Package ‘PortfolioOptim’

February 7, 2019

Title Small/Large Sample Portfolio Optimization

Version 1.1.1

Description Two functions for financial portfolio optimization by linear programming are provided. One function implements Benders decomposition algorithm and can be used for very large data sets. The other, applicable for moderate sample sizes, finds optimal portfolio which has the smallest distance to a given benchmark portfolio.

Depends R (>= 3.3.0)

License GNU General Public License version 3

Encoding UTF-8

LazyData true

Author Andrzej Palczewski [aut, cre],
Aleksandra Dabrowska [ctb]

Maintainer Andrzej Palczewski <A.Palczewski@mimuw.edu.pl>

Imports Rsymphony

RoxygenNote 6.1.1

Suggests mvtnorm, Rglpk, testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-07 12:53:25 UTC

R topics documented:

BDportfolio_optim *Portfolio Optimization by Benders decomposition*

Description

BDportfolio_optim is a linear program for financial portfolio optimization. Portfolio risk is measured by one of the risk measures from the list c("CVAR", "DCVAR", "LSAD", "MAD"). Benders decomposition method is explored to enable optimization for very large returns samples ($\sim 10^6$).

The optimization problem is:
$\min F(\theta^T r)$
over
$\theta^T E(r) \geq portfolio\_return,$
$LB \leq \theta \leq UB,$
$Aconstr\, \theta \leq bconstr,$
where
F is a measure of risk;
r is a time series of returns of assets;
θ is a vector of portfolio weights.

Usage

```
BDportfolio_optim(dat, portfolio_return,
risk=c("CVAR", "DCVAR","LSAD","MAD"), alpha=0.95,
Aconstr=NULL, bconstr=NULL, LB=NULL, UB=NULL, maxiter=500,tol=1e-8)
```

Arguments

`dat` Time series of returns data; dat = cbind(rr, pk), where rr is an array (time series) of asset returns, for n returns and k assets it is an array with $\dim(rr) = (n, k)$, pk is a vector of length n containing probabilities of returns.

`portfolio_return` Target portfolio return.

`risk` Risk measure chosen for optimization; one of "CVAR", "DCVAR", "LSAD", "MAD", where "CVAR" – denotes Conditional Value-at-Risk (CVaR), "DCVAR" – denotes deviation CVaR, "LSAD" – denotes Lower Semi Absolute Deviation, "MAD" – denotes Mean Absolute Deviation.

`alpha` Value of alpha quantile used to compute portfolio VaR and CVaR; used also as quantile value for risk measures CVAR and DCVAR.

`Aconstr` Matrix defining additional constraints, $\dim(Aconstr) = (m, k)$, where k – number of assets, m – number of constraints.

`bconstr` Vector defining additional constraints, length $(bconstr) = m$.

`LB` Vector of length k, lower bounds of portfolio weights θ; warning: condition LB = NULL is equivalent to LB = rep(0, k) (lower bound zero).

`UB` Vector of length k, upper bounds for portfolio weights θ.

`maxiter`	Maximal number of iterations.
`tol`	Accuracy of computations, stopping rule.

Value

BDportfolio_optim returns a list with items:

`return_mean`	vector of asset returns mean values.
`mu`	realized portfolio return.
`theta`	portfolio weights.
`CVaR`	portfolio CVaR.
`VaR`	portfolio VaR.
`MAD`	portfolio MAD.
`risk`	portfolio risk measured by the risk measure chosen for optimization.
`new_portfolio_return`	modified target portfolio return; when the original target portfolio return is to high for the problem, the optimization problem is solved for new_portfolio_return as the target return.

References

Benders, J.F., Partitioning procedures for solving mixed-variables programming problems. Number. Math., 4 (1962), 238–252, reprinted in Computational Management Science 2 (2005), 3–19. DOI: 10.1007/s10287-004-0020-y.

Konno, H., Piecewise linear risk function and portfolio optimization, Journal of the Operations Research Society of Japan, 33 (1990), 139–156.

Konno, H., Yamazaki, H., Mean-absolute deviation portfolio optimization model and its application to Tokyo stock market. Management Science, 37 (1991), 519–531.

Konno, H., Waki, H., Yuuki, A., Portfolio optimization under lower partial risk measures, Asia-Pacific Financial Markets, 9 (2002), 127–140. DOI: 10.1023/A:1022238119491.

Kunzi-Bay, A., Mayer, J., Computational aspects of minimizing conditional value at risk. Computational Management Science, 3 (2006), 3–27. DOI: 10.1007/s10287-005-0042-0.

Rockafellar, R.T., Uryasev, S., Optimization of conditional value-at-risk. Journal of Risk, 2 (2000), 21–41. DOI: 10.21314/JOR.2000.038.

Rockafellar, R. T., Uryasev, S., Zabarankin, M., Generalized deviations in risk analysis. Finance and Stochastics, 10 (2006), 51–74. DOI: 10.1007/s00780-005-0165-8.

Examples

```
library (Rsymphony)
library(Rglpk)
library(mvtnorm)
k = 3
num =100
dat <-  cbind(rmvnorm (n=num, mean = rep(0,k), sigma=diag(k)), matrix(1/num,num,1))
# a data sample with num rows and (k+1) columns for k assets;
port_ret = 0.05 # target portfolio return
```

```
alpha_optim = 0.95

# minimal constraints set: \eqn{\sum \theta_{i} = 1}
# has to be in two inequalities: \eqn{1 - \epsilon <= \sum \theta_{i} <= 1 + \epsilon}
a0 <- rep(1,k)
Aconstr <- rbind(a0,-a0)
bconstr <- c(1+1e-8, -1+1e-8)

LB <- rep(0,k)
UB <- rep(1,k)

res <- BDportfolio_optim(dat, port_ret, "CVAR", alpha_optim,
Aconstr, bconstr, LB, UB, maxiter=200, tol=1e-8)

cat ( c("Benders decomposition portfolio:\n\n"))
cat(c("weights \n"))
print(res$theta)

cat(c("\n mean = ", res$mu, " risk  = ", res$risk,
"\n CVaR = ", res$CVaR, " VaR = ", res$VaR, "\n MAD = ", res$MAD, "\n\n"))
```

PortfolioOptimProjection

Portfolio optimization which finds an optimal portfolio with the smallest distance to a benchmark.

Description

PortfolioOptimProjection is a linear program for financial portfolio optimization. The function finds an optimal portfolio which has the smallest distance to a benchmark portfolio given by bvec. Solution is by the algorithm due to Zhao and Li modified to account for the fact that the benchmark portfolio bvec has the dimension of portfolio weights and the solved linear program has a much higher dimension since the solution vector to the LP problem consists of a set of primal variables: financial portfolio weights, auxiliary variables coming from the reduction of the mean-risk problem to a linear program and also a set of dual variables depending on the number of constrains in the primal problem (see Palczewski).

Usage

```
PortfolioOptimProjection (dat, portfolio_return,
risk=c("CVAR","DCVAR","LSAD","MAD"), alpha=0.95, bvec,
Aconstr=NULL, bconstr=NULL, LB=NULL, UB=NULL, maxiter=500, tol=1e-7)
```

Arguments

`dat` Time series of returns data; dat = cbind(rr, pk), where rr is an array (time series) of asset returns, for n returns and k assets it is an array with $\dim(rr) = (n, k)$, pk is a vector of length n containing probabilities of returns.

`portfolio_return`
: Target portfolio return.

`risk`
: Risk measure chosen for optimization; one of "CVAR", "DCVAR", "LSAD", "MAD", where "CVAR" – denotes Conditional Value-at-Risk (CVaR), "DCVAR" – denotes deviation CVaR, "LSAD" – denotes Lower Semi Absolute Deviation, "MAD" – denotes Mean Absolute Deviation.

`alpha`
: Value of alpha quantile used to compute portfolio VaR and CVaR; used also as quantile value for risk measures CVAR and DCVAR.

`bvec`
: Benchmark portfolio, a vector of length k; function `PortfolioOptimProjection` finds an optimal portfolio with the smallest distance to `bvec`.

`Aconstr`
: Matrix defining additional constraints, $\dim(Aconstr) = (m, k)$, where k – number of assets, m – number of constraints.

`bconstr`
: Vector defining additional constraints, length $(bconstr) = m$.

`LB`
: Vector of length k, lower bounds of portfolio weights θ; warning: condition LB = NULL is equivalent to LB = rep(0, k) (lower bound zero).

`UB`
: Vector of length k, upper bounds for portfolio weights θ.

`maxiter`
: Maximal number of iterations.

`tol`
: Accuracy of computations, stopping rule.

Value

PortfolioOptimProjection returns a list with items:

`return_mean` vector of asset returns mean values.
`mu` realized portfolio return.
`theta` portfolio weights.
`CVaR` portfolio CVaR.
`VaR` portfolio VaR.
`MAD` portfolio MAD.
`risk` portfolio risk measured by the risk measure chosen for optimization.
`new_portfolio_return` modified target portfolio return; when the original target portfolio return is to high for the problem, the optimization problem is solved for new_portfolio_return as the target return.

References

Palczewski, A., LP Algorithms for Portfolio Optimization: The PortfolioOptim Package, R Journal, 10(1) (2018), 308–327. DOI:10.32614/RJ-2018-028.

Zhao, Y-B., Li, D., Locating the least 2-norm solution of linear programs via a path-following method, SIAM Journal on Optimization, 12 (2002), 893–912. DOI:10.1137/S1052623401386368.

Examples

```
library(mvtnorm)
```

```
k = 3
num =100
dat <-  cbind(rmvnorm (n=num, mean = rep(0,k), sigma=diag(k)), matrix(1/num,num,1))
# a data sample with num rows and (k+1) columns for k assets;
w_m <- rep(1/k,k) # benchmark portfolio, a vector of length k,
port_ret = 0.05 # portfolio target return
alpha_optim = 0.95

# minimal constraints set: \sum theta_i = 1
# has to be in two inequalities: 1 - \epsilon <= \sum theta_i <= 1 +\epsilon
a0 <- rep(1,k)
Aconstr <- rbind(a0,-a0)
bconstr <- c(1+1e-8, -1+1e-8)

LB <- rep(0,k)
UB <- rep(1,k)

res <- PortfolioOptimProjection(dat, port_ret, risk="MAD",
alpha=alpha_optim, w_m, Aconstr, bconstr, LB, UB, maxiter=200, tol=1e-7)

cat ( c("Projection optimal portfolio:\n\n"))
cat(c("weights \n"))
print(res$theta)


cat (c ("\n mean = ", res$mu, " risk  = ", res$risk,    "\n CVaR = ", res$CVaR, " VaR = ",
res$VaR, "\n MAD = ", res$MAD,  "\n\n"))
```

Index