Vignette for `robRiskBudget` Function

Thomas K. Philips and Chindhanai Uthaisaad

August 28, 2017

Abstract

This vignette describes the use of the new simple and robust risk budgeting function `robRiskBudget` in the R package `factorAnalyticsAddon`, implemented during the Google Summer of Code (GSoC) 2017 *Advancing factorAnalytics* project. GSoC funding for this project was awarded to UW AMATH MS-CFRM student Chindhanai Uthaisaad, with Douglas Martin and Thomas Phillips as mentors.

1 Simple Robust Risk Budgeting with Expected Shortfall

In spite of the many complaints directed at it, the mean-variance paradigm has captured the hearts and minds of investment professionals for a good reason: it captures the essential aspects of the investor's problem in a parsimonious way, particularly when optimizing risk controlled portfolios without derivatives. The majority of complaints about the paradigm are centered around its robustness to errors in its inputs and its failure to account for tail risk, while the majority of proposed solutions address one or more of its deficiencies with some sacrifice of its simplicity and elegance. In this section, we describe the implementation of the Philips and Liu (2011) method for robust risk budgeting algorithm, a simple enhancement to the mean-variance paradigm that preserves its simplicity and its intuitive appeal, while accounting for risk in a more sophisticated way. In particular, Philips and Liu usefully extend a closed-form risk budgeting solution to allow the allocation of tracking error (or volatility) taking into account both tracking error and tail risk. Their methodology charts a pragmatic middle course between the simple and elegant (but tail risk insensitive) world of mean-variance optimization, and the analytically intractable and computationally complex (but tail-risk sensitive) world of most robust risk allocation algorithms.

In the absence of any constraints, mean-variance risk budgets are given by

$$\vec{\sigma} = \frac{C^{-1}}{\sqrt{(IR)^T C^{-1} IR}} \cdot \sigma_{Target}$$

where IR is the Information Ratio: $IR = \frac{E[r]}{\sigma}$ and C is the correlation matrix. In general, it is not obvious how one can allocate Expected Shortfall between a set of strategies in a way that achieves a target level of Expected Shortfall for the portfolio. The Philips and Liu algorithm starts by allocating risk using volatility as the measure of risk to obtain the above closed form solution, but then uses Expected Shortfall to modify its allocations in such a way that the Expected Shortfall of the overall portfolio is reduced. To do so, they first observe that the Information Ratio is the ratio of a measure of return (expected return) to a measure of risk (volatility), and then define a new measure of performance - the Modified Information Ratio - in terms of expected return and an alternative measure of risk (Expected Shortfall).

$$IR' = \frac{E[r]}{ES} = \frac{E[r]}{\sigma} \cdot \frac{\sigma}{ES} = \frac{IR}{{}^{ES}\!/_{\sigma}}$$

For distributions that are characterized by a scale parameter, the Expected Shortfall will be a constant multiple of the standard deviation, and the Modified Information Ratio is proportional to the Information Ratio. We call $\frac{ES}{\sigma}$ the Tail Risk Ratio. To bias risk budgets away from strategies with high Tail Risk Ratios, and towards strategies and securities with low tail risk ratios, they make an ad-hoc substitution and rewrite the expression for the risk budgets as follows:

$$\vec{\sigma} = \frac{C^{-1} IR'}{\sqrt{(IR')^T C^{-1} IR'}} \cdot \sigma_{Target}. \tag{1}$$

For distributions for which the Expected Shortfall is a constant multiple of the standard deviation (e.g. the Normal distribution), the constant cancels out from the numerator and denominator and leaves the risk budgets unchanged. So in this important special case, the correct answer is recovered. The primary limitation of this solution is the fact that not all of its components are guaranteed to be positive. That said, in many risk budgeting applications, the strategies employed are weakly correlated, and in these cases, the solution can be made robust by averaging the off-diagonal entries in the correlation matrix, so that $\bar{C}_{ij} = Average(C_{ij};\ i \neq j)$, and

$$\vec{\sigma} = \frac{\bar{C}^{-1} IR'}{\sqrt{(IR')^T \bar{C}^{-1} IR'}} \cdot \sigma_{Target}. \tag{2}$$

This simple closed form solution, with the inclusion of tail risk and with a stabilized correlation matrix, works surprisingly well in practice - it does not often result in negative solutions, and obviates the need for a long-only constraint - we just round up to 0 on the few occasions when a small negative risk budget appears. However, in the general case, if we want to bound σ_i between σ_i^U and σ_i^L, we either have to solve a full mean-variance optimization (and ignore tail risk) or create a simulation based historical optimization that includes it. Instead, we approximate the solution using an iterative scheme that clamps each risk budget between its applicable bounds, and uses a simple heuristic to increment or decrement all risk budgets in a way that converges in a few iterations.

1.1 Robust Risk Budgeting Procedure Description

Typically, the user will provide all the inputs needed for an optimization: prospective Information Ratios, volatilities, Expected Shortfalls and correlations. In this case, the optimization may be carried out immediately. At other times, only a subset of these inputs may be provided, along with a time series of returns for each strategy. In the second case, the missing parameters can be estimated from data, under the assumption that their future values will be similar to their past values over the estimation interval. To protect against error maximizaion, we shrink the historical estimates of the information ratios of the various strategies towards their grand mean using the James-Stein formula. This is made simple because each monthly estimate of the Information Ratio has a variance of 1.

1. Compute the optimal risk allocation to achieve the targeted volatility using 1 or 2.

   (a) Use the provided inputs (*IR* / *ES* / Volatility) to compute the robust risk budgets. In the event that the user does not provide estimates of the information ratio (or estimates of expected return and volatility which can be used to create estimates of the Information Ratio), compute the historical mean and volatility of each sequence using robust estimators of location and scale. Also estimate the expected shortfall of each strategy at the 95th percentile.

   (b) Take the ratio of the estimated excess return to the estimated volatility to compute a historical *IR*, and then shrink these *IR*'s using James - Stein shrinkage using a shrinkage factor of

   $$\delta = 1 - \frac{(k-3)}{\sum_i n_i(\bar{IR}_i - \bar{\bar{IR}})},$$

   where n_i is the number of observations for asset i. The shrunk *IR*'s are given by

   $$\bar{IR} \leftarrow \delta\ \bar{IR} + (1-\delta)\bar{\bar{IR}}.$$

(c) Compute the robust correlation matrix using a robust estimator such as the MCD estimator. If the strategies are weakly correlated, (say $|C_{ij}| < 0.2$ for at least $3/4$ of the off-diagonal elements), set all off-diagonal elements to their common average to further stabilize the matrix.

(d) Compute the the tail risk ratio of each strategy $TRR_i = ES_i/\sigma_i$, and the modified information ratio IR'_i by $IR'_i = IR_i/TRR_i$.

(e) Compute the unconstrained robust risk budgets $\vec{\sigma}$ using 1 or 2

2. Iteratively modify the various risk budgets till all the constraints are satisfied or the maximum number of iterations has been exceeded:

(a) Increase risk budgets that lie below the lower bound and lower risk budgets that lie above the upper bound till they hit their lower and upper bounds$\boldsymbol{\sigma}^L = (\sigma_1^L,\ \sigma_2^L,\ \ldots,\ \sigma_k^L)$ and $\boldsymbol{\sigma}^U = (\sigma_1^U,\ \sigma_2^U,\ \ldots,\ \sigma_k^U)$ respectively. Create sets of indicator variables $L = (L_1,\ L_2, \ldots,\ L_k)$ and $U = (U_1, U_2, \ldots,\ U_k)$ that identify the strategies whose risk budgets have been increased or decreased till they rest at a bound:

$$L_i = 1\ if\ \sigma_i = \sigma_1^L,\ L_i = 0.$$

and

$$U_i = 1\ if\ \sigma_i = \sigma_1^U,\ U_i = 0.$$

(b) Next, compute the current-to-target volatility ratio$d$ to determine if the portfolio's volatility is higher or lower than the target level volatility:

$$d = \frac{\boldsymbol{\sigma}_{new}^T C\ \boldsymbol{\sigma}_{new}}{\sigma_{Target}}.$$

If $d \in (1-\varepsilon, 1+\varepsilon)$, Stop. The portfolio satisfies all its constraints. Otherwise, further modify the risk budgets as follows

(c) If convergence is not achieved, use a Newton-like iteration to find a vector of risk budgets to add to the current risk budget so that the risk of the portfolio converges to its target.:

i. Determine the sets $U^{frozen} = \{i \in 1:k \mid U_i = 1\}$ and $L^{frozen} = \{i \in 1:k \mid L_i = 1\}$ of frozen indices of the risk budgets that cannot be increased or decreased any further, as they have already hit a bound.

ii. Calculate a set of scaled reciprocal distances $\mathbf{p} = (p_1,\ p_2,\ \ldots,\ p_k)$ between each risk budget and its initial values when our current portfolio's volatility is not within a factor of $1 \pm \varepsilon$ ofσ_{Target}.

$$p_i \leftarrow \begin{cases} \frac{1}{1+\frac{K}{\sigma_{Target}}(\sigma_{i,\ new}-\sigma_i)} & ;\ d > 1,\ i \notin L^{frozen}, \\ \frac{1}{1+\frac{K}{\sigma_{Target}}(\sigma_i-\sigma_{i,\ new})} & ;\ d < 1,\ i \notin U^{frozen}, \end{cases}$$

where K is a tuning constant (in practice we set K to 100). If the current value of a risk budget is very close to its initial unconstrained value, it must be that the bounds have not impacted it significantly and it can therefore be moved a fair bit. In this case, p_i must be close to 1 for any reasonable value of K. Likewise, if the current value of a risk budget is very far from its initial unconstrained value, it must be that the bounds have already impacted it significantly, and it should not be moved much further. In this case p_i must be close to 0 for any reasonable value of K.

iii. Solve for the constant m that matches the portfolio volatility after the addition of $m \cdot \mathbf{p}$ to σ_{Target}. It is not hard to show that m is given by the solution to a quadratic with the sign of the discriminant always being positive:

$$m = \frac{-(\mathbf{p}^T C\ \boldsymbol{\sigma}_{\mathbf{new}}) + \sqrt{(\mathbf{p}^T C\ \boldsymbol{\sigma}_{\mathbf{new}}) - (\mathbf{p}^T C\ \mathbf{p})(\boldsymbol{\sigma}_{new}^T C\ \boldsymbol{\sigma}_{new} - \sigma_{Target}^2)}}{(\mathbf{p}^T C\ \mathbf{p})}$$

iv. Update the risk budgets and return to step (a):

$$\boldsymbol{\sigma}_{new} \leftarrow \boldsymbol{\sigma}_{new} + m\mathbf{p}$$

1.2 R Code

The function `robRiskBudget` implements the Philips and Liu (2011) method to compute an optimal set of risk budgets. It takes into account both the volatility and the tail risk of strategies to create a portfolio with a targeted level of volatility but with a lower level of tail risk than achieved by a mean-variance risk budget. One of the option it provides is the option to average all off diagonal entries in the correlation matrix, and with this option in use, it works particularly well with weakly correlated strategies, as one often finds in multi-strategy hedge funds and absolute return portfolios.

The following is the R code implementation for the function `cusumActMgr()`.

```r
data(RussellData)
rf = RussellData[, 16]
robRiskData = RussellData[, 1:15]
riskBudgetObj = robRiskBudget(robRiskData, rf = rf, shrink = TRUE, avgCor = TRUE,
    ESMethod = "historical", corMatMethod = "mcd")
```

This `riskBudgetObj` is the `riskBudget` object in R. It contains the following elements

```r
names(riskBudgetObj)

##  [1] "initialRiskBudget"   "finalRiskBudget"     "iterativeRiskBudget"
##  [4] "excessReturns"       "corMat"              "avgCor"
##  [7] "targetVol"           "TE"                  "ES"
## [10] "modIR"
```

In this example, the initial risk budgets and the adjusted risk budgets are

```r
cbind(Initial = riskBudgetObj$initialRiskBudget,
        Final = riskBudgetObj$finalRiskBudget)

##                              Initial       Final
## Russell.3000            0.0073639025 0.003437997
## Russell.3000.Growth     0.0076993120 0.003773407
## Russell.3000.Value      0.0009869987 0.000000000
## Russell.1000            0.0053255427 0.001399638
## Russell.1000.Growth     0.0098534691 0.005927564
## Russell.1000.Value     -0.0011684619 0.000000000
## Russell.Midcap         -0.0147163795 0.000000000
## Russell.Midcap.Growth  -0.0036483365 0.000000000
## Russell.Midcap.Value   -0.0131450784 0.000000000
## Russell.2000            0.0033346322 0.000000000
## Russell.2000.Growth     0.0133787682 0.009452863
## Russell.2000.Value     -0.0002531944 0.000000000
## Russell.2500           -0.0049905712 0.000000000
## Russell.2500.Growth     0.0020455256 0.000000000
## Russell.2500.Value     -0.0095850735 0.000000000
```

To see how the initial risk budgets have changed, we use the function `chartRobRisk` and obtain the bar charts comparing the old and new risk budgets

```
chartRobRisk(riskBudgetObj)
```