Baruch MFE Capstone Report
Risk Budgeting Portfolio: Mathematical Theory, Numerical Techniques and Applications

Zichao (David) Wang

Instructor: Prof. Andrew Lesniewski

December 20, 2019

Note: A complete implementation of this report in Python can be found at https://github.com/davidwang1618/MTH9903_capstone_project.

1 Some notations

Below we define some notations which will be used throughout this report.

For any column vector v, v_i refers to its ith element. The index starts at 1 by default.

For any $(m \times n)$ matrix A, A_{ij} refers to the element in ith row and jth column. The row and column indices start at 1 by default.

- n: the number of assets in the universe; positive integer.
- x: dollar holding of each asset; $(n \times 1)$ vector. We assume that GMV $= 1$, i.e., $\sum_{i=1}^n x_i = 1$.
- b: normalized risk budget; $(n \times 1)$ vector. It follows that $\sum_{i=1}^n b_i = 1$.
- μ: estimated return of tickers; $(n \times 1)$ vector.
- Σ: estimated covariance matrix of tickers; $(n \times n)$ matrix.
- r: risk-free rate; scalar. Unless specified, we shall assume $r = 0$ throughout our research process.
- x_{RB}: risk-budgeting portfolio represented by dollar holding of each asset; $(n \times 1)$ vector.
- $\mathbb{1}$: a column vector whose elements are all 1.

2 Problem description

2.1 Measure of risk

Given a universe of n assets (typically equities), we can compute the expected return vector μ and estimated covariance matrix Σ. We have $\Sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$, where σ_i, σ_j are the volatility of asset i and j, and ρ_{ij} is the correlation of the returns of the two assets.

The risk measure is defined as

$$\mathcal{R}(x) = -x^\top(\mu - r) + c\,\sqrt{x^\top \Sigma x} \tag{1}$$

where c is a scalar measuring the trade-off between the expected return and volatility of our portfolio. Also, the **risk contribution (RC)** of the ith asset is defined as

$$\mathcal{RC}_i(x) := x_i \frac{\partial \mathcal{R}(x)}{\partial x_i} = x_i \cdot [-(\mu_i - r) + c\frac{(\Sigma x)_i}{\sqrt{x^\top \Sigma x}}]$$

2.2 Unconstrained case: original optimization problem

[1] defined the risk budgeting portfolio problem as follows

$$\begin{cases} \frac{\mathcal{RC}_i(x)}{\mathcal{R}(x)} = b_i \quad \text{(risk budget)} \\ b_i > 0,\ \forall i \\ x_i > 0,\ \forall i \quad \text{(long-only portfolio)} \\ \sum_{i=1}^n b_i = 1 \\ \sum_{i=1}^n x_i = 1 \end{cases} \tag{2}$$

where we want to solve x from this non-linear system.

Note that the constraint $b_i > 0$ is necessary for the uniqueness of the **risk-budgeting (RB) portfolio** x.

2.3 Unconstrained case: other mathematical formulations

[1] shows that the problem in the section above is equivalent to

$$x^\star(\kappa) = \arg\min\ \mathcal{R}(x)$$

$$\text{s.t.}\ \begin{cases} \sum_{i=1}^n b_i \log x_i \geq \kappa \\ x \geq 0 \end{cases}$$

where κ is an arbitrary constant and we use

$$x_{\text{RB}} = \frac{x^\star(\kappa)}{\mathbb{1}^\top x^\star(\kappa)}$$

to compute our RB portfolio.

The problem above can be shown to be equivalent to

$$x^\star(\lambda) = \arg\min\ \mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \log x_i \tag{3}$$

$$\text{s.t. } x \geq 0$$

and

$$x_{\text{RB}} = \frac{x^\star(\lambda)}{\mathbb{1}^\top x^\star(\lambda)} \tag{4}$$

Apparently λ is the Lagrange multiplier.

We will use formulation equations 3 and 4 in the following to solve the RB problem numerically.

2.4 Constrained case: original optimization problem

Since the solution to the unconstrained RB problem is unique, if we introduce the constraints directly, in general we will have no solution. Therefore, intuitively speaking, we need to replace the equality risk budget by an approximate budget

$$\begin{cases} \mathcal{RC}_i(x) \approx b_i \mathcal{R}(x) \\ x \in \mathcal{S} \\ x \in \Omega \end{cases}$$

where Ω is the additional set of constraints we further require and $\mathcal{S}$ is the standard simplex

$$S := \{x_i \geq 0 : \sum_{i=1}^n x_i = 1\}$$

We can see that $\mathcal{S}$ is just a fancy way to express the normalization constraint.

The problem above can be formulated as

$$x^\star(\mathcal{S}, \Omega) = \arg\min \sum_{i=1}^n \sum_{j=1}^n \left(\frac{\mathcal{RC}_i(x)}{b_i} - \frac{\mathcal{RC}_j(x)}{b_j}\right)^2 \tag{5}$$

$$\text{s.t. } x \in (\mathcal{S} \cap \Omega)$$

Throughout the project, we will only consider a certain type of separable constraints called **box-type constraints**:

$$\Omega = \{x \in \mathbb{R}^n : x^- \leq x \leq x^+\}$$

where $x^- \leq x^+$ element-wise. That is to say,

$$\Omega = \bigcap_{i=1}^n \Omega_i \quad (\text{“separable”})$$

where

$$\Omega_i = \{x_i \in \mathbb{R} : x_i^- \leq x_i \leq x_i^+\} \tag{6}$$

This type of constraints is called “box-type” because the feasible domain is essentially a box in the n-dimensional linear space.

In the unconstrained case, we have $\frac{\mathcal{RC}_i(x)}{b_i} = \frac{\mathcal{RC}_j(x)}{b_j}$, $\forall i, j$, as shown in [1]. Thus, the target function 5 reaches its minimum of 0.

2.5 Constrained case: other mathematical formulations

[2] shows that problem 5 is equivalent to the following problem

$$x^\star(\Omega, \lambda) = \arg\min\ \mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \log x_i \tag{7}$$

$$\text{s.t. } x \in \Omega$$

Here λ is still the Lagrange multiplier and we use

$$x^\star(\mathcal{S}, \Omega) = \{x^\star(\Omega, \lambda^\star) : \sum_{i=1}^n x_i^\star(\Omega, \lambda^\star) = 1\} \tag{8}$$

to calculate the final constrained RB portfolio.

We will use formulation equation 7 and 8 to solve the constrained RB problem numerically.

3 Numerical algorithms

3.1 Algorithms for unconstrained RB problem

3.1.1 Newton algorithm

An idea one will naturally come up with is to use the Newton algorithm. More precisely, in iteration $k+1$, Newton algorithm uses the following updating equation:

$$x^{(k+1)} = x^{(k)} - \eta^{(k)} \left(\frac{\partial^2 f(x^{(k)})}{\partial x \partial x^\top} \right)^{-1} \frac{\partial f(x^{(k)})}{\partial x}$$

where $\eta^{(k)} \in [0,1]$ is the learning rate in step k. We will set $\eta^{(k)} = 1, \ \forall k$ in this report.

The Newton algorithm has a convergence speed of order two. As a trade-off, it requires the Hessian of the target function. Luckily, it can be shown that both the gradient and the Hessian can be expressed explicitly[1].

3.1.2 CD algorithms

Except from second-order algorithms, we can also use first-order optimization algorithms such as **coordinate descent (CD)** algorithm family. The CD algorithm family has the advantage of transforming the problem from high dimension to one dimension (i.e., a scalar problem). In terms of choosing the variable to optimize in each iteration, there are typically two algorithms: CCD and RCD algorithm.

The **cyclical coordinate descent (CCD) algorithm** is a type of gradient descent algorithm which minimizes the target function along one coordinate at each step, and we cyclically iterate through all coordinates. That is to say, in iteration $k+1$,

$$x_i^{(k+1)} = \arg\min_x f(x_1^{(k+1)}, ..., x_{i-1}^{(k+1)}, x, x_{i+1}^{(k)}, ..., x_n^{(k)})$$

where k is the iteration index and i is the coordinate index. In every loop k, the i index goes from 1 to n.

[1]See the footnote on page 4 of [2].

Griveau-Billion *et al.* (2013) proposed a closed-form formula for the CCD algorithm in RB problem with the risk measure defined as equation 1. This explicit formula can be found on page 5 of [2] and is used in our implementation.

The **random coordinate descent (RCD) algorithm** is very similar to CCD, except that in the case of RCD, we randomly choose an index to optimize for each iteration.

3.2 Algorithms for constrained RB problem

The Lagrangian of constrained RB problem is

$$\mathcal{L} = \mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \log x_i + \mathbb{I}_\Omega(x) \tag{9}$$

where

$$\mathbb{I}_\Omega(x) = \begin{cases} 0, & x \in \Omega \\ +\infty, & \text{otherwise} \end{cases}$$

In order to find the constrained RB portfolio $x^\star(\mathcal{S}, \Omega)$, we can implement the following two-step procedure:

1. Find $x^\star(\Omega, \lambda)$ for various λ.
2. Given $x^\star(\Omega, \lambda)$ from Step 1, find $x^\star(\mathcal{S}, \Omega)$.

Step 2 is relatively easy: we can regard $x^\star(\Omega, \lambda)$ as a function of λ and apply bisection algorithm to find $\lambda^\star$. The pseudo-code can be found as Algorithm 1 on page 10 of [2].

Step 1 itself is a constrained optimization problem. There are typically two ways of solving it: CCD algorithm and mixed ADMM-CCD algorithm.

3.2.1 CCD algorithm

The idea of CCD algorithm is already illustrated in previous sections. In terms of implementing CCD in the constrained RB problem, there is only one technical issue left: computing $\frac{\partial \mathbb{I}_\Omega(x)}{\partial x_i}$.

This can be achieved by first applying the gradient descent on the Lagrangian without $\mathbb{I}_\Omega(x)$, which is

$$\mathcal{L}_0 = \mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \log x_i$$

and then project each of the coordinate on set Ω

$$x_i \leftarrow \mathcal{P}_{\Omega_i}(x_i) = \begin{cases} l_i, \ x_i \leq l_i \\ x_i, \ l_i \leq x_i \leq h_i \\ h_i, \ x_i \geq h_i \end{cases}$$

where Ω_i is defined in 6 and l_i, h_i is the lower and upper bound of the "box" in dimension i.

The pseudo-code of CCD algorithm in constrained RB problem can be found on page 13 of [2].

3.2.2 Mixed ADMM-CCD algorithm

We notice that we can write the Lagrangian 9 in the following form

$$\mathcal{L} = f(x) + g(x) \tag{10}$$

where

$$\begin{cases} f(x) := \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \log x_i \\ g(x) := \mathbb{I}_\Omega(x) \end{cases}$$

Solving problem 10 is equivalent to solving

$$\{x^\star, z^\star\} = \arg\min f(x) + g(z) \tag{11}$$

$$\text{s.t. } x^\star - z^\star = 0$$

It follows that problem 11 can be solved in the framework of **alternative direction method of multipliers (ADMM)**[2]. It can be easily seen that one advantage of ADMM algorithm is that it can be used in distributed computation.

Therefore, we can

1. use CCD to optimize the $f(x)$ part;
2. use projection operator to deal with $g(x)$;
3. use ADMM framework to synchronize the solutions $x^\star, z^\star$ in each iteration.

This is exactly the intuition of **mixed ADMM-CCD algorithm**. Pseudo-code can be found on page 14 of [2].

[2] A good introduction to ADMM algorithm can be found on https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

4 Efficiency of optimization algorithms

4.1 Unconstrained case: Newton algorithm, CCD and RCD

We compare the three algorithms above in the following settings:

- 5 assets in the market, with $\mu = \mathrm{E}[r]$ and $\Sigma = \mathrm{Cov}[r]$ generated randomly.
- $r = 0.02, \lambda = 1, c = 0.1$.
- For Newton algorithm, $\eta = 0.1$ in every iteration.
- Run Monte Carlo simulation 100 times.

The results can be found as table 1, figure 1 and figure 2.

	Newton	CCD	RCD
times the algo **fails to converge** (100 simulations in total)	100	2	0

Table 1: Monte Carlo results: times that the algorithm fails to converge out of 100 simulations

From the results, we can see that RCD is the best among these three algorithms. Therefore, we will use RCD to solve the unconstrained RB problem in the following.

4.2 Constrained case: CCD and mixed ADMM-CCD

The parameter setting of the constrained case is far less arbitrary, because

- if the constraints are too loose to make an effect, the constrained problem will reduce to its corresponding unconstrained problem and we shall not get the "real performance" of our constrained optimization algorithm;
- if the constraints are too restrictive, they will affect the original unconstrained problem heavily, leading to a solution which is very far from that of its corresponding unconstrained problem. In the worst case, there might even be no solution.

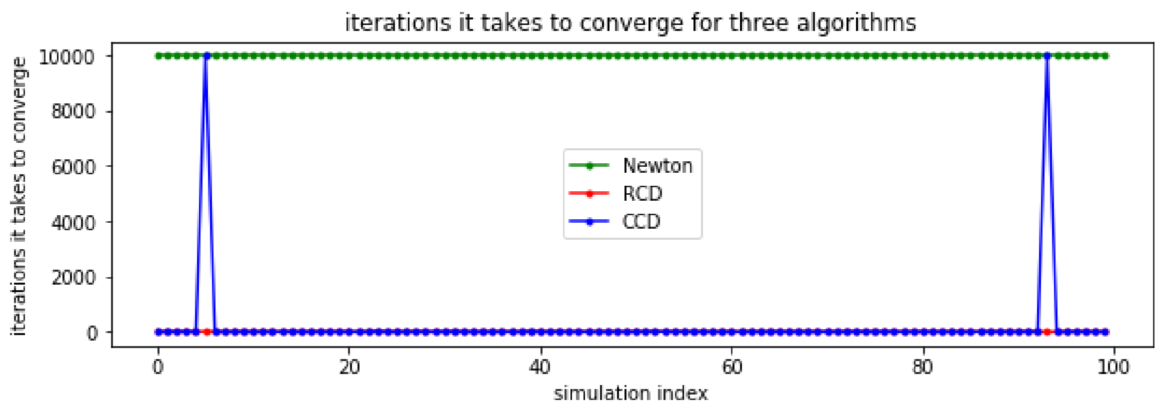

Figure 1: iterations it takes for three algorithms to converge

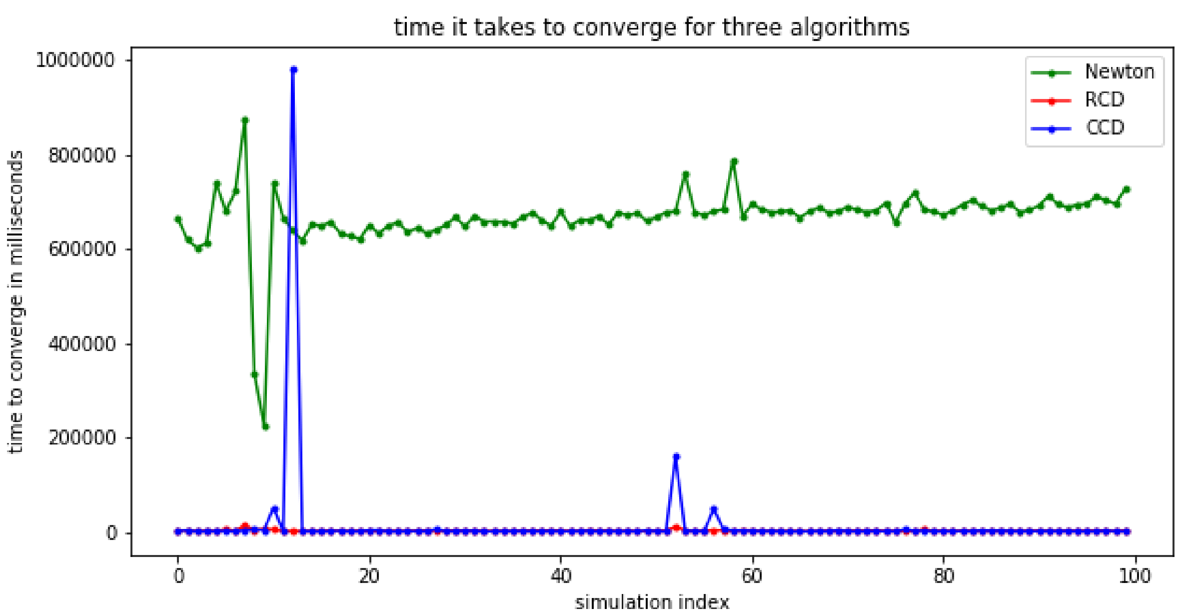

Figure 2: time it takes for three algorithms to converge

As a result, I only tested the example provided on page 18 of [2]. At least for this single example, both CCD and mixed ADMM-CCD give us reasonable results and the running speeds are both fast enough. We will use CCD algorithm in the following because of its simplicity in logic.

5 Application of RB portfolio

5.1 Choice of universe[3]

We want to choose a set of stocks that

- have high liquidity, so that the transaction cost will not affect our conclusion of the research.
- have been listed for at least 6 years (1-year burn-in period plus 5-year research period).
- have a total number of tickers around 50.
- are the components of an index.

We need historical data for 6 years to do the backtest: we will monitor the behaviour of different strategies in the last 5 years; in order to avoid lookahead bias, we need the data for one more year before our research period to calculate the covariance matrix and the expected return vector, which will be used to construct the weight matrix of the first year in our research period.

A 50-ticker universe is large and diversified enough so that we can analyze the difference among our strategies. It is also small enough for us to handle the numerical computation on our laptops.

If the stocks can compose an index and can thus be traded as an ETF, then our analysis will be more realistic and the index itself can be set as a benchmark to our analysis.

Based on the discussions above, we choose our universe to be the holding compositions of XLG (Invesco S&P 500 Top 50 ETF) except PYPL, which is listed only since 2015, and hence has too short history.

The number of stocks in our universe is 49. The tickers are as follows.

[3]This subsection is mostly written by Chao Peng from Baruch MFE program, who also kindly provided me all the close price and cap size data for the research.

- FB
- VZ
- NFLX
- MCD
- PEP
- PM
- BRK-B
- WFC
- UNH
- MDT
- TMO
- BA
- UTX
- V
- CSCO
- ORCL
- AVGO
- GOOGL
- DIS
- AMZN
- PG
- WMT
- XOM
- JPM
- C
- MRK
- ABT
- ABBV
- HON
- MSFT
- MA
- ADBE
- IBM
- T
- CMCSA
- HD
- KO
- COST
- CVX
- BAC
- JNJ
- PFE
- AMGN
- LLY
- UNP
- AAPL
- INTC
- CRM
- ACN

5.2 General settings of our strategies

For all the trading strategies in this part, unless specifically stated, we will adopt the following settings:

- GMV (Gross Market Value) = 1
- long-only strategy
- yearly rebalancing
- burn-in period: 2014.01.01 - 2014.12.31
- research period: 2015.01.01 - 2019.06.30

5.3 Application of unconstrained RB portfolio: ERC portfolio

The **equal risk contribution (ERC) portfolio** is defined as the RB portfolio with equal risk budget, i.e.,

$$b_i = \frac{1}{n}, \ \forall i$$

Since the 2008 financial crisis, ERC portfolio has been very popular and has influenced the asset management industry, according to Roncalli (2017).

Before constructing the ERC portfolio, we have first constructed two kinds of portfolio which are frequently used as the benchmarks in the evaluation of a trading strategy: **the cap-weighted portfolio (which is also called the index portfolio)** and **the equal-weighted portfolio**.

The index portfolio, which is defined as the yearly-rebalancing cap-weighted portfolio in our universe has the performance as shown in figure 3 and table 2. In this case, the cross-sectional weight of any ticker is proportional to its cap size.

Figure 3: pnl of index (cap-weighted) portfolio

	turnover	Sharpe ratio	maximum drawdown	annual return
2015	0.0007	0.6056	-0.1368	0.0965
2016	0.0007	1.0396	-0.0743	0.1483
2017	0.0006	3.7760	-0.0210	0.2522
2018	0.0008	0.1605	-0.1990	0.0207
2019	0.0000	2.0166	-0.0800	0.1900
average	0.0006	1.5197	-0.1990	0.1415

Table 2: performance of index portfolio

Another popular portfolio is equal-weighted portfolio, as shown in figure 4 and table 3. In this case, we give every ticker in our universe equal dollar weight.

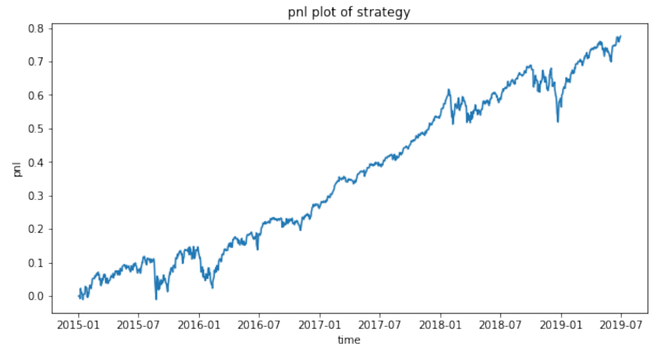

Figure 4: pnl of equal-weighted portfolio

	turnover	Sharpe ratio	maximum drawdown	annual return
2015	0.0000	0.8121	-0.1283	0.1295
2016	0.0000	1.0063	-0.0908	0.1485
2017	0.0000	4.1052	-0.0218	0.2605
2018	0.0000	0.3402	-0.1700	0.0494
2019	0.0000	2.1592	-0.0607	0.1855
average	0.0000	1.6846	-0.1700	0.1547

Table 3: performance of equal-weighted portfolio

Compared with cap-weighted portfolio, we have larger Sharpe, higher return and lower (absolute) drawdown. This is probably a result of the cap size effect: small cap stocks tend to outperform large cap stocks in the long term.

Now we construct the ERC portfolio. On the last trading day in each year, we

1. calculate the empirical mean vector $\hat{\mu}$ and covariance matrix $\hat{\Sigma}$ of this year;

2. use RCD algorithms with $\mu = \hat{\mu},\ \Sigma = \hat{\Sigma}$ to compute the dollar holding of each ticker;

3. use the (normalized) optimization result as our weight for the next year.

Also, we set $c = 1$, meaning that we put equal focus on the expected return and the volatility of our portfolio. λ is set to 1 but it can be proven that the choice of λ will not affect our optimization result.

The performance of ERC portfolio can be found in figure 5 and table 4.

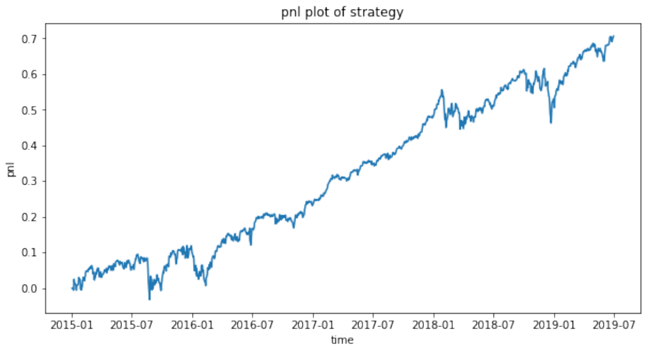

Figure 5: pnl of ERC portfolio

	turnover	Sharpe ratio	maximum drawdown	annaul return
2015	0.0000	0.6667	-0.1270	0.1022
2016	0.0007	1.0312	-0.0823	0.1438
2017	0.0007	4.0556	-0.0183	0.2373
2018	0.0010	0.3290	-0.1529	0.0460
2019	0.0018	2.2250	-0.0499	0.1767
average	0.0009	1.6615	-0.1529	0.1412

Table 4: performance of ERC portfolio

From the results, we can find that

- compared with the cap-weighted portfolio, our ERC portfolio has higher Sharpe and lower (absolute) drawdown.
- the performance of ERC portfolio is more like equal-weighted portfolio than cap-weighted portfolio, which is probably because there are not many significant correlation relationships among tickers[4].

By constructing the ERC portfolio, we have a better control of the volatility of our portfolio by incorporating the correlation matrix of our universe. Therefore, we get less exposure on the volatility in both "normal zone" and "tail risk zone", as illustrated in figure 6[5].

I have also tried the monthly-rebalancing ERC portfolio. Compared with the annually-rebalancing version, there is almost no difference in performance. This is probably because that the correlation matrix Σ and expected return vector μ are slow-changing.

[4]Correlation profile of this universe can be found in the Appendices part.

[5]This plot is credited to Yiyang Zhao from Peking University.

5.4 The effect of parameter c in ERC portfolio

In this part, we change the value of c in equation 1, which will definitely change the composition as well as the performance of our final ERC portfolio.

In the test below, c varies from $e^{-0.5}$ to $e^{1.5}$ in equal step in the log space. Since we run a long-only strategy, the most important factor affecting the performance is apparently the market beta. Therefore, we will plot the spread (i.e., difference) of performance between our strategy and the index portfolio for each c, in order to get rid of the market exposure and find the underlying trend.

The results can be found in figure 7.

Note that since the drawdown in our backtesting system is a negative number, the curve in the lower left corner actually means that the ERC portfolio has larger (absolute) drawdown for larger c.

As c increases,

- turnover deceases, which means that the covariance matrix of the previous year indeed has predicting power on the covariance profile of this year.
- Sharpe increases, which is a natural payoff for us when we are more careful about the volatility in the "normal zone".
- drawdown is larger, which is very interesting. This tells us that the more careful we are about the volatilities in the "normal zone", the greater loss we will suffer when there is a tail event (i.e., volatility spike).

5.5 Application of constrained RB portfolio: smart beta portfolio

For a universe which contains some small cap stocks, such as Russell 3000, the ERC portfolio might be hard to achieve. This is because in ERC portfolio, we assign the same risk contribution to large and small cap stocks, and that may lead to some liquidity issues, since the transaction cost of small cap stocks is typically much higher than that of large cap ones.

To solve the problem, we can force the small cap stocks to have the same weights in our portfolio as

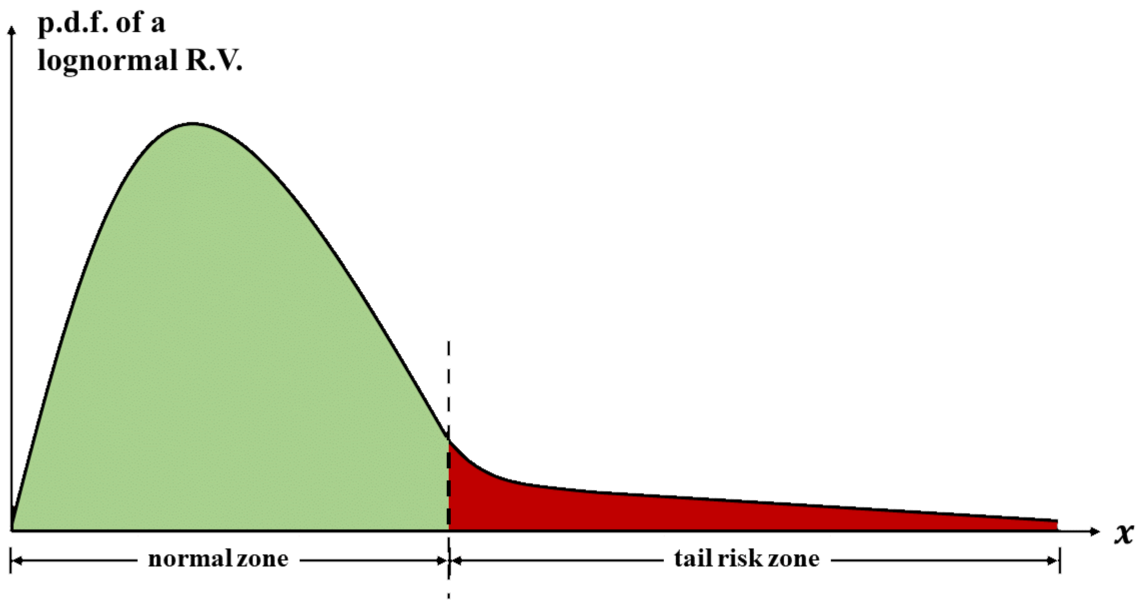

Figure 6: distribution of daily volatility

what they should have in the cap-weighted portfolio, and apply the idea of ERC portfolio only on mid and large cap stocks. Mathematically speaking, let Ω_{SC} be the universe of small cap stocks, we want to apply the following constraints on our ERC portfolio:

$$\begin{cases} x_i \geq 0, & \text{if } i \notin \Omega_{\text{SC}} \\ x_{cw,i} \leq x_i \leq x_{cw,i}, & \text{if } i \in \Omega_{\text{SC}} \end{cases} \tag{12}$$

where x_{cw} is the weight matrix of cap-weighted portfolio and i is the ticker index.

ERC portfolio which further satisfies constraints 12 is called **smart beta portfolio**. Since constraints 12 are box-type, we can use the algorithms we have talked about above to construct the smart beta portfolio.

In our 49-ticker universe, on the first day of each year, we define the 5 tickers with the smallest cap sizes as "small cap stocks". The tickers names of "small cap stocks" are listed in table 5. The performance of smart beta portfolio can be found in figure 8 and table 6.

	1	2	3	4	5
2015	AVGO	NFLX	CRM	ADBE	TMO
2016	JPM	AVGO	NFLX	ADBE	CRM
2017	NFLX	ADBE	CRM	TMO	ABT
2018	NFLX	CRM	COST	TMO	ABT
2019	MDT	UTX	COST	CRM	TMO

Table 5: 5 smallest tickers in our universe for each year

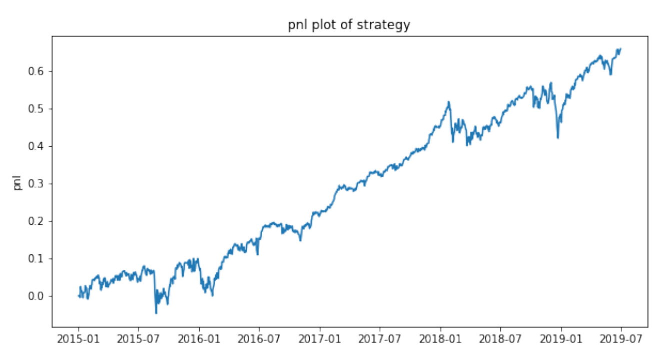

Figure 8: pnl of smart beta portfolio

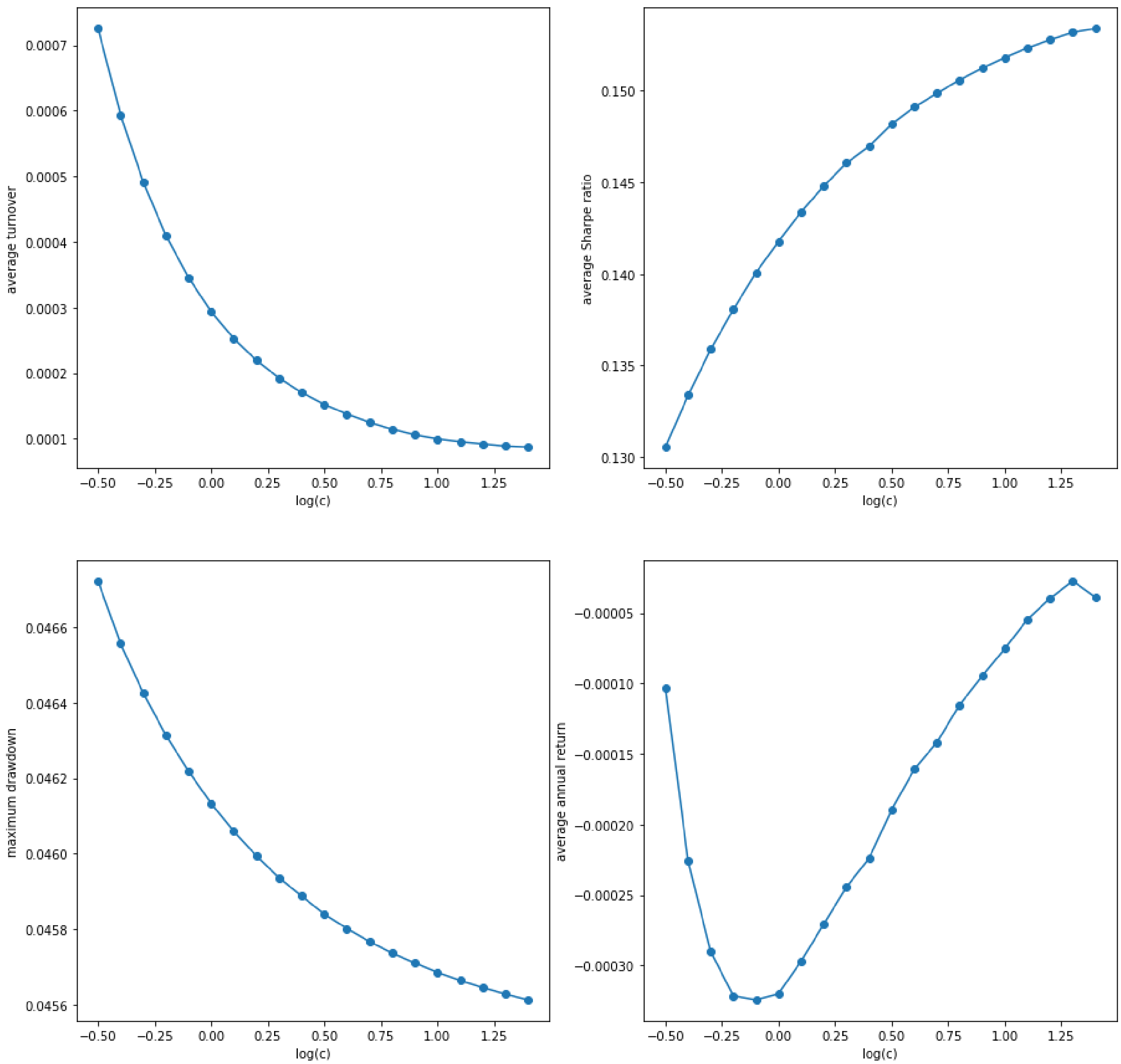

Figure 7: difference in performance (ERC−index) w.r.t. parameter c

	turnover	Sharpe ratio	maximum drawdown	annual return
2015	0.0000	0.5500	-0.1268	0.0827
2016	0.0008	1.0683	-0.0722	0.1429
2017	0.0008	3.9865	-0.0173	0.2287
2018	0.0012	0.2352	-0.1486	0.0301
2019	0.0022	2.2777	-0.0521	0.1747
average	0.0010	1.6235	-0.1486	0.1318

Table 6: performance of smart beta portfolio

- Compared with ERC portfolio, we have lower Sharpe ratio and lower return, which is understandable because we have lost some of the benefits of "cap size effect" by forcing the small cap stocks to have small weights.

- But we did decrease our drawdown consistently. This shows that the smart beta portfolio is a further improvement to the traditional ERC portfolio in terms of controlling the tail risk.

6 Possible topics for further research

6.1 Return: raw return v.s. residual return

We can find that the pnl curves of almost all the strategies above have very similar shapes: every strategy, including the cap-weighted and equal-weighted one, performs reasonably well during our research period, with several large drawdowns in mid 2015, early 2018 and late 2018.

That is probably because the dominant factor which affects our performance is market beta rather than the alpha of each strategy. In order to get rid of beta and make the difference between the strategies above more significant, we can use residual return rather than raw return to compute the pnl and calculate all the evaluation metrics[6]. We do not test the performance with residual return due to the limited access of data.

[6] There are many ways to construct residual return, such as cross-sectional linear regression or cross-sectional PCA. But the methodologies are beyond the scope of our discussion in this paper.

6.2 "Real small cap stocks" v.s. "fake small cap stocks"

Another concern is that since our universe is by nature a large cap universe, there is actually no small cap stocks, which casts doubts on our conclusion on the verification of the cap-size effect in section 5.3 and on the smart-beta portfolio in section 5.5. It would be much more convincing if a universe with both large cap stocks and small cap stocks, such as Russell 3000, is chosen and we select the "real small cap stocks" to re-do the search in section 5.3 and 5.5.

Appendices

A Correlation profile of the universe

Below is an example of the correlation profile of our universe, showing the correlation matrix in the year of 2016.

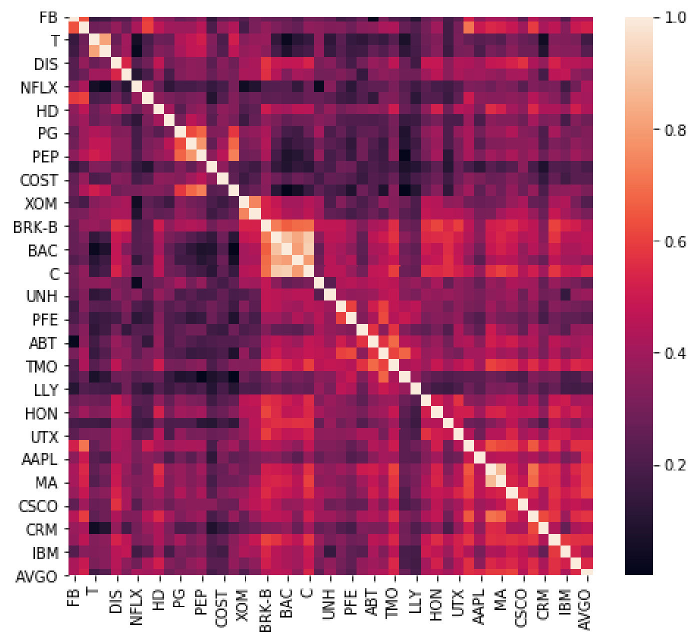

Figure 9: Correlation matrix of our universe in 2016

From the plot, we can find that except for the financial companies ("BRK-B", "BAC" and "C"), there seems to be no significant correlation.

References

[1] Benjamin Bruder and Thierry Roncalli. “Managing Risk Exposures using the Risk Budgeting Approach”. In: *Risk Management eJournal* (2012). DOI: http://dx.doi.org/10.2139/ssrn.2009778.

[2] Jean-Charles Richard and Thierry Roncalli. “Constrained Risk Budgeting Portfolios: Theory, Algorithms, Applications Puzzles”. In: (2019). URL: https://arxiv.org/pdf/1902.05710.pdf.