Package ‘xdcclarge’

July 12, 2018

Type Package

Title Estimating a (c)DCC-GARCH Model in Large Dimensions

Version 0.1.0

Description Functions for Estimating a (c)DCC-GARCH Model in large dimensions
    This estimation method is consist of composite likelihood method by Pakel et al. (2014) <http://paneldataconference2015.ceu.hu/Program/Cavit-Pakel.pdf>

License GPL (>= 2)

Author Kei Nakagawa [aut, cre] (<https://orcid.org/0000-0001-5046-8128>),
    Mitsuyoshi Imamura [aut]

Maintainer Kei Nakagawa <kei.nak.0315@gmail.com>

LazyData TRUE

Imports stats, Rcpp(>= 0.10.6), nlshrink

Depends R(>= 3.0.2)

LinkingTo Rcpp(>= 0.10.6), RcppArmadillo(>= 0.2.34)

Suggests rugarch(>= 1.0.0)

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-07-12 18:50:03 UTC

R topics documented:

  cdcc_correlations .................................................. 2
  cdcc_estimation .................................................... 3
  cdcc_gradient ...................................................... 4
This function get the correlation matrix (Rt) of estimated cDCC-GARCH model.

Usage

```r
cdcc_correlations(param, stdresids, uncR, ts)
```

Arguments

- `param`  
  cDCC-GARCH parameters(alpha, beta)
- `stdresids`  
  matrix of standardized(De-GARCH) residual returns (T by N)
- `uncR`  
  unconditional correlation matrix of stdresids (N by N)
- `ts`  
  ts how many time series are you taking

Value

the correlation matrix (Rt) of estimated cDCC-GARCH model (T by N^2)

Note

Rt are vectorized values of the conditional correlation matrix(Rt) until time t(ts) for each row.
cdcc_estimation

This function estimates the parameters(alpha, beta) and time-varying correlation matrices(Rt) of cDCC-GARCH model.

Description

This function estimates the parameters(alpha, beta) and time-varying correlation matrices(Rt) of cDCC-GARCH model.

Usage

cdcc_estimation(iniPara = c(0.05, 0.93), ht, residuals, method = c("COV", "LS", "NLS"), ts = 1)

Arguments

iniPara initial cDCC-GARCH parameters(alpha, beta) of optimization
ht matrix of conditional variance vectors
residuals matrix of residual(de-mean) returns
method shrinkage method of unconditional correlation matrix(Cov: sample, LS: Linear Shrinkage, NLS: NonLinear Shrinkage)
ts ts how many time series are you taking(default: 1 latest value)

Value

time-varying correlations(Rt) and the result of estimation

Note

Rt are vectorized values of the conditional correlation matrix(Rt) until time t(ts) for each row.

Examples

library(rugarch)
library(xdcclarge)
#load data
data(us_stocks)
n<-3
Rtn<-log(us_stocks[-1:1:n]/us_stocks[-nrow(us_stocks):1:n])

# Step 1:GARCH Parameter Estimation with rugarch
spec = ugarchspec()
mmsp = multispec( replicate(spec, n = n) )
fitlist = multifit(multispec = mmsp, data = Rtn)
ht<-sigma(fitlist)^2
residuals<-residuals(fitlist)

# Step 2:cDCC-GARCH Parameter Estimation with xdcclarge
This function calculates numerical gradient of log-likelihood of cDCC-GARCH model.

**Usage**

```r
cdcc_gradient(param, ht, residuals, stdresids, uncR, d = 1e-05)
```

**Arguments**

- `param` cDCC-GARCH parameters (alpha, beta)
- `ht` matrix of conditional variance vectors (T by N)
- `residuals` matrix of residual (de-mean) returns (T by N)
- `stdresids` matrix of standrdized (De-GARCH) residual returns (T by N)
- `uncR` unconditional correlation matrix of stdresids (N by N)
- `d` \((\text{log-lik}(x+d) - \text{log-lik}(x))/d\)

**Value**

numerical gradient of log-likelihood of cDCC-GARCH model (vector)
cdcc_loglikelihood

This function calculates log-likelihood of cDCC-GARCH model.

**Description**

This function calculates log-likelihood of cDCC-GARCH model.

**Usage**

cdcc_loglikelihood(param, ht, residuals, stdresids, uncR)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>cDCC-GARCH parameters(alpha,beta)</td>
</tr>
<tr>
<td>ht</td>
<td>matrix of conditional variance vectors (T by N)</td>
</tr>
<tr>
<td>residuals</td>
<td>matrix of residual(de-mean) returns (T by N)</td>
</tr>
<tr>
<td>stdresids</td>
<td>matrix of standrdized(De-GARCH) residual returns (T by N)</td>
</tr>
<tr>
<td>uncR</td>
<td>unconditional correlation matrix of stdresids (N by N)</td>
</tr>
</tbody>
</table>

**Value**

log-likelihood of cDCC-GARCH model (scaler)

cdcc_optim

This function optimizes log-likelihood of cDCC-GARCH model.

**Description**

This function optimizes log-likelihood of cDCC-GARCH model.

**Usage**

cdcc_optim(param, ht, residuals, stdresids, uncR)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>cDCC-GARCH parameters(alpha,beta)</td>
</tr>
<tr>
<td>ht</td>
<td>matrix of conditional variance vectors (T by N)</td>
</tr>
<tr>
<td>residuals</td>
<td>matrix of residual(de-mean) returns (T by N)</td>
</tr>
<tr>
<td>stdresids</td>
<td>matrix of standrdized(De-GARCH) residual returns (T by N)</td>
</tr>
<tr>
<td>uncR</td>
<td>unconditional correlation matrix of stdresids (N by N)</td>
</tr>
</tbody>
</table>

**Value**

results of optimization
**dcc_correlations**

This function get the correlation matrix (Rt) of estimated DCC-GARCH model.

### Usage

dcc_correlations(param, stdresids, uncR, ts)

### Arguments

- **param**: DCC-GARCH parameters (alpha, beta)
- **stdresids**: matrix of standrdized(De-GARCH) residual returns (T by N)
- **uncR**: unconditional correlation matrix of stdresids (N by N)
- **ts**: ts how many time series are you taking

### Value

the correlation matrix (Rt) of estimated DCC-GARCH model (T by N^2)

### Note

Rt are vectorized values of the conditional correlation matrix(Rt) until time t(ts) for each row.

**dcc_estimation**

This function estimates the parameters(alpha, beta) and time-varying correlation matrices(Rt) of DCC-GARCH model.

### Usage

dcc_estimation(ini.para = c(0.05, 0.93), ht, residuals, method = c("COV", "LS", "NLS"), ts = 1)
**dcc_estimation**

**Arguments**

- **ini.Npara**: initial DCC-GARCH parameters (alpha, beta) of optimization
- **ht**: matrix of conditional variance vectors
- **residuals**: matrix of residual (de-mean) returns
- **method**: shrinkage method of unconditional correlation matrix (Cov: sample, LS: Linear Shrinkage, NLS: Non Linear Shrinkage)
- **ts**: ts how many time series are you taking (default: 1 latest value)

**Value**

time-varying correlations (Rt) and the result of estimation

**Note**

Rt are vectorized values of the conditional correlation matrix (Rt) until time t (ts) for each row.

**Examples**

```r
library(rugarch)
library(xdcclarge)
# load data
data(US_stocks)
n<-3
Rtn<-log(US_stocks[-1,1:n]/US_stocks[-nrow(US_stocks),1:n])

# Step 1: GARCH Parameter Estimation with rugarch
spec <- ugarchspec()
mspec = multispec( replicate(spec, n = n) )
fitlist = multifit(mspec = mspec, data = Rtn)
ht<-sigma(fitlist)^2
residuals<-residuals(fitlist)

# Step 2: DCC-GARCH Parameter Estimation with xdcclarge
DCC<-dcc_estimation(ini.Npara=c(0.05,0.93),ht,residuals)
# Time varying correlation matrix Rt at time t
(Rt<-matrix(DCC$dcc_Rt,n,n))

## Not run:
# If you want Rt at time t-s, then
s<-10
DCC<-dcc_estimation(ini.Npara=c(0.05,0.93),ht,residuals,ts = s)
matrix(DCC$dcc_Rt[s,:),n,n)

## End(Not run)
```


```plaintext
dcc_gradient

This function calculates numerical gradient of log-likelihood of DCC-GARCH model.

Usage

dcc_gradient(param, ht, residuals, stdresids, uncr, d = 1e-05)

Arguments

- `param`: DCC-GARCH parameters(alpha,beta)
- `ht`: matrix of conditional variance vectors (T by N)
- `residuals`: matrix of residual(de-mean) returns (T by N)
- `stdresids`: matrix of standrdized(De-GARCH) residual returns (T by N)
- `uncR`: unconditional correlation matrix of stdresids (N by N)
- `d`: (log-lik(x+d) - log-lik(x))/d

Value

- numerical gradient of log-likelihood of DCC-GARCH model (vector)
```

```plaintext
dcc_loglikelihood

This function calculates log-likelihood of DCC-GARCH model.

Usage

dcc_loglikelihood(param, ht, residuals, stdresids, uncr)

Arguments

- `param`: DCC-GARCH parameters(alpha,beta)
- `ht`: matrix of conditional variance vectors (T by N)
- `residuals`: matrix of residual(de-mean) returns (T by N)
- `stdresids`: matrix of standrdized(De-GARCH) residual returns (T by N)
- `uncR`: unconditional correlation matrix of stdresids (N by N)
```
Value

log-likelihood of DCC-GARCH model (scaler)

Description

This function optimizes log-likelihood of DCC-GARCH model.

Usage

dcc_optim(param, ht, residuals, stdresids, uncR)

Arguments

param  DCC-GARCH parameters(alpha,beta)
ht     matrix of conditional variance vectors (T by N)
residuals  matrix of residual(de-mean) returns (T by N)
stdresids  matrix of standrdized(De-GARCH) residual returns (T by N)
uncR    unconditional correlation matrix of stdresids (N by N)

Value

results of optimization

us_stocks  the closing price data of us stocks in SP500 index from 2006-03-31 to 2014-03-31 from yahoo finance.

Description

the closing price data of us stocks in SP500 index from 2006-03-31 to 2014-03-31 from yahoo finance.

Format

A data frame with 2013 rows and 460 variables:

Source

Yahoo finance
Description

Functions for Estimating a (c)DCC-GARCH Model in large dimensions based on a publication by Engle et al. (2017) and Nakagawa et al. (2018). This estimation method consists of composite likelihood method by Pakel et al. (2014) and (Non-)linear shrinkage estimation of covariance matrices by Ledoit and Wolf (2004, 2015, 2016).

Details

To estimate the covariance matrix in financial time series, it is necessary to consider two important aspects: the cross section and the time series. With regard to the cross section, we have the difficulty of correcting the biases of the sample covariance matrix eigenvalues in a large number of time series. With regard to the time series aspect, we have to account for volatility clustering and time-varying correlations. This package implements the improved estimation of the covariance matrix based on the following publications:


Index

cdcc_correlations, 2  
cdcc_estimation, 3  
cdcc_gradient, 4  
cdcc_loglikelihood, 5  
cdcc_optim, 5  

dcc_correlations, 6  
dcc_estimation, 6  
dcc_gradient, 8  
dcc_loglikelihood, 8  
dcc_optim, 9  

us_stocks, 9  

xdcclarge, 10  
xdcclarge-package (xdcclarge), 10