Using the ROI solvers with PortfolioAnalytics

Ross Bennett

August 17, 2013

Abstract

The purpose of this vignette is to demonstrate a sample of the optimzation problems that can be solved in PortfolioAnalytics with the ROI solvers. See `demo(demo_ROI)` for a more complete set of examples.

Contents

1 Getting Started

1.1 Load Packages

Load the necessary packages.

```
suppressMessages(library(PortfolioAnalytics))

## Warning:  package 'PerformanceAnalytics' was built under R version 2.15.2

suppressMessages(library(Rglpk))

## Warning:  package 'slam' was built under R version 2.15.2

suppressMessages(library(foreach))
suppressMessages(library(iterators))
suppressMessages(library(ROI))
suppressMessages(require(ROI.plugin.glpk))
suppressMessages(require(ROI.plugin.quadprog))
```

1.2 Data

The edhec data set from the PerformanceAnalytics package will be used as example data.

```
data(edhec)

# Use the first 4 columns in edhec for a returns object
returns <- edhec[, 1:4]
print(head(returns, 5))

##            Convertible Arbitrage CTA Global Distressed Securities
## 1997-01-31                0.0119     0.0393                0.0178
```

```
## 1997-02-28                    0.0123     0.0298                     0.0122
## 1997-03-31                    0.0078    -0.0021                    -0.0012
## 1997-04-30                    0.0086    -0.0170                     0.0030
## 1997-05-31                    0.0156    -0.0015                     0.0233
##            Emerging Markets
## 1997-01-31           0.0791
## 1997-02-28           0.0525
## 1997-03-31          -0.0120
## 1997-04-30           0.0119
## 1997-05-31           0.0315


# Get a character vector of the fund names
funds <- colnames(returns)
```

2 Maximizing Mean Return

The objective to maximize mean return is a linear problem of the form:

$$\underset{\boldsymbol{w}}{\text{maximize}} \quad \hat{\boldsymbol{\mu}}'\boldsymbol{w}$$

Where $\hat{\boldsymbol{\mu}}$ is the estimated mean asset returns and $\boldsymbol{w}$ is the set of weights. Because this is a linear problem, it is well suited to be solved using a linear programming solver. For these types of problems, PortfolioAnalytics uses the ROI package with the glpk plugin.

2.1 Portfolio Object

The first step is to create the portfolio object. Then add constraints and a return objective.

```
# Create portfolio object
portf_maxret <- portfolio.spec(assets = funds)

# Add constraints to the portfolio object
portf_maxret <- add.constraint(portfolio = portf_maxret, type = "full_investment")
portf_maxret <- add.constraint(portfolio = portf_maxret, type = "box", min = c(0.02
    0.05, 0.03, 0.02), max = c(0.55, 0.6, 0.65, 0.5))

# Add objective to the portfolio object
portf_maxret <- add.objective(portfolio = portf_maxret, type = "return", name = "mea
```

The print method for the portfolio object shows a high level overview while the summary method shows much more detail of the assets, constraints, and objectives that are specified in the portfolio object.

```
print(portf_maxret)

## **************************************************
## PortfolioAnalytics Portfolio Specification
## **************************************************
##
## Call:
## portfolio.spec(assets = funds)
##
## Assets
## Number of assets: 4
##
## Constraints
## Number of constraints: 2
## Number of enabled constraints: 2
## Enabled constraint types
##  - full_investment
##  - box
## Number of disabled constraints: 0
##
## Objectives
## Number of objectives: 1
## Number of enabled objectives: 1
## Enabled objective names
##  - mean
## Number of disabled objectives: 0

summary(portf_maxret)

## **************************************************
## PortfolioAnalytics Portfolio Specification Summary
## **************************************************
## Assets and Seed Weights:
## Convertible Arbitrage           CTA Global Distressed Securities
##                  0.25                 0.25                 0.25
##      Emerging Markets
```

```
##                  0.25
##
## Constraints:
##
## ****************************************
## full_investment constraint
## ****************************************
## $type
## [1] "full_investment"
##
## $enabled
## [1] TRUE
##
## $message
## [1] FALSE
##
## $min_sum
## [1] 1
##
## $max_sum
## [1] 1
##
## $call
## add.constraint(portfolio = portf_maxret, type = "full_investment")
##
## attr(,"class")
## [1] "weight_sum_constraint" "constraint"
##
##
## ****************************************
## box constraint
## ****************************************
## $type
## [1] "box"
##
## $enabled
## [1] TRUE
##
## $min
```

```
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.02                  0.05                  0.03
##      Emerging Markets
##                  0.02
##
## $max
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.55                  0.60                  0.65
##      Emerging Markets
##                  0.50
##
## $call
## add.constraint(portfolio = portf_maxret, type = "box", min = c(0.02,
##     0.05, 0.03, 0.02), max = c(0.55, 0.6, 0.65, 0.5))
##
## attr(,"class")
## [1] "box_constraint" "constraint"
##
##
## Objectives:
##
## ****************************************
## return_objective
## ****************************************
## $name
## [1] "mean"
##
## $target
## NULL
##
## $arguments
## list()
##
## $enabled
## [1] TRUE
##
## $multiplier
## [1] -1
##
```

```
## $call
## add.objective(portfolio = portf_maxret, type = "return", name = "mean")
##
## attr(,"class")
## [1] "return_objective" "objective"
```

2.2 Optimization

The next step is to run the optimization. Note that `optimize_method="ROI"` is specified in the call to `optimize.portfolio` to select the solver used for the optimization.

```
# Run the optimization
opt_maxret <- optimize.portfolio(R = returns, portfolio = portf_maxret, optimize_met
```

The print method for the `opt_maxret` object shows the call, optimal weights, and the objective measure

```
print(opt_maxret)

## ***********************************
## PortfolioAnalytics Optimization
## ***********************************
##
## Call:
## optimize.portfolio(R = returns, portfolio = portf_maxret, optimize_method = "ROI'
##
## Optimal Weights:
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.02                  0.05                  0.43
##      Emerging Markets
##                  0.50
##
## Objective Measure:
## [1] -0.008
```

The sumary method for the `opt_maxret` object shows details of the object with constraints, objectives, and other portfolio statistics.

```r
summary(opt_maxret)

## **************************************************
## PortfolioAnalytics Optimization Summary
## **************************************************
##
## Call:
## optimize.portfolio(R = returns, portfolio = portf_maxret, optimize_method = "ROI"
##
## Optimal Weights:
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.02                  0.05                  0.43
##      Emerging Markets
##                  0.50
##
## Objective Measures:
## [1] -0.007996
##
## Portfolio Assets and Seed Weights:
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.25                  0.25                  0.25
##      Emerging Markets
##                  0.25
##
## **************************************************
## PortfolioAnalytics Portfolio Specification
## **************************************************
##
## Call:
## portfolio.spec(assets = funds)
##
## Assets
## Number of assets: 4
##
## Constraints
## Number of constraints: 2
## Number of enabled constraints: 2
## Enabled constraint types
##  - full_investment
##  - box
```

```
## Number of disabled constraints: 0
##
## Objectives
## Number of objectives: 1
## Number of enabled objectives: 1
## Enabled objective names
##  - mean
## Number of disabled objectives: 0
##
## ****************************************
## Constraints
## ****************************************
## Leverage Constraint:
## min_sum = 1
## max_sum = 1
##
## Box Constraints:
## min:
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.02                  0.05                  0.03
##      Emerging Markets
##                  0.02
## max:
## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.55                  0.60                  0.65
##      Emerging Markets
##                  0.50
##
## Group Constraints:
## Position Limit Constraints:
## Maximum number of non-zero weights, max_pos:
## NULL
## Realized number of non-zero weights (i.e. positions):
## [1] 4
##
## Maximum number of long positions, max_pos_long:
## NULL
## Realized number of long positions:
## [1] 4
```

```
##
## Maximum number of short positions, max_pos_short:
## NULL
## Realized number of short positions:
## [1] 0
##
##
## Diversification Target Constraint:
## NULL
##
## Realized diversification:
## [1] 0.5622
##
## Turnover Target Constraint:
## NULL
##
## Realized turnover from seed weights:
## [1] 0.215
##
## Factor Exposure Constraints:
## ****************************************
## Objectives
## ****************************************
##
## Objective: return_objective
## $name
## [1] "mean"
##
## $target
## NULL
##
## $arguments
## list()
##
## $enabled
## [1] TRUE
##
## $multiplier
## [1] -1
```

```
##
## $call
## add.objective(portfolio = portf_maxret, type = "return", name = "mean")
##
## attr(,"class")
## [1] "return_objective" "objective"
##
## ****************************************
##
## Elapsed Time:
## Time difference of 0.01058 secs
```

The `opt_maxret` object is of class `optimize.portfolio.ROI` and contains the following elements. Objects of class `optimize.portfolio.ROI` are S3 objects and elements can be accessed with the `$` operator.

```
names(opt_maxret)

## [1] "weights"      "out"          "call"         "portfolio"
## [5] "data_summary" "elapsed_time" "end_t"
```

The optimal weights and value of the objective function at the optimum can be accessed with the `extractStats` function.

```
extractStats(opt_maxret)

##                      out w.Convertible Arbitrage             w.CTA Global
##                -0.007996                0.020000                 0.050000
## w.Distressed Securities      w.Emerging Markets
##                 0.430000                0.500000
```

The optimal weights can be accessed with the `extractWeights` function.

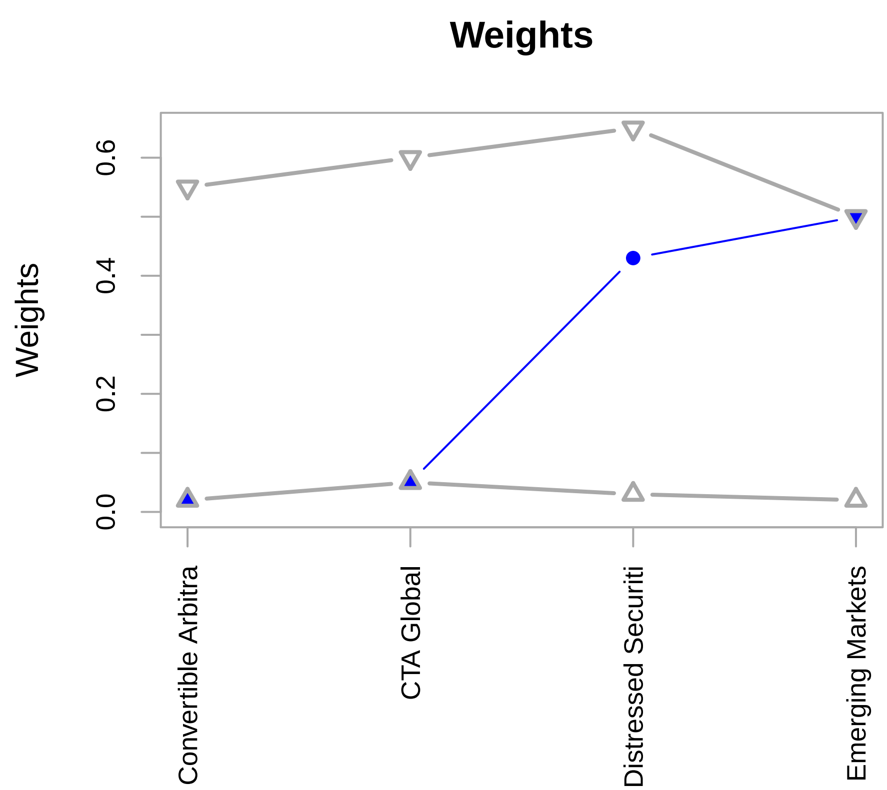
```
extractWeights(opt_maxret)

## Convertible Arbitrage            CTA Global Distressed Securities
##                  0.02                  0.05                  0.43
##      Emerging Markets
##                  0.50
```

2.3 Visualization

The chart of the optimal weights as well as the box constraints can be created with `chart.Weights.ROI`. The blue dots are the optimal weights and the gray triangles are the `min` and `max` of the box constraints.

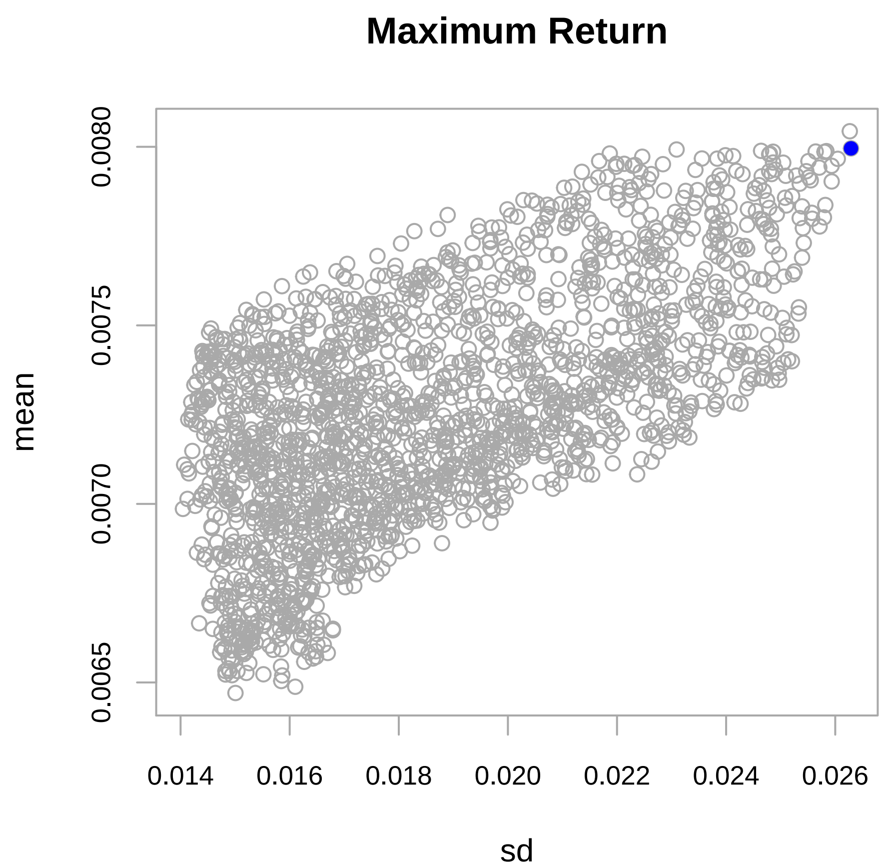
```
chart.Weights.ROI(opt_maxret)
```



The optimal portfolio can be plotted in risk-return space along with other feasible portfolios. The return metric is defined in the `return.col` argument and the risk metric is defined in the `risk.col` argument. The scatter chart includes the optimal portfolio (blue dot) and other feasible portfolios (gray circles) to show the overall feasible space given the constraints. By default, if `rp` is not passed in, the feasible portfolios are generated with `random_portfolios` to satisfy the constraints of the portfolio object.

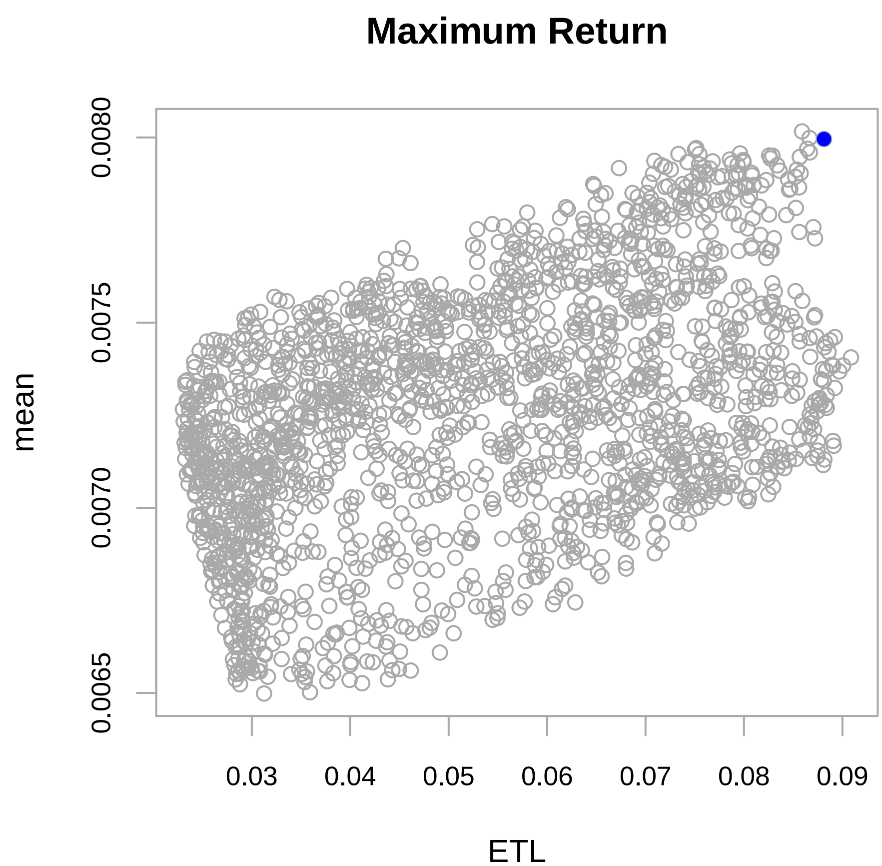
Volatility as the risk metric

```r
chart.Scatter.ROI(opt_maxret, R = returns, return.col = "mean", risk.col = "sd",
    main = "Maximum Return")
```



Expected tail loss as the risk metric

```r
chart.Scatter.ROI(opt_maxret, R = returns, return.col = "mean", risk.col = "ETL",
    main = "Maximum Return", invert = FALSE, p = 0.9)
```



2.4 Backtesting

An out of sample backtest is run with `optimize.portfolio.rebalancing`. In this example, an initial training period of 36 months is used and the portfolio is rebalanced quarterly.

```
bt_maxret <- optimize.portfolio.rebalancing(R = returns, portfolio = portf_maxret,
    optimize_method = "ROI", rebalance_on = "quarters", training_period = 36,
    trace = TRUE)

## Warning:  executing %dopar% sequentially:  no parallel backend registered
## overall elapsed time:0.354373931884766
```

The `bt_maxret` object is a list containing the optimal weights and objective measure at each rebalance period.

3 Minimizing Portfolio Variance

The objective to minimize portfolio variance is a quadratic problem of the form:

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \boldsymbol{w}' \boldsymbol{\Sigma} \boldsymbol{w}$$

Where $\boldsymbol{\Sigma}$ is the estimated covariance matrix of asset returns and $\boldsymbol{w}$ is the set of weights. Because this is a quadratic problem, it is well suited to be solved using a quadratic programming solver. For these types of problems, PortfolioAnalytics uses the ROI package with the quadprog plugin.

3.1 Global Minimum Variance Portfolio

3.1.1 Portfolio Object

```
# Create portfolio object
portf_minvar <- portfolio.spec(assets = funds)

# Add full investment constraint to the portfolio object
portf_minvar <- add.constraint(portfolio = portf_minvar, type = "full_investment")

# Add objective to minimize variance
portf_minvar <- add.objective(portfolio = portf_minvar, type = "risk", name = "var")
```

The only constraint specified is the full investment constraint, therefore the optimization problem is solving for the global minimum variance portfolio.

3.1.2 Optimization

```
# Run the optimization
opt_gmv <- optimize.portfolio(R = returns, portfolio = portf_minvar, optimize_metho
print(opt_gmv)

## ***********************************
## PortfolioAnalytics Optimization
## ***********************************
##
## Call:
## optimize.portfolio(R = returns, portfolio = portf_minvar, optimize_method = "ROI"
```

```
##
## Optimal Weights:
## Convertible Arbitrage            CTA Global Distressed Securities
##                 0.169                 0.301                 0.761
##      Emerging Markets
##                -0.231
##
## Objective Measure:
## [1] 0.000158
```

3.1.3 Backtesting

```
bt_gmv <- optimize.portfolio.rebalancing(R = returns, portfolio = portf_minvar,
    optimize_method = "ROI", rebalance_on = "quarters", training_period = 36)

## overall elapsed time:3.04221105575562
```

3.2 Constrained Minimum Variance Portfolio

3.2.1 Portfolio Object

Constraints can be added to the `portf_minvar` portfolio object previously created.

```
# Add long only constraints
portf_minvar <- add.constraint(portfolio = portf_minvar, type = "box", min = 0,
    max = 1)

# Add group constraints
portf_minvar <- add.constraint(portfolio = portf_minvar, type = "group", groups = c
    2, 1), group_min = c(0, 0.25, 0.1), group_max = c(0.45, 0.6, 0.5))
```

3.2.2 Optimization

```
# Run the optimization
opt_minvar <- optimize.portfolio(R = returns, portfolio = portf_minvar, optimize_met
print(opt_minvar)
```

```
## ***********************************
## PortfolioAnalytics Optimization
## ***********************************
##
## Call:
## optimize.portfolio(R = returns, portfolio = portf_minvar, optimize_method = "ROI"
##
## Optimal Weights:
## Convertible Arbitrage            CTA Global Distressed Securities
##                 0.300                 0.356                 0.244
##      Emerging Markets
##                 0.100
##
## Objective Measure:
## [1] 0.000228
```

3.2.3 Backtesting

```
bt_minvar <- optimize.portfolio.rebalancing(R = returns, portfolio = portf_minvar,
    optimize_method = "ROI", rebalance_on = "quarters", training_period = 36)

## overall elapsed time:2.97715401649475
```

4 Maximizing Quadratic Utility

The objective to maximize quadratic utility is a quadratic problem of the form:

$$\underset{\boldsymbol{w}}{\text{maximize}} \quad \boldsymbol{w}'\boldsymbol{\mu} - \frac{\lambda}{2}\boldsymbol{w}'\boldsymbol{\Sigma}\boldsymbol{w}$$

Where μ is the estimated mean asset returns, λ is the risk aversion parameter, $\boldsymbol{\Sigma}$ is the estimated covariance matrix of asset returns and $\boldsymbol{w}$ is the set of weights. Quadratic utility maximizes return while penalizing variance. The λ risk aversion parameter controls how much portfolio variance is penalized. Because this is a quadratic problem, it is well suited to be solved using a quadratic programming solver. For these types of problems, PortfolioAnalytics uses the ROI package with the quadprog plugin.

4.1 Portfolio Object

The portfolio object is specified, and constraints and objectives are created separately. The constraints and objectives are created separately as an alternative example and could also have been added directly to the portfolio object as in the previous sections.

```
# Create initial portfolio object
init_portf <- portfolio.spec(assets = funds)

# Create full investment constraint
fi_constr <- weight_sum_constraint(type = "full_investment")

# Create long only constraint
lo_constr <- box_constraint(type = "long_only", assets = init_portf$assets)

# Combine the constraints in a list
qu_constr <- list(fi_constr, lo_constr)

# Create return objective
ret_obj <- return_objective(name = "mean")

# Create variance objective specifying a risk_aversion parameter which
# controls how much the variance is penalized
var_obj <- portfolio_risk_objective(name = "var", risk_aversion = 0.25)

# Combine the objectives into a list
qu_obj <- list(ret_obj, var_obj)
```

4.2 Optimization

Note how the constraints and objectives are passed to optimize.portfolio.

```
# Run the optimization
opt_qu <- optimize.portfolio(R = returns, portfolio = init_portf, constraints = qu_c
    objectives = qu_obj, optimize_method = "ROI")
```

4.3 Backtesting

```
bt_qu <- optimize.portfolio.rebalancing(R = returns, portfolio = init_portf,
    constraints = qu_constr, objectives = qu_obj, optimize_method = "ROI", rebalance
    training_period = 36)

## overall elapsed time:3.02108693122864
```

5 Minimizing Expected Tail Loss

TODO