Package ‘FinCovRegularization’

April 25, 2016

Type Package
Title Covariance Matrix Estimation and Regularization for Finance
Version 1.1.0
Description Estimation and regularization for covariance matrix of asset returns. For covariance matrix estimation, three major types of factor models are included: macroeconomic factor model, fundamental factor model and statistical factor model. For covariance matrix regularization, four regularized estimators are included: banding, tapering, hard-thresholding and soft-thresholding. The tuning parameters of these regularized estimators are selected via cross-validation.

URL http://github.com/yanyachen/FinCovRegularization

BugReports http://github.com/yanyachen/FinCovRegularization/issues

Depends R (>= 2.10)
Imports stats, graphics, quadprog
License GPL-2
LazyData true
RoxygenNote 5.0.1

Author YaChen Yan [aut, cre], FangZhu Lin [aut]
Maintainer YaChen Yan <yanyachen21@gmail.com>

Repository CRAN

Date/Publication 2016-04-25 15:32:07

R topics documented:

banding ................................................................. 2
banding.cv .............................................................. 3
F.norm2 ................................................................. 4
FinCovRegularization .................................................. 4
**banding**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FundamentalFactor.Cov</td>
<td>5</td>
</tr>
<tr>
<td>GMVP</td>
<td>5</td>
</tr>
<tr>
<td>hard.thresholding</td>
<td>6</td>
</tr>
<tr>
<td>Ind.Cov</td>
<td>7</td>
</tr>
<tr>
<td>m.excess.c10sp9003</td>
<td>7</td>
</tr>
<tr>
<td>MacroFactor.Cov</td>
<td>8</td>
</tr>
<tr>
<td>O.norm2</td>
<td>8</td>
</tr>
<tr>
<td>RiskParity</td>
<td>9</td>
</tr>
<tr>
<td>soft.thresholding</td>
<td>9</td>
</tr>
<tr>
<td>StatFactor.Cov</td>
<td>10</td>
</tr>
<tr>
<td>tapering</td>
<td>11</td>
</tr>
<tr>
<td>tapering.cv</td>
<td>11</td>
</tr>
<tr>
<td>threshold.cv</td>
<td>13</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>banding</td>
<td>15</td>
</tr>
</tbody>
</table>

---

**Banding Operator on Covariance Matrix**

**Description**

Apply banding operator on a covariance matrix with a banding parameter.

**Usage**

```r
banding(sigma, k = 0)
```

**Arguments**

- `sigma`: a p*p covariance matrix
- `k`: banding parameter

**Value**

A regularized covariance matrix after banding operation

**References**

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

**Examples**

```r
data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
banding(cov.SAM, 7)
```
banding.cv

Select Tuning Parameter for Banding Covariance Matrix by CV

Description
Apply K-fold cross-validation for selecting tuning parameters for banding covariance matrix using grid search strategy

Usage
banding.cv(matrix, n.cv = 10, norm = "F", seed = 142857)

Arguments
- **matrix**: a N*p matrix, N indicates sample size and p indicates the dimension
- **n.cv**: times that cross-validation repeated, the default number is 10
- **norm**: the norms used to measure the cross-validation errors, which can be the Frobenius norm "F" or the operator norm "O"
- **seed**: random seed, the default value is 142857

Details
For cross-validation, this function split the sample randomly into two pieces of size n1 = n-n/log(n) and n2 = n/log(n), and repeat this k times

Value
An object of class "CovCv" containing the cross-validation's result for covariance matrix regularization, including:

- **regularization**: regularization method, which is "Banding"
- **parameter.opt**: selected optimal parameter by cross-validation
- **cv.error**: the corresponding cross-validation errors
- **n.cv**: times that cross-validation repeated
- **norm**: the norm used to measure the cross-validation error
- **seed**: random seed

References
"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi
Examples

```r
data(m.excess.c10sp9003)
retcov.cv <- banding.cv(m.excess.c10sp9003, n.cv = 10,
                       norm = "F", seed = 142857)
summary(retcov.cv)
plot(retcov.cv)
# Low dimension
```

---

**F.norm2**

*The Squared Frobenius Norm*

Description

Calculate the squared Frobenius norm of a matrix

Usage

```r
F.norm2(matrix)
```

Arguments

- `matrix`: a matrix

Value

a scalar of the squared Frobenius norm

Examples

```r
data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
F.norm2(cov.SAM)
```

---

**FinCovRegularization**

*FinCovRegularization: Covariance Matrix Estimation and Regularization for Finance*

Description

Estimation and regularization for covariance matrix of asset returns. For covariance matrix estimation, three major types of factor models are included: macroeconomic factor model, fundamental factor model and statistical factor model. For covariance matrix regularization, four regularized estimators are included: banding, tapering, hard-thresholding and soft-thresholding. The tuning parameters of these regularized estimators are selected via cross-validation.
**Description**

Estimate covariance matrix by fitting a fundamental factor model using OLS or WLS regression

**Usage**

FundamentalFactor.Cov(assets, exposure, method = "WLS")

**Arguments**

- **assets**: a N*p matrix of asset returns, N indicates sample size and p indicates the dimension of asset returns
- **exposure**: a p*q matrix of exposure indicator for the fundamental factor model, p corresponds to the dimension of asset returns, q indicates the number of fundamental industries
- **method**: a character, indicating regression method: "OLS" or "WLS"

**Value**

an estimated p*p covariance matrix

**Examples**

```r
data(m.excess.c10sp9003)
assets <- m.excess.c10sp9003[,1:10]
Indicator <- matrix(0,10,3)
dimnames(Indicator) <- list(colnames(assets),c("Drug","Auto","Oil"))
Indicator[c("ABT","LLY","MRK","PFE"),"Drug"] <- 1
Indicator[c("F","GM"),"Auto"] <- 1
Indicator[c("BP","CVX","RD","XOM"),"Oil"] <- 1
FundamentalFactor.Cov(assets,exposure=Indicator,method="WLS")
```

---

**GMVP**

**Global Minimum Variance Portfolio**

**Description**

Computing a global minimum variance portfolio weights from the estimated covariance matrix of return series.

**Usage**

GMVP(cov.mat, short = TRUE)
hard.thresholding

Arguments

- cov.mat: an estimated p*p covariance matrix
- short: logical flag, indicating whether shortsales on the risky assets are allowed

Value

a numerical vector containing the estimated portfolio weights

Examples

data(m.excess.c10sp9003)
assets <- m.excess.c10sp9003[,1:10]
GMVP(cov(assets), short=TRUE)
GMVP(cov(assets), short=FALSE)

Description

Hard-Thresholding Operator on Covariance Matrix

Apply hard-thresholding operator on a covariance matrix with a hard-thresholding parameter.

Usage

hard.thresholding(sigma, threshold = 0.5)

Arguments

- sigma: a p*p covariance matrix
- threshold: hard-thresholding parameter

Value

a regularized covariance matrix after hard-thresholding operation

References

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

Examples

data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
hard.thresholding(cov.SAM, threshold = 0.001)
**Ind.Cov**

*Independence opeator on Covariance Matrix*

**Description**

Apply independence model on a covariance matrix.

**Usage**

`Ind.Cov(sigma)`

**Arguments**

- `sigma`: a covariance matrix

**Value**

a regularized covariance matrix after applying independence model

**Examples**

```r
data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
Ind.Cov(cov.SAM)
```

---

**m.excess.c10sp9003**

*10 stock and S&P 500 excess returns*

**Description**

A dataset containing monthly excess returns of 10 stocks and S&P 500 index return from January 1990 to December 2003

**Usage**

`data(m.excess.c10sp9003)`

**Format**

A matrix with 168 rows and 11 variables
MacroFactor.Cov  
*Covariance Matrix Estimation by Macroeconomic Factor Model*

**Description**

Estimate covariance matrix by fitting a macroeconomic factor model using time series regression

**Usage**

```r
MacroFactor.Cov(assets, factor)
```

**Arguments**

- `assets` : a N*p matrix of asset returns, N indicates sample size and p indicates the dimension of asset returns
- `factor` : a numerical vector of length N, or a N*q matrix of macroeconomic factor(s), q indicates the dimension of factors

**Value**

- an estimated p*p covariance matrix

**Examples**

```r
data(m.excess.c10sp9003)
assets <- m.excess.c10sp9003[,1:10]
factor <- m.excess.c10sp9003[,11]
MacroFactor.Cov(assets, factor)
```

---

O.norm2  
*The Squared Operator Norm*

**Description**

Calculate the squared Operator norm of a matrix

**Usage**

```r
O.norm2(matrix)
```

**Arguments**

- `matrix` : a matrix

**Value**

- a scalar of the squared Operator norm
**RiskParity**

**Examples**

```r
data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
O.norm2(cov.SAM)
```

---

**Description**

Computing a Risk Parity portfolio weights from the estimated covariance matrix of return series.

**Usage**

```r
RiskParity(cov.mat)
```

**Arguments**

- `cov.mat`: an estimated p*p covariance matrix

**Value**

a numerical vector containing the estimated portfolio weights

**Examples**

```r
data(m.excess.c10sp9003)
assets <- m.excess.c10sp9003[,1:10]
RiskParity(cov(assets))
```

---

**soft.thresholding**

**Soft-Thresholding Operator on Covariance Matrix**

**Description**

Apply soft-thresholding operator on a covariance matrix with a soft-thresholding parameter.

**Usage**

```r
soft.thresholding(sigma, threshold = 0.5)
```

**Arguments**

- `sigma`: a covariance matrix
- `threshold`: soft-thresholding parameter
a regularized covariance matrix after soft-thresholding operation

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
soft.thresholding(cov.SAM, threshold = 0.001)

StatFactor.Cov

Covariance Matrix Estimation by Statistical Factor Model

Estimate covariance matrix by fitting a statistical factor model using principle components analysis

StatFactor.Cov(assets, k = 0)

assets a matrix of asset returns
k numbers of factors, if k = 0, automatically estimating by Kaiser method

an estimated p*p covariance matrix

data(m.excess.c10sp9003)
assets <- m.excess.c10sp9003[,1:10]
StatFactor.Cov(assets, 3)
tapering

Tapering Operator on Covariance Matrix

Description

Apply tapering operator on a covariance matrix with tapering parameters.

Usage

tapering(sigma, l, h = 1/2)

Arguments

- **sigma**: a p*p covariance matrix
- **l**: tapering parameter
- **h**: the ratio between taper l_h and parameter l

Value

a regularized covariance matrix after tapering operation

References

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

Examples

data(m.excess.c10sp9003)
cov.SAM <- cov(m.excess.c10sp9003)
tapering(cov.SAM, l=7, h = 1/2)

tapering.cv

Select Tuning Parameter for Tapering Covariance Matrix by CV

Description

Apply K-fold cross-validation for selecting tuning parameters for tapering covariance matrix using grid search strategy

Usage

tapering.cv(matrix, h = 1/2, n.cv = 10, norm = "F", seed = 142857)
tapering.cv

Arguments

matrix  a N*p matrix, N indicates sample size and p indicates the dimension
h       the ratio between taper l_h and parameter l
n.cv    times that cross-validation repeated, the default number is 10
norm    the norms used to measure the cross-validation errors, which can be the Frobenius norm "F" or the operator norm "O"
seed    random seed, the default value is 142857

Details

For cross-validation, this function split the sample randomly into two pieces of size n1 = n-n/log(n) and n2 = n/log(n), and repeat this k times

Value

An object of class "CovCv" containing the cross-validation's result for covariance matrix regularization, including:

regularization regularization method, which is "Tapering"
parameter.opt  selected optimal parameter by cross-validation
cv.error      the corresponding cross-validation errors
n.cv          times that cross-validation repeated
norm          the norm used to measure the cross-validation error
seed          random seed

References

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

Examples

data(m.excess.c10sp9003)
retcov.cv <- tapering.cv(m.excess.c10sp9003, n.cv = 10,
                         norm = "F", seed = 142857)
summary(retcov.cv)
plot(retcov.cv)
# Low dimension
**threshold.cv**

*Select Tuning Parameter for Thresholding Covariance Matrix by CV*

**Description**

Apply K-fold cross-validation for selecting tuning parameters for thresholding covariance matrix using grid search strategy.

**Usage**

```r
threshold.cv(matrix, method = "hard", thresh.len = 20, n.cv = 10, 
             norm = "F", seed = 142857)
```

**Arguments**

- `matrix`: a N*p matrix, N indicates sample size and p indicates the dimension
- `method`: thresholding method, "hard" or "soft"
- `thresh.len`: the number of thresholding values tested in cross-validation, the thresholding values will be a sequence of `thresh.len` equally spaced values from minimum threshold constant to largest covariance in sample covariance matrix
- `n.cv`: times that cross-validation repeated, the default number is 10
- `norm`: the norms used to measure the cross-validation errors, which can be the Frobenius norm "F" or the operator norm "O"
- `seed`: random seed, the default value is 142857

**Details**

For cross-validation, this function split the sample randomly into two pieces of size \( n_1 = n - n/\log(n) \) and \( n_2 = n/\log(n) \), and repeat this k times.

**Value**

An object of class "CovCv" containing the cross-validation's result for covariance matrix regularization, including:

- `regularization`: regularization method, which is "Hard Thresholding" or "Soft Thresholding"
- `parameter.opt`: selected optimal parameter by cross-validation
- `cv.error`: the corresponding cross-validation errors
- `n.cv`: times that cross-validation repeated
- `norm`: the norm used to measure the cross-validation error
- `seed`: random seed
- `threshold.grid`: thresholding values tested in cross-validation
References

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

Examples

```r
data(m.excess.c10sp9003)
retpcv <- threshold.cv(m.excess.c10sp9003, method = "hard",
                       thresh.len = 20, n.cv = 10, norm = "F", seed = 142857)
summary(retpcv)
plot(retpcv)
# Low dimension
```
Index

*Topic datasets
  m.excess.c10sp9003, 7

banding, 2
banding.cv, 3

F.norm2, 4
FinCovRegularization, 4
FinCovRegularization-package
  (FinCovRegularization), 4
FundamentalFactor.Cov, 5

GMVP, 5

hard.thresholding, 6

Ind.Cov, 7
m.excess.c10sp9003, 7
MacroFactor.Cov, 8

O.norm2, 8

RiskParity, 9

soft.thresholding, 9
StatFactor.Cov, 10

tapering, 11
tapering.cv, 11
threshold.cv, 13