Package ‘CombinePortfolio’

May 7, 2026

Type Package

Title Estimation of Optimal Portfolio Weights by Combining Simple Portfolio Strategies

Version 0.4

Date 2019-02-10

Author Florian Ziel

Maintainer Florian Ziel <florian.ziel@uni-due.de>

Description Estimation of optimal portfolio weights as combination of simple portfolio strategies, like the tangency, global minimum variance (GMV) or naive (1/N) portfolio. It is based on a utility maximizing 8-fund rule. Popular special cases like the Kan-Zhou(2007) 2-fund and 3-fund rule or the Tu-Zhou(2011) estimator are nested.

Depends R (>= 3.0.2)

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-10 21:43:28 UTC

Contents

CombinePortfolio-package *Estimation of optimal combined portfolios based on an 8-fund rule.*

Description

This package computes optimal portfolio weights as combination of simple portfolio strategies, like the tangency, GMV or naive (1/N). It is based on an 8-fund rule.

Details

Package:	CombinePortfolio
Type:	Package
Version:	1.0
Date:	2016-06-01
License:	GPL-3
Depends:	R (>= 3.0), methods
URL:	http://www.cran.r-project.org, http://www.bioconductor.org, http://www.statomics.com

Author(s)

Code: Florian Ziel
Documentation: Florian Ziel
Maintainer: Florian Ziel <florian.ziel@uni-due.de>

References

(list of references)

Examples

```
ret<- diff(log(EuStockMarkets)) ## sample asset returns
crule<- combination.rule(ret,detailed.output=TRUE)
crule$w["1'",] ## Adjusted Kan-Zhou(2007) 2-fund rule
crule$w["1''2",] ## Adjusted Kan-Zhou(2007) 3-fund rule
crule$w["124",] ## Combination rule: Tangency+GMV+naive 4-fund rule, plug-in estimator
crule$delta["124",] ## Combination weights
crule$V[,c(1,2,4)] ## Combination targets: Tangency, GMV and naive (1/N)
```

`combination.rule` *Function for estimating portfolio weights by the 8fund rule*

Description

This function computes optimal portfolio weights based on an 8-fund rule.

Usage

```
combination.rule(ret, gamma=1, superset=1:7, subset=NULL, detailed.output=FALSE,
RHO.grid.size= 100, Kmax.init= 500, tail.cut.exp= 20)
```

Arguments

`ret`	Matrix or data.frame of excess returns
`gamma`	Relative risk aversion parameter
`superset`	Vector of integers from 1,2,...,7. It gives the possible included target rules, `1:7` provides all full 8-fund rule solutions.
`subset`	Vector of integers of subset. It gives the target rules that must be included in the model, `NULL` provides all possible solutions.
`detailed.output`	If `FALSE` only the estimated portfolio weight vectors of the models are returned. If `TRUE` a list of the portfolio weight vectors, the combination weights, and the target rules is provided.
`RHO.grid.size`	Just for convergence issues, the larger the more time-consuming, but the higher the precision of the results, only relevant if one of 5, 6 or 7 rule is included.
`Kmax.init`	See description of `RHO.grid.size`
`tail.cut.exp`	See description of `RHO.grid.size`

Details

The target vectors are scaled so that their weights sum up to 1. Thus target rules are interpretable, i.e. 1 = tancency, 2 = GMV and 4 = naive (1/N). The function computes optimal portfolio weights given any combination rule of the riskfree asset and several target rule. These rules are called (and ordered) by and proportional to

$$1 \equiv \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}$$

$$2 \equiv \widehat{\boldsymbol{\Sigma}}^{-1}\mathbf{1}$$

$$3 \equiv \widehat{\boldsymbol{\mu}}$$

$$4 \equiv \mathbf{1}$$

$$5 \equiv \widehat{\boldsymbol{S}}^{-2}\widehat{\boldsymbol{\mu}}$$

$$6 \equiv \widehat{\boldsymbol{S}}^{-2}\mathbf{1}$$

$$7 \equiv \widehat{\boldsymbol{S}}^{-1}\mathbf{1}$$

where $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ are the Gaussian ML-estimators of the asset mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. Moreover, we use the decomposition $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{S}}\widehat{\boldsymbol{R}}\widehat{\boldsymbol{S}}$ with $\widehat{\boldsymbol{R}}$ as sample correlation matrix and $\widehat{\boldsymbol{S}}$ as diagonal matrix with the sample standard deviations on the diagonal.

Value

Returns matrix of estimated weights for possible combination rules. If `detailed.output` is `TRUE` `TRUE` a list of the portfolio weight vectors, the combination weights, and the target rules is provided. The names of the combination rule are coded by their portfolio that is incorporated. If "’" is contained is the name θ^2-adjusted estimation is used, if "”" is contained is the name θ^2-adjusted estimation is used. Hence e.g. "1’" represents the θ^2-adjusted 2-fund rule of Kan-Zhou(2007) and "1”2" represents the ψ^2-adjusted 3-fund rule of Kan-Zhou(2007).

Author(s)

Florian Ziel
<florian.ziel@uni-due.de>

See Also

combination.rule

Examples

```
ret<- diff(log(EuStockMarkets))

combination.rule(ret) ## all 8-fund rule estimates

crule<- combination.rule(ret,gamma=5,detailed.output=TRUE)
crule$w["1'",] ## Adjusted Kan-Zhou(2007) 2-fund rule
crule$w["1''2",] ## Adjusted Kan-Zhou(2007) 3-fund rule
crule$w["124",] ## Combination rule: Tangency+GMV+naive 4-fund rule, plug-in estimator
crule$delta["124",] ## Combination weights
crule$V[,c(1,2,4)] ## Combination targets: Tangency, GMV and naive

## only models that can contain Tangency, GMV and naive, but must contain GMV
crule2<- combination.rule(ret, superset=c(1,2,4), subset=2, detailed.output=TRUE)
crule2$w # weights
crule2$delta # combination weights
crule2$V # target vectors

## case where T <= N - 4
ret2<- cbind(ret[1:10,], ret[11:20,], ret[21:30,]) ## (TxN) 10x12-matrix
combination.rule(ret2) ## only accessible solutions
```

combination.rule.restriction

Function for estimating portfolio weights of a restricted 8-fund rule

Description

This function computes optimal portfolio weights based on a restricted 8-fund rule.

Usage

```
combination.rule.restriction(ret,  HC, h0, rule, gamma=1, detailed.output=FALSE,
RHO.grid.size= 100, Kmax.init= 500, tail.cut.exp= 20)
```

Arguments

`ret`	Matrix or data.frame of excess returns
`HC`	Scaled restriction matrix
`h0`	Scaled restriction vector
`rule`	Vector of combination rule, subset of 1,2,... 7
`gamma`	Relative risk aversion parameter
`detailed.output`	If `FALSE` only the estimated portfolio weight vectors of the models are returned. If TRUE a list of the portfolio weight vectors, the combination weights, and the target rules is provided.
`RHO.grid.size`	Just for convergence issues, the larger the more time-consuming, but the higher the precision of the results, only relevant if one of 5, 6 or 7 rule is included.
`Kmax.init`	See description of `RHO.grid.size`
`tail.cut.exp`	See description of `RHO.grid.size`

Details

Note that only C=I is implemented. So HC = H.

Value

Returns matrix of estimated weights for possible combination rules. If `detailed.output` is `TRUE` `TRUE` a list of the portfolio weight vectors, the combination weights, and the target rules is provided.

Author(s)

Florian Ziel
<florian.ziel@uni-due.de>

See Also

`combination.rule`

Examples

```
##setting
ret<- diff(log(EuStockMarkets))
T<- dim(ret)[1]
N<- dim(ret)[2]
gamma<- 1
## Example Tu-Zhou(2011) on Markowitz portfolio
a1<- T/(T-N-2)
rule<- c(1,4) ## as. TZ on Tangency and naive  restriction index
HC<- array( c(c(gamma*a1,N ) ) , dim=c(length(rule), 1) )## C^{-1} H conditions...
h0<- c(1)
## plug-in estimator, theta^2-adjusted, psi^2-adjusted:
rcrule<-combination.rule.restriction(ret,rule=rule,HC=HC,h0=h0,gamma=gamma,detailed.output=TRUE)
rcrule

## compare with TZ:
we<- rep.int(1/N, N)
TT<- T
mu<- apply(ret, 2, mean)## exess return
Sigma<- cov(ret) * (TT-1)/TT
Sigma.inv<- solve(Sigma)
sharpe.squared<- as.numeric( tcrossprod(crossprod(mu, Sigma.inv),mu) )
Sigma.inv.unb<- Sigma.inv * (TT-N-2)/TT
w.Markowitz<- 1/gamma * crossprod(Sigma.inv.unb, mu) ##
weSigmawe<- as.numeric( tcrossprod(crossprod(we, Sigma),we) )
wemu<- crossprod(we,mu)
pi1<- as.numeric( weSigmawe - 2/gamma * wemu + 1/gamma^2 *sharpe.squared )
bb<- (TT-2)*(TT-N-2)/( (TT-N-1)*(TT-N-4) ) ##c1 in tu-zhou
pi2<- (bb-1) * sharpe.squared /gamma^2 + bb/gamma^2 * N/TT
pi3<- 0
delta.TZ.Markowitz<- (pi1 - pi3)/(pi1 + pi2 - 2*pi3)
w.TZ.Markowitz<- (1- delta.TZ.Markowitz)* we + delta.TZ.Markowitz * w.Markowitz
w.TZ.Markowitz
rcrule$w["r:14",]

## adjusted Tu-Zhou on Markowitz
ibeta<- function(x,a,b) pbeta(x,a,b) * beta(a,b) ## incomplete beta
sharpe.squared.adj<- ((TT-N-2)*sharpe.squared - N)/TT + 2*(sharpe.squared^(N/2)*
(1+ sharpe.squared)^(-(TT-2)/2))/TT/ibeta(sharpe.squared/(1+sharpe.squared),N/2,(TT-N)/2)
pi1.adj<- as.numeric( weSigmawe - 2/gamma * wemu + 1/gamma^2 *sharpe.squared.adj )
pi2.adj<- (bb-1) * sharpe.squared.adj /gamma^2 + bb/gamma^2 * N/TT
delta.TZ.Markowitz.adj<- (pi1.adj - pi3)/(pi1.adj + pi2.adj - 2*pi3)
w.TZ.Markowitz.adj<- (1- delta.TZ.Markowitz.adj)* we + delta.TZ.Markowitz.adj * w.Markowitz
w.TZ.Markowitz.adj
rcrule$w["r:1'4",]


## Example Tu-Zhou(2011) on Kan-Zhou(2007) 3-fund
cd<- combination.rule(ret, detailed.output=TRUE)[[2]]["1''2",1:2] ## KZ3fund combination weights
rule<- c(1,2,4) ## as. TZ on KZ3fund  restriction index
HC<- array( c(c(gamma,0, N*cd[1] ), c(0, gamma, N*cd[2] )) , dim=c(length(rule), 2) )
h0<- c(cd[1]/N, cd[2]/N)
```

```
combination.rule.restriction(ret, rule=rule, HC=HC, h0=h0)
```

Index