Package ‘FinCovRegularization’

March 1, 2015

Type Package
Title Covariance Matrix Estimation and Regularization for Finance
Version 1.0.0
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 quadprog
License GPL-2
LazyData true
Author Yachen Yan [aut, cre]
Maintainer Yachen Yan <yanyachen21@gmail.com>
NeedsCompilation no
Repository CRAN
Date/Publication 2015-03-01 08:16:11

R topics documented:

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

**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**

```r
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 \(n1 = \frac{n-n}{\log(n)}\) and \(n2 = \frac{n}{\log(n)}\), 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.
**FundamentalFactor.Cov**  
Covariance Matrix Estimation by Fundamental Factor Model

**Description**

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

**Usage**

```r
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**

```r
GMVP(cov.mat, short = TRUE)
```
**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**

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

---

**hard.thresholding**

- **Hard-Thresholding Operator on Covariance Matrix**

**Description**

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

**Usage**

```r
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**

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

Ind.Cov

Independence operator 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

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
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

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
O.norm2(matrix)

Arguments

matrix  a matrix

Value
a scalar of the squared Operator norm
plot.CovCv

**Examples**

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

---

**Description**

Visualizes the results of covariance matrix regularization parameter tuning

**Usage**

```r
## S3 method for class 'CovCv'
plot(x, ...)
```

**Arguments**

- `x` CovCv object to plot
- `...` other arguments ignored (for compatibility with generic)

---

**RiskParity**

**Risk Parity Portfolio**

**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 \hspace{1em} \textit{Soft-Thresholding Operator on Covariance Matrix}

\textbf{Description}

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

\textbf{Usage}

\begin{verbatim}
  soft.thresholding(sigma, threshold = 0.5)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{sigma} \hspace{1em} a covariance matrix
  \item \texttt{threshold} \hspace{1em} soft-thresholding parameter
\end{itemize}

\textbf{Value}

a regularized covariance matrix after soft-thresholding operation

\textbf{References}

"High-Dimensional Covariance Estimation" by Mohsen Pourahmadi

\textbf{Examples}

\begin{verbatim}
  data(m.excess.c10sp9003)
  cov.SAM <- cov(m.excess.c10sp9003)
  soft.thresholding(cov.SAM, threshold = 0.001)
\end{verbatim}

\textbf{StatFactor.Cov} \hspace{1em} \textit{Covariance Matrix Estimation by Statistical Factor Model}

\textbf{Description}

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

\textbf{Usage}

\begin{verbatim}
  StatFactor.Cov(assets, k = 0)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{assets} \hspace{1em} a matrix of asset returns
  \item \texttt{k} \hspace{1em} numbers of factors, if \texttt{k} = 0, automatically estimating by Kaiser method
\end{itemize}
tapering

Value
an estimated p*p covariance matrix

Examples

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)
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)

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.
threshold.cv

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

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 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 "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)
retcov.cv <- threshold.cv(m.excess.c10sp9003, method = "hard",
                          thresh.len = 20, n.cv = 10, norm = "F", seed = 142857)
summary(retcov.cv)
plot(retcov.cv)
# 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

plot.CovCv, 9

RiskParity, 9

soft.thresholding, 10
StatFactor.Cov, 10

tapering, 11
tapering.cv, 12
threshold.cv, 13