changepoints=c(250,750)
#pw.CCC.obj <- pc_cccsim(pw.CCC.obj)
#dcs.obj=garch.seg(x=empirObj@y,do.parallel = 4)
```

---

gen_pc_coef-class *A method to generate piecewise constant coefficients*

Description

An auxilliary method to calculate piecewise constant coefficients for a user-specified vector of coefficients. The change-points are controlled by the changepoints slot in the simMGarch object.

Usage

```
gen_pc_coef(object, coef)

## S4 method for signature 'simMGarch'
gen_pc_coef(object, coef)
```

Arguments

`object`	A `simMGarch` object.
`coef`	A vector of coefficients.

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2018).

Examples

```
pw.CCC.obj <- new("simMGarch")
coef.vector <- gen_pc_coef(pw.CCC.obj,c(0.2,0.4))
ts.plot(coef.vector,main="piecewise constant coefficients",ylab="coefficient",xlab="time")
```

---

kupiec — *Method to backtest VaR violation using the Kupiec statistics*

Description

An S4 method that performs backtest for VaR models using the Kupiec statistics. For a sample of n observations, the Kupiec test statistics takes the form of likelihood ratio

$$LR_{PoF} = -2\log\left(\frac{(1-\alpha)^{T-n_f}\alpha^{n_f}}{\left(1-\frac{n_f}{T}\right)^{T-n_f}\left(\frac{n_f}{T}\right)^{n_f}}\right)$$

$$LR_{TFF} = -2\log\left(\frac{\alpha(1-\alpha)^{t_f-1}}{\left(\frac{1}{t_f}\right)\left(1-\frac{1}{t_f}\right)^{t_f-1}}\right),$$

where n_f denotes the number of failures occurred and t_f the number of days until the first failure within the n observations. Under H_0, both LR_{PoF} and LR_{TFF} are asymptotically χ^2_1-distributed, and their exceedance of the critical value implies that the VaR model is inadequate.

Usage

```
kupiec(y, VaR, VaR_level, verbose = TRUE, test = "PoF")

## S4 method for signature 'ANY'
kupiec(y, VaR, VaR_level, verbose = TRUE, test = "PoF")
```

Arguments

`y`	The time series to apply a VaR model (a single asset rerurn or portfolio return).
`VaR`	The forecast VaR.
`VaR_level`	The VaR level, typically 95% or 99%.
`verbose`	If `TRUE` show the outcome. Default is `TRUE`.
`test`	Choose between PoF or TFF. Default is `PoF`.

References

Kupiec, P. "Techniques for Verifying the Accuracy of Risk Management Models." Journal of Derivatives. Vol. 3, 1995, pp. 73–84.

Examples

```
pw.CCC.obj = new("simMGarch")
pw.CCC.obj@d = 10
pw.CCC.obj@n = 1000
pw.CCC.obj@changepoints = c(250,750)
pw.CCC.obj = pc_cccsim(pw.CCC.obj)
y_out_of_sample = t(pw.CCC.obj@y[,900:1000])
w=rep(1/pw.CCC.obj@d,pw.CCC.obj@d) #an equally weighted portfolio
#VaR = quantile(t(pw.CCC.obj@y[,1:899])%*%w,0.05)
#ts.plot(y_out_of_sample%*%w,ylab="portfolio return");abline(h=VaR,col="red")
#kupiec(y_out_of_sample%*%w,rep(VaR,100),.95,verbose=TRUE,test="PoF")
```

---

pc_cccsim-class — *A method to simulate nonstationary high-dimensional CCC GARCH models.*

Description

A S4 method that takes as an input a simMGarch object and outputs a simulated nonstationary CCC model. The formulation of the of the piecewise constant CCC model is given in the simMGarch class.

Usage

```
pc_cccsim(object)

## S4 method for signature 'simMGarch'
pc_cccsim(object)
```

Arguments

object	a simMGarch object

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2018).

Examples

```
pw.CCC.obj <- new("simMGarch")
pw.CCC.obj <- pc_cccsim(pw.CCC.obj)
par(mfrow=c(1,2))
ts.plot(pw.CCC.obj@y[1,],main="a single simulated time series",ylab="series")
ts.plot(pw.CCC.obj@h[1,],main="a single simulated conditional variance",ylab="variance")
```

pc_Sigma *Method to simulate correlated variables with change-points*

Description

An S4 method that takes a simMGarch object and outputs simulated correlated time series with a piecewise constant covariance matrix. The correlations are generated as $\sigma_{i,i'} = \rho^{|i-i'|}$ with ρ taking values from $(-1, 1)$. The exact variables that will contain a change-point are randomly selected and controlled by r in the simMGarch object.

Usage

```
pc_Sigma(object)

## S4 method for signature 'simMGarch'
pc_Sigma(object)
```

Arguments

object	A simMGarch object.

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2017).

Examples

```
cp=500
n=2000
pw.CCC.obj <- new("simMGarch")
pw.CCC.obj@changepoints=cp
pw.CCC.obj@n=n
pc_Sigma.obj <- pc_Sigma(pw.CCC.obj)
par(mfrow=c(1,2))
#requires corrplot library
#correlation matrix before the changepoint
#corrplot::corrplot.mixed(cor(pc_Sigma.obj@cor_errors[1:cp,]), order="hclust", tl.col="black")
#correlation matrix after the changepoint
#corrplot::corrplot.mixed(cor(pc_Sigma.obj@cor_errors[(cp+1):n,]), order="hclust", tl.col="black")
```

simMGarch-class *An S4 class for a nonstationary CCC model.*

Description

A specification class to create an object of a simulated piecewise constant conditional correlation (CCC) model denoted by $r_t = (r_{1,t}, \ldots, r_{n,t})^T$, $t = 1, \ldots, n$ with $r_{i,t} = \sqrt{h_{i,t}}\epsilon_{i,t}$ where $h_{i,t} = \omega_i(t) + \sum_{j=1}^{p} \alpha_{i,j}(t) r_{i,t-j}^2 + \sum_{k=1}^{q} \beta_{i,k}(t) h_{i,t-k}$. In this package, we assume a piecewise constant CCC with $p = q = 1$.

Slots

y The $n \times d$ time series.

cor_errors The $n \times d$ matrix of the errors.

h The $n \times d$ matrix of the time-varying variances.

n Size of the time series.

d The number of variables (assets).

r A sparsity parameter to conrol the impact of changepoint across the series.

multp A parameter to control the covariance of errors.

changepoints The vector with the location of the changepoints.

pw A logical parameter to allow for changepoints in the error covariance matrix.

a0 The vector of the parameters a0 in the individual GARCH processes denoted by $\omega_i(t)$ in the above formula.

a1 The vector of the parameters a1 in the individual GARCH processes denoted by $\alpha_i(t)$ in the above formula.

b1 The vector of the parameters b1 in the individual GARCH processes denoted by $\beta_i(t)$ in the above formula.

BurnIn The size of the burn-in sample. Note that this only applies at the first simulated segment. Default is 50.

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2017).

Examples

```
pw.CCC.obj <- new("simMGarch")
pw.CCC.obj <- pc_cccsim(pw.CCC.obj)
par(mfrow=c(2,2))
ts.plot(pw.CCC.obj@y[1,]);ts.plot(pw.CCC.obj@y[2,])
ts.plot(pw.CCC.obj@h[1,]);ts.plot(pw.CCC.obj@h[1,])
```

TL — *Method to backtest VaR violation using the Traffic Light (TL) approach of Basel*

Description

A method that performs backtest for VaR models using the TL approach. According to Basel, a VaR model is deemed valid if the cumulative probability of observing up to n_f failures is less than 0.95 (green zone) under the binomial distribution with n (sample size) and Var level as the parameters. If the cumulative probability is between 0.95 and 0.9999 a VaR model is in yellow zone. Otherwise (>0.9999) a VaR model is in red zone.

Usage

```
TL(y, n = NULL, no_fail = NULL, VaR, VaR_level)

## S4 method for signature 'ANY'
TL(y, n = NULL, no_fail = NULL, VaR, VaR_level)
```

Arguments

`y`	The time series to apply a VaR model (a single asset rerurn or portfolio return).
`n`	If `y` is not provided, then insert sample size. Default is NULL.
`no_fail`	If `y` is not provided, then insert number of fails. Default is NULL.
`VaR`	The forecast VaR.
`VaR_level`	The VaR level, typically 95% or 99%.

References

Basle Committee on Banking Supervision (1996). "Supervisory Framework for the Use of ‘Back-testing’ in Conjunction with the Internal Models Approach to Market Risk Capital Requirements".

Examples

```
pw.CCC.obj = new("simMGarch")
pw.CCC.obj@d = 10
pw.CCC.obj@n = 1000
pw.CCC.obj@changepoints = c(250,750)
pw.CCC.obj = pc_cccsim(pw.CCC.obj)
y_out_of_sample = t(pw.CCC.obj@y[,900:1000])
w=rep(1/pw.CCC.obj@d,pw.CCC.obj@d) #an equally weighted portfolio
#VaR = quantile(t(pw.CCC.obj@y[,1:899])%*%w,0.05)
#ts.plot(y_out_of_sample%*%w,ylab="portfolio return");abline(h=VaR,col="red")
#TL(y=y_out_of_sample%*%w,VaR=rep(VaR,100),VaR_level = 0.95)
```

`tvMGarch-class` *An S4 class for a nonstationary multivariate class model.*

Description

A specification class to create an object of a nonstationary multivariate class model reserved for real (empirical) applications. It inherits from `simMGarch`.

Slots

`out_of_sample_prop` Proportion of y to keep for out-of-sample forecasting expressed in %.

`out_of_sample_y` The out of sample y matrix reserved for forecasting and backtesting exercises.

`in_sample_y` The in-sample y matrix reserved for estimation (calibration) and change-point detection.

References

Cho, Haeran, and Karolos Korkas. "High-dimensional GARCH process segmentation with an application to Value-at-Risk." arXiv preprint arXiv:1706.01155 (2018).

Examples

```
simObj <- new("simMGarch")
simObj@d <- 10
simObj@n <- 1000
simObj@changepoints <- c(250,750)
simObj <- pc_cccsim(simObj)
empirObj <- new("tvMGarch") #simulated, but treated as a real dataset for illustration
empirObj@y <- simObj@y
empirObj@out_of_sample_prop <- 0.1
#empirObj=garch.seg(object=empirObj,do.parallel = 4)##Not run
```

Index