Package ‘robustcov’

August 4, 2021

Type Package

Title Collection of Robust Covariance and (Sparse) Precision Matrix Estimators

Version 0.1

Date 2021-07-18

Maintainer Yunyi Shen <yshen99@wisc.edu>


License GPL-3

Imports Rcpp (>= 1.0.6), glasso, caret

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

NeedsCompilation yes

Author Yunyi Shen [aut, cre] (<https://orcid.org/0000-0003-2779-6507>),
        David Simcha [cph]

Repository CRAN

Date/Publication 2021-08-04 10:00:05 UTC

R topics documented:

  conta_normal ......................................................... 2
  corKendall .......................................................... 3
  corQuadrant .......................................................... 3
  corSpearman ......................................................... 4
  covGK ................................................................. 4
  covNPD ............................................................... 5
  covOGK ............................................................... 5
  covSpearmanU ......................................................... 6
  cvglasso .............................................................. 6
  nearPPSD ............................................................. 7
  negLLrobOmega ....................................................... 8
Sample contaminated normal

Description

This function samples normal distribution with normal contamination

Usage

conta_normal(
  n,
  Omega,
  byrow = FALSE,
  cont_rate = 0.05,
  mu = 10,
  sd = sqrt(0.2)
)

Arguments

n          samplesize
Omega      precision matrix of the normal
byrow      whether the contamination happened by row? FALSE stand for cellwise contamination
cont_rate  how many cells/rows are contaminated?
mu         mean of the contamination
sd         standard deviation of the contamination

Value

a matrix of contaminated (multivariate) normal distributed data, row as sample
**corKendall**

**Kendall’s tau**

Description
This routine calculates the Kendall’s tau

Usage
```r
corKendall(data)
```

Arguments
- `data` the n by p raw data matrix

Value
A matrix with dimension p by p, Kendall’s tau

Examples
```r
corKendall(matrix(rnorm(500),100,5))
```

**corQuadrant**

**Quadrant correlation coefficients**

Description
This routine calculates Quadrant correlation coefficients

Usage
```r
corQuadrant(data)
```

Arguments
- `data` the n by p raw data matrix

Value
A matrix with dimension p by p, Quadrant correlation coefficients

Examples
```r
corQuadrant(matrix(rnorm(500),100,5))
```
corSpearman  

**Spearman correlation**

**Description**
This routine calculates the Spearman correlation

**Usage**
corSpearman(data)

**Arguments**
data  
the n by p raw data matrix

**Value**
a matrix with dimension p by p of spearman correlations

**Examples**
corSpearman(matrix(rnorm(500),100,5))

covGK  

**Gnanadesikan-Kettenring estimator for *covariance***

**Description**
This routine calculates the Gnanadesikan-Kettenring estimator, diagonal will be MAD

**Usage**
covGK(data)

**Arguments**
data  
the n by p raw data matrix

**Value**
a matrix with dimension p by p, GK estimator, note that it’s not necessarily positive

**Examples**
covGK(matrix(rnorm(500),100,5))
**covNPD**

*NPD estimator for *covariance* based on Qn*

### Description
This routine calculates the NPD estimator for *covariance* based on Qn.

### Usage
```
covNPD(data, eigenTol = 1e-06, convTol = 1e-07, psdTol = 1e-08, maxit = 1000L)
```

### Arguments
- **data**: the n by p raw data matrix
- **eigenTol**: tolerance in eigen system, used in finding nearest positive matrix
- **convTol**: tolerance in cov, used in finding nearest positive matrix
- **psdTol**: tolerance in psd, used in finding nearest positive matrix
- **maxit**: max iterations in finding nearest positive matrix

### Value
A matrix with dimension p by p, NPD estimator.

### Examples
```
covNPD(matrix(rnorm(500),100,5))
```

---

**covOGK**

*Orthogonalized Gnanadesikan-Kettenring (OGK) estimator for *covariance**

### Description
This routine calculates the Orthogonalized Gnanadesikan-Kettenring (OGK) estimator for *covariance*, using scale estimation of Gk, as in Maronna and Zamar.

### Usage
```
covOGK(data)
```

### Arguments
- **data**: the n by p raw data matrix
Value
   a matrix with dimension p by p, OGK estimator

Examples
   covOGK(matrix(rnorm(500),100,5))

---

covSpearmanU  SpearmanU estimator for covariance

Description
   This routine calculates the SpearmanU, the pairwise covariance matrix estimator proposed in Oellererand Croux

Usage
   covSpearmanU(data)

Arguments
   data  the n by p raw data matrix

Value
   a matrix with dimension p by p of spearmanU correlation

Examples
   covSpearmanU(matrix(rnorm(500),100,5))

---

cvglasso  Cross validation to chose tuning parameter of glasso

Description
   This routine use k fold cross validation to chose tuning parameter

Usage
   cvglasso(
       data,
       k = 10,
       covest = cov,
       rhos = seq(0.1, 1, 0.1),
       evaluation = negLLrobOmega,
       ...
   )
Arguments

- **data**
  The full dataset, should be a matrix or a data.frame, row as sample
- **k**
  number of folds
- **covest**
  a *function* or name of a function (string) that takes a matrix to estimate covariance
- **rhos**
  a vector of tuning parameter to be tested
- **evaluation**
  a *function* or name of a function (string) that takes only two arguments, the estimated covariance and the test covariance, when NULL, we use negative log likelihood on test sets
- **...**
  extra arguments send to glasso

Value

A matrix with k rows, each row is the evaluation loss of that fold

Examples

```r
cvglasso(matrix(rnorm(100),20,5))
```

---

**nearPPSD**

**nearest positive semi-definite projection of a matrix**

Description

This routine calculate the nearest positive semi-definite projection

Usage

```r
nearPPSD(X, eigenTol = 1e-06, convTol = 1e-07, psdTol = 1e-08, maxit = 1000L)
```

Arguments

- **X**
  the matrix
- **eigenTol**
  tolerance in eigen system, used in finding nearest positive matrix
- **convTol**
  tolerance in cov, used in finding nearest positive matrix
- **psdTol**
  tolerance in psd, used in finding nearest positive matrix
- **maxit**
  max iterations in finding nearest positive matrix

Value

A matrix which is the nearest positive semi-definite matrix of input X
### Description
The default evaluation function in cross-validation, -log likelihood on test set.

### Usage

```r
negLLrobOmega(Sigma_hat, Sigma)
```

### Arguments
- **Sigma_hat**: the estimated covariance matrix of training set.
- **Sigma**: the covariance matrix of test sets.

### Value
- -log likelihood

---

### Description
This routine samples alternative multivariate t distribution.

### Usage

```r
raltert(n, Omega, nu)
```

### Arguments
- **n**: sample size.
- **Omega**: precision matrix of dimension p by p.
- **nu**: degree of freedom.

### Value
- a matrix with dimension n by p, each row is a sample.
**rmvnorm**

*Multivariate normal distribution with 0 mean*

**Description**

This routine samples multivariate normal distribution of mean 0 from precision matrix

**Usage**

```r
rmvnorm(n, Omega)
```

**Arguments**

- `n` sample size
- `Omega` **precision** matrix of dimension p by p

**Value**

A matrix with dimension n by p, each row is a sample

---

**rmvt**

*Multivariate t distribution*

**Description**

This routine samples multivariate t distribution

**Usage**

```r
rmvt(n, Omega, nu)
```

**Arguments**

- `n` sample size
- `Omega` **precision** matrix of dimension p by p
- `nu` degree of freedom

**Value**

A matrix with dimension n by p, each row is a sample
Description

This routine fits glasso using a robust covariance matrix

Usage

robglasso(
  data,
  covest = cov,
  rho = 0.1,
  CV = FALSE,
  k = 10,
  grids = 15,
  evaluation = negLLrobOmega,
  ...
)

Arguments

data      raw data, should be a matrix or a data.frame, row as sample

covest    a *function* or name of a function (string) that takes a matrix to estimate covariance

rho       a scalar or vector of tuning parameters to be chosen, if CV=FALSE, should be a scalar. If CV=TRUE scalar input will be override and tuning parameter will be chosen based on CV

CV        bool, whether doing cross validation for tuning parameter, if set to TRUE and rho is a scalar, the candidate will be chosen automatically by log spacing between 0.01 max covariance and max covariance with number of grids

k         fold for cross validation if applicable

grids     number of candidate tuning parameters in cross validation

evaluation a *function* or name of a function (string) that takes only two arguments, the estimated covariance and the test covariance, when NULL, we use negative log likelihood on test sets

...       extra argument sent to glasso::glasso

Value

a glasso return (see ?glasso::glasso), most important one is $X$ the estimated sparse precision, with an extra entry of tuning parameter lambda

Examples

robglasso(matrix(rnorm(100),20,5))
Index

conta_normal, 2
corKendall, 3
corQuadrant, 3
corSpearman, 4
covGK, 4
covNPD, 5
covOGK, 5
covSpearmanU, 6
cvglasso, 6

nearPPSD, 7
negLLrobOmega, 8

raltert, 8
rmvnorm, 9
rmvt, 9
robglasso, 10