Package ‘highOrderPortfolios’

October 20, 2022

Title  Design of High-Order Portfolios Including Skewness and Kurtosis

Version  0.1.1

Date  2022-10-20

Description  The classical Markowitz’s mean-variance portfolio formulation ignores heavy tails and skewness. High-order portfolios use higher order moments to better characterize the return distribution. Different formulations and fast algorithms are proposed for high-order portfolios based on the mean, variance, skewness, and kurtosis.

The package is based on the papers:

Maintainer  Daniel P. Palomar <daniel.p.palomar@gmail.com>

URL  https://github.com/dppalomar/highOrderPortfolios,
     https://www.danielppalomar.com

BugReports  https://github.com/dppalomar/highOrderPortfolios/issues

License  GPL-3

Encoding  UTF-8

LazyData  true

RoxygenNote  7.2.1

Imports  ECOSolveR, lpSolveAPI, nloptr, PerformanceAnalytics, quadprog,
          fitHeavyTail (>= 0.1.4), stats, utils

Suggests  knitr, ggplot2, rmarkdown, R.rsp, testthat (>= 3.0.0)

VignetteBuilder  knitr, rmarkdown, R.rsp

Config/testthat/edition  3

NeedsCompilation  yes

Author  Daniel P. Palomar [cre, aut],
        Rui Zhou [aut],
        Xiwen Wang [aut]
Description

The classical Markowitz's mean-variance portfolio formulation ignores heavy tails and skewness. High-order portfolios use higher order moments to better characterize the return distribution. Different formulations and fast algorithms are proposed for high-order portfolios based on the mean, variance, skewness, and kurtosis.

Functions

design_MVK_portfolio_via_sample_moments(), design_MVK_portfolio_via_skew_t(), and design_MVKtilting_portfolio_via_sample_moments()

Help

For a quick help see the README file: GitHub-README.

Author(s)

Rui Zhou, Xiwen Wang, and Daniel P. Palomar
References


Description

Design high-order portfolio by tilting a given portfolio to the MVSK efficient frontier (i.e., mean, variance, skewness, and kurtosis):

\[
\begin{align*}
\text{minimize} & \quad - \delta \\
m_1(w) & \geq m_1(w_0) + \delta d_1 \\
m_2(w) & \leq m_2(w_0) - \delta d_2 \\
m_3(w) & \geq m_3(w_0) + \delta d_3 \\
m_4(w) & \leq m_4(w_0) - \delta d_4 \\
(w - w_0)'\Sigma(w - w_0) & \leq \kappa^2
\end{align*}
\]

subject to \( ||w||_1 \leq \text{leverage}, \sum(w) = 1 \).

Usage

```r
design_MVSKtilting_portfolio_via_sample_moments(
    d = rep(1, 4),
    X_moments,
    w_init = rep(1/length(X_moments$mu), length(X_moments$mu)),
    w0 = w_init,
    w0_moments = NULL,
    leverage = 1,
    kappa = 0,
    method = c("Q-MVSKT", "L-MVSKT"),
    tau_w = 1e-05,
    tau_delta = 1e-05,
    gamma = 1,
    zeta = 1e-08,
    maxiter = 100,
    ftol = 1e-05,
    wtol = 1e-05,
    theta = 0.5,
    stopval = -Inf
)
```
Arguments

\begin{itemize}
  \item \texttt{d} \hfill Numerical vector of length 4 indicating the weights of first four moments.
  \item \texttt{X_moments} \hfill List of moment parameters, see \texttt{estimate_sample_moments}.
  \item \texttt{w_init} \hfill Numerical vector indicating the initial value of portfolio weights.
  \item \texttt{w0} \hfill Numerical vector indicating the reference portfolio vector.
  \item \texttt{w0_moments} \hfill Numerical vector indicating the reference moments.
  \item \texttt{leverage} \hfill Number (\geq 1) indicating the leverage of portfolio.
  \item \texttt{kappa} \hfill Number indicating the maximum tracking error volatility.
  \item \texttt{method} \hfill String indicating the algorithm method, must be one of: "Q-MVSK", "MM", "DC".
  \item \texttt{tau_w} \hfill Number (\geq 0) guaranteeing the strong convexity of approximating function.
  \item \texttt{tau_delta} \hfill Number (\geq 0) guaranteeing the strong convexity of approximating function.
  \item \texttt{gamma} \hfill Number (0 < gamma \leq 1) indicating the initial value of gamma.
  \item \texttt{zeta} \hfill Number (0 < zeta < 1) indicating the diminishing parameter of gamma.
  \item \texttt{maxiter} \hfill Positive integer setting the maximum iteration.
  \item \texttt{ftol} \hfill Positive number setting the convergence criterion of function objective.
  \item \texttt{wtol} \hfill Positive number setting the convergence criterion of portfolio weights.
  \item \texttt{theta} \hfill Number (0 < theta < 1) setting the combination coefficient when enlarge feasible set.
  \item \texttt{stopval} \hfill Number setting the stop value of objective.
\end{itemize}

Value

A list containing the following elements:

\begin{itemize}
  \item \texttt{w} \hfill Optimal portfolio vector.
  \item \texttt{delta} \hfill Maximum tilting distance of the optimal portfolio.
  \item \texttt{cpu_time_vs_iterations} \hfill Time usage over iterations.
  \item \texttt{objfun_vs_iterations} \hfill Objective function over iterations.
  \item \texttt{iterations} \hfill Iterations index.
  \item \texttt{moments} \hfill Moments of portfolio return at optimal portfolio weights.
  \item \texttt{improvement} \hfill The relative improvement of moments of designed portfolio w.r.t. the reference portfolio.
\end{itemize}

Author(s)

Rui Zhou and Daniel P. Palomar

References

Examples

library(highOrderPortfolios)
data(X50)

# estimate moments
X_moments <- estimate_sample_moments(X50[, 1:10])

# decide problem setting
w0 <- rep(1/10, 10)
w0_moments <- eval_portfolio_moments(w0, X_moments)
d <- abs(w0_moments)
kappa <- 0.3 * sqrt(w0 %*% X_moments$Sigma %*% w0)

# portfolio optimization
sol <- design_MVSKtilting_portfolio_via_sample_moments(d, X_moments, w_init = w0, w0 = w0,
w0_moments = w0_moments, kappa = kappa)

---

design_MVSK_portfolio_via_sample_moments

Design high-order portfolio based on weighted linear combination of first four moments

Description

Design high-order portfolio based on weighted linear combination of first four moments (i.e., mean, variance, skewness, and kurtosis):

\[
\begin{align*}
\text{minimize} & \quad - \lambda_1 w^\top \mu + \lambda_2 w^\top \Sigma w + \lambda_3 w^\top \Phi w w + \lambda_4 w^\top \Psi w w w \\
\text{subject to} & \quad ||w||_1 \leq \text{leverage}, \sum(w) = 1.
\end{align*}
\]

Usage

design_MVSK_portfolio_via_sample_moments(
  lmd = rep(1, 4),
  X_moments,
  w_init = rep(1/length(X_moments$mu), length(X_moments$mu)),
  leverage = 1,
  method = c("Q-MVSK", "MM", "DC"),
  tau_w = 0,
  gamma = 1,
  zeta = 1e-08,
  maxiter = 100,
  ftol = 1e-05,
  wtol = 1e-04,
stopval = -Inf
}

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lmd</td>
<td>Numerical vector of length 4 indicating the weights of first four moments.</td>
</tr>
<tr>
<td>X_moments</td>
<td>List of moment parameters, see <code>estimate_sample_moments()</code></td>
</tr>
<tr>
<td>w_init</td>
<td>Numerical vector indicating the initial value of portfolio weights.</td>
</tr>
<tr>
<td>leverage</td>
<td>Number (&gt;= 1) indicating the leverage of portfolio.</td>
</tr>
<tr>
<td>method</td>
<td>String indicating the algorithm method, must be one of: &quot;Q-MVSK&quot;, &quot;MM&quot;, &quot;DC&quot;</td>
</tr>
<tr>
<td>tau_w</td>
<td>Number (&gt;= 0) guaranteeing the strong convexity of approximating function.</td>
</tr>
<tr>
<td>gamma</td>
<td>Number (0 &lt; gamma &lt;= 1) indicating the initial value of gamma.</td>
</tr>
<tr>
<td>zeta</td>
<td>Number (0 &lt; zeta &lt; 1) indicating the diminishing parameter of gamma.</td>
</tr>
<tr>
<td>maxiter</td>
<td>Positive integer setting the maximum iteration.</td>
</tr>
<tr>
<td>ftol</td>
<td>Positive number setting the convergence criterion of function objective.</td>
</tr>
<tr>
<td>wtol</td>
<td>Positive number setting the convergence criterion of portfolio weights.</td>
</tr>
<tr>
<td>stopval</td>
<td>Number setting the stop value of objective.</td>
</tr>
</tbody>
</table>

Value

A list containing the following elements:

- **w** Optimal portfolio vector.
- **cpu_time_vs_iterations** Time usage over iterations.
- **objfun_vs_iterations** Objective function over iterations.
- **iterations** Iterations index.
- **convergence** Boolean flag to indicate whether or not the optimization converged.
- **moments** Moments of portfolio return at optimal portfolio weights.

Author(s)

Rui Zhou and Daniel P. Palomar

References


Examples

```r
library(highOrderPortfolios)
data(X50)

# estimate moments
X_moments <- estimate_sample_moments(X50[, 1:10])

# decide moment weights
xi <- 10
lmd <- c(1, xi/2, xi*(xi+1)/6, xi*(xi+1)*(xi+2)/24)

# portfolio optimization
sol <- design_MVSK_portfolio_via_sample_moments(lmd, X_moments)
```

---

design_MVSK_portfolio_via_skew_t

*Design MVSK portfolio without shorting based on the parameters of generalized hyperbolic skew-t distribution*

**Description**

Design MVSK portfolio without shorting based on the parameters of generalized hyperbolic skew-t distribution:

\[
\begin{align*}
\text{minimize} & \quad - \lambda_1 \phi_1(w) + \lambda_2 \phi_2(w) \\
& \quad - \lambda_3 \phi_3(w) + \lambda_4 \phi_4(w) \\
\text{subject to} & \quad w \geq 0, \sum w = 1.
\end{align*}
\]

**Usage**

```r
design_MVSK_portfolio_via_skew_t(
  lambda,
  X_skew_t_params,
  w_init = rep(1/length(X_skew_t_params$mu), length(X_skew_t_params$mu)),
  method = c("LMVSK", "DC", "Q-MVSK", "SQUAREM", "RFPA", "PGD"),
  gamma = 1,
  zeta = 1e-08,
  tau_w = 0,
  beta = 0.5,
  tau = 1e+05,
  initial_eta = 5,
  maxiter = 1000,
  ftol = 1e-06,
  wtol = 1e-06,
  stopval = -Inf
)
```

Arguments

lambda  Numerical vector of length 4 indicating the weights of first four moments.
X_skew_t_params  List of fitted parameters, including location vector, skewness vector, scatter matrix, and the degree of freedom, see estimate_skew_t().
w_init  Numerical vector indicating the initial value of portfolio weights.
method  String indicating the algorithm method, must be one of: "L-MVSK", "DC", "Q-MVSK", "SQUAREM", "RFPA", "PGD".
gamma  Number (0 < gamma <= 1) indicating the initial value of gamma for the Q-MVSK method.
zeta  Number (0 < zeta < 1) indicating the diminishing parameter of gamma for the Q-MVSK method.
tau_w  Number (>= 0) guaranteeing the strong convexity of approximating function.
beta  Number (0 < beta < 1) decreasing the step size of the projected gradient methods.
tau  Number (tau > 0) hyper-parameters for the fixed-point acceleration.
initial_eta  Initial eta for projected gradient methods
maxiter  Positive integer setting the maximum iteration.
ftol  Positive number setting the convergence criterion of function objective.
wtol  Positive number setting the convergence criterion of portfolio weights.
stopval  Number setting the stop value of objective.

Value

A list containing the following elements:

w  Optimal portfolio vector.
cpu_time_vs_iterations  Time usage over iterations.
objfun_vs_iterations  Objective function over iterations.
iterations  Iterations index.
convergence  Boolean flag to indicate whether or not the optimization converged.
moments  Moments of portfolio return at optimal portfolio weights.

Author(s)

Xiwen Wang, Rui Zhou and Daniel P. Palomar

References

Examples

```r
library(highOrderPortfolios)
data(X50)

# estimate skew t distribution
X_skew_t_params <- estimate_skew_t(X50)

# decide moment weights
xi <- 10
lambda <- c(1, 4, 10, 20)

# portfolio optimization
sol <- design_MVSK_portfolio_via_skew_t(lambda, X_skew_t_params, method = "RFPA", tau = 10)
```

---

estimate_sample_moments

*Estimate first four moment parameters of multivariate observations*

Description

Estimate first four moments of multivariate observations, namely, mean vector, covariance matrix, coskewness matrix, and cokurtosis matrix.

Usage

```r
estimate_sample_moments(X, adjust_magnitude = FALSE)
```

Arguments

- `X`: Data matrix.
- `adjust_magnitude`: Boolean indicating whether to adjust the order of magnitude of parameters.

Note: this is specially designed for the function `design_MVSKtilting_portfolio_via_sample_moments`

Value

A list containing the following elements:

- `mu`: Mean vector.
- `Sgm`: Covariance matrix.
- `Phi_mat`: Co-skewness matrix.
- `Psi_mat`: Co-kurtosis matrix.
- `Phi`: Co-skewness matrix in vector form (collecting only the unique elements).
- `Psi`: Co-kurtosis matrix in vector form (collecting only the unique elements).
- `Phi_shred`: Partition on Phi (see reference).
- `Psi_shred`: Partition on Psi (see reference).
Author(s)
Rui Zhou and Daniel P. Palomar

References

Examples

library(highOrderPortfolios)
data(X50)

X_moments <- estimate_sample_moments(X50[, 1:10])

---

estimate_skew_t

Estimate the parameters of skew-t distribution from multivariate observations

Description
Using the package fitHeavyTail to estimate the parameters of ghMST distribution from multivariate observations, namely, location vector (mu), skewness vector (gamma), scatter matrix (scatter), degree of freedom (nu), parameters a, and the Cholesky decomposition of the scatter matrix (chol_Sigma).

Usage

estimate_skew_t(
  X,
  initial = NULL,
  nu_lb = 9,
  max_iter = 100,
  ptol = 0.001,
  ftol = Inf,
  PXEM = TRUE,
  return_iterates = FALSE,
  verbose = FALSE
)
**estimate_skew_t**

**Arguments**

- **X**  
  Data matrix containing the multivariate time series (each column is one time series).

- **initial**  
  List of initial values of the parameters for the iterative estimation method. Possible elements include:
  - **nu**: default is 4,
  - **mu**: default is the data sample mean,
  - **gamma**: default is the sample skewness vector,
  - **scatter**: default follows from the scaled sample covariance matrix,

- **nu_lb**  
  Minimum value for the degree of freedom to maintain the existence of high-order moments (default is 9).

- **max_iter**  
  Integer indicating the maximum number of iterations for the iterative estimation method (default is 100).

- **ptol**  
  Positive number indicating the relative tolerance for the change of the variables to determine convergence of the iterative method (default is $1e^{-3}$).

- **ftol**  
  Positive number indicating the relative tolerance for the change of the log-likelihood value to determine convergence of the iterative method (default is Inf, so it is not active). Note that using this argument might have a computational cost as a convergence criterion due to the computation of the log-likelihood (especially when X is high-dimensional).

- **PXEM**  
  Logical value indicating whether to use the parameter expansion (PX) EM method to accelerating the convergence.

- **return_iterates**  
  Logical value indicating whether to record the values of the parameters (and possibly the log-likelihood if ftol < Inf) at each iteration (default is FALSE).

- **verbose**  
  Logical value indicating whether to allow the function to print messages (default is FALSE).

**Value**

A list containing the following elements:

- **mu**  
  Location vector estimate (not the mean).

- **gamma**  
  Skewness vector estimate.

- **scatter**  
  Scatter matrix estimate.

- **nu**  
  Degrees of freedom estimate.

- **chol_Sigma**  
  Choleski decomposition of the Scatter matrix estimate.

- **a**  
  A list of coefficients useful for later computation

**Author(s)**

Xiwen Wang, Rui Zhou, and Daniel P. Palomar
References


Examples

library(highOrderPortfolios)
data("X50")
X_skew_t_params <- estimate_skew_t(X50)

eval_portfolio_moments

Evaluate first four moments of a given portfolio

Description

Evaluate first four moments of a given portfolio’s return, namely, mean, variance, skewness, and kurtosis.

Usage

eval_portfolio_moments(w, X_statistics)

Arguments

w Numerical vector with portfolio weights.

X_statistics Argument characterizing the constituents assets. Either the sample parameters as obtained by function estimate_sample_moments() or the multivariate skew t parameters as obtained by function estimate_skew_t().

Value

Four moments of the given portfolio.

Author(s)

Rui Zhou, Xiwen Wang, and Daniel P. Palomar

References


Examples

```r
library(highOrderPortfolios)
data(X50)

# nonparametric case
X_moments <- estimate_sample_moments(X50[, 1:10])
w_moments <- eval_portfolio_moments(w = rep(1/10, 10), X_statistics = X_moments)

# parametric case (based on the multivariate skew t distribution)
X_skew_t_params <- estimate_skew_t(X50[, 1:10])
w_moments <- eval_portfolio_moments(w = rep(1/10, 10), X_statistics = X_skew_t_params)
```

---

**X100**

*Synthetic 500x100 matrix dataset*

**Description**

Synthetic 500x100 matrix dataset containing 500 realizations of 100 variables.

**Usage**

```r
data(X100)
```

**Format**

An object of class `xts` (inherits from `zoo`) with 500 rows and 100 columns.

---

**X200**

*Synthetic 1000x200 matrix dataset*

**Description**

Synthetic 1000x200 matrix dataset containing 1000 realizations of 200 variables.

**Usage**

```r
data(X200)
```

**Format**

An object of class `xts` (inherits from `zoo`) with 1000 rows and 100 columns.
**X50**

*Synthetic 250x50 matrix dataset*

---

**Description**

Synthetic 250x50 matrix dataset containing 250 realizations of 50 variables.

**Usage**

data(X50)

**Format**

An object of class `matrix` (inherits from `array`) with 250 rows and 50 columns.
Index

* dataset
  - X100, 13
  - X200, 13
  - X50, 14

design_MVSK_portfolio_via_sample_moments, 2, 5

design_MVSK_portfolio_via_skew_t, 2, 7

design_MVSKtilting_portfolio_via_sample_moments, 2, 3, 9

estimate_sample_moments, 4, 6, 9, /2

estimate_skew_t, 8, 10, /2

eval_portfolio_moments, 12

highOrderPortfolios-package, 2

X100, 13
X200, 13
X50, 14