Midterm

ELEC/IEDA3180 - Data Driven Portfolio Optimization
Spring 2022/2023

Prof. Daniel Palomar, TAs: Yifan YU

2023-03-17

The HKUST Academic Honor Code applies. This assignment is to be done individually. Cheating **won't** be tolerated. Total marks: 100 points.

Midterm

The attached file `"dataset.csv"` is the **log-returns** of 10 stocks.

Question 1 : Risk Parity Portfolio (20 points)

Risk Parity Portfolio (RPP) aims at equalizing the risk contribution from the invested assets in the global portfolio risk. Given a portfolio $\mathbf{w} \in \mathbb{R}^N$ and the return covariance matrix $\mathbf{\Sigma}$, the portfolio volatility is

$$\sigma(\mathbf{w}) = \sqrt{\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}}.$$

The relative risk contribution (RCC) form the ith asset is defined as

$$\mathrm{RRC}_i = \frac{w_i (\mathbf{\Sigma} \mathbf{w})_i}{\sigma^2(\mathbf{w})} = c_i,$$

where $\mathbf{c}$ is the given risk budget vector.

One RPP formulation is given by

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \frac{1}{2}\mathbf{x}^\top \mathbf{\Sigma}\mathbf{x} - \mathbf{c}^\top \log(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \geq \mathbf{0}. \end{array}$$

We can recover the portfolio by $\mathbf{w} = \mathbf{x}/(\mathbf{1}^\top \mathbf{x})$.

(a)

Prove the problem is convex.

(b)

Use log-returns from `"2015-01-06"` to `"2015-05-29"` to find $\mathbf{w}$ and the corresponding relative risk contribution (RRC) for following $\mathbf{c}$:

$$\mathbf{c}_1 = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]^\top$$

$$\mathbf{c}_2 = [0.2, 0.2, 0.2, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.1]^\top$$

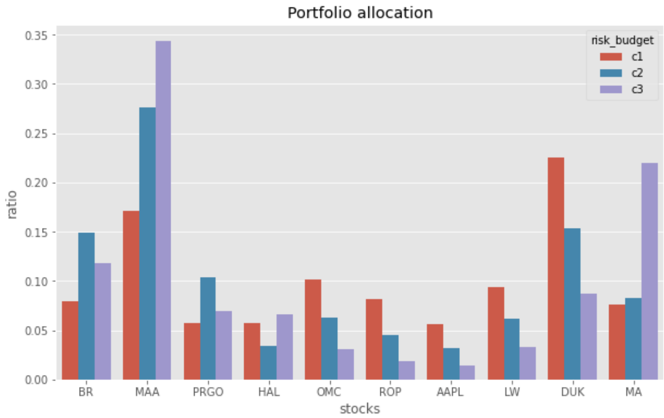
$$\mathbf{c}_3 = [0.15, 0.25, 0.1, 0.1, 0.02, 0.02, 0.02, 0.02, 0.02, 0.3]^\top$$

(c)

Plot capital allocation distribution as the format of the following figure.

(d)

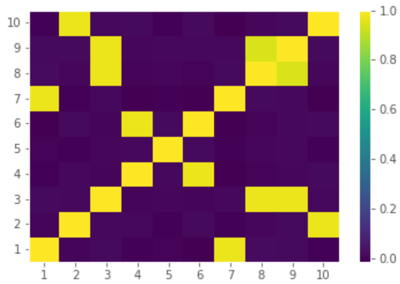
Plot RRC distribution as the same format.

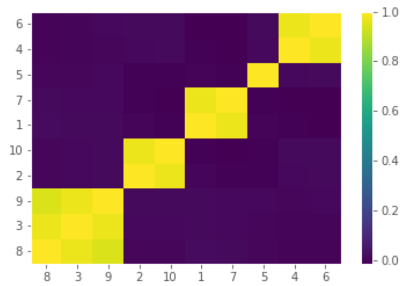


Question 2: Hierarchical Risk Parity (25 points)

Hierarchical Risk Parity (HRP) combines the ideas of hierarchical clustering and inverse-variance allocation. The algorithm operates in three stages: tree clustering, quasi-diagonalization and recursive bisection. The code of stage 1 and 2 is provided in the attached file `"Q2code.py"`.

Stage 1 & 2 reorder the stocks. The figures[1] show the correlation matrix before and after Stage 1 & 2. Values of correlation matrix are not changed. We just reorder the stocks so that the similar ones are next to each other and form a quasi-diagonal matrix.





Stage 3 will be illustrated with a small example. Suppose we have 6 stocks and after Stage 1 & 2, the order of stocks are {“JNJ”, “PG”, “AMZN”, “JPM”, “V”, “FB”}. The steps of Stage 3 are as follows.

1.Initialization.

Set the list of stocks $L = \{1, 2, 3, 4, 5, 6\}$. The numbers of L correspond to the stocks.

Assign a unit weight to all items: $w_n = 1, n = 1, \ldots, 6$.

2.Bisection and weight allocation.

2.1 Bisect L into two subsets $L_1 = \{1, 2, 3\}$ and $L_2 = \{4, 5, 6\}$, which correspond to {“JNJ”, “PG”, “AMZN”} and {“JPM”, “V”, “FB”} .

2.2 Calculate the following value for $L_i, i = 1, 2$:

$$\widetilde{V}_i = \widetilde{w}_i^\top V_i \widetilde{w}_i,$$

$$\widetilde{w}_i = \mathrm{diag}(V_i)^{-1} / \mathrm{tr}(\mathrm{diag}(V_i)^{-1}),$$

where V_i is the covariance matrix of the L_i, $\mathrm{diag}()$ is the diagonal operator and $\mathtt{tr}()$ is the trace operator.

2.3 Compute the split factor:

$$\alpha = 1 - \frac{\widetilde{V}_1}{\widetilde{V}_1 + \widetilde{V}_2}.$$

2.4 Re-scale allocations $w_n, n \in L_1$ by α: $w_n = w_n \times \alpha, n = 1, 2, 3$.

Re-scale allocations $w_n, n \in L_2$ by $1 - \alpha$:
$w_n = w_n \times (1 - \alpha), n = 4, 5, 6$.

3.Continue the bisection and weight allocation for L_1 and L_2 until all subsets have only 1 stock.
In this example, we get subsets {1} and {2,3} from L_1. Then we apply step 2 to {1} and {2,3}, updating the allocations of stock {1} and {2,3}. After this round, {1} has only one stock, so no more bisection can be done. But {2,3} need to be bisected and update the allocations one more time.
The same is for L_2.

(a)

Plot the correlation matrix before and after the stage 1 & 2 with the code provided. (Use the function `"hrp_s12()"`.)

Please read the notes in the attached code file carefully. The `"hrp_s12()"`, `"hrp_s3()"` and `"getRecBipart()"` are three functions you need to focus. You do not need to read the code of other functions in detail.

(b)

Finish the code for Stage 3 (the function `"getRecBipart()"`). Please read the notes in the attached code file carefully.

(c)

Print the final weight for log-returns from `"2015-01-06"` to `"2015-05-29"` .

Question 3: (55 points)

(a)

Consider a 3-factor model. Find the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the model with the factors provided in the attached file `"factors.csv"`. Now we have the three explicit factors in matrix $\mathbf{F}$ and want to fit the model

$$\mathbf{X}^{\top} = \boldsymbol{\alpha}\mathbf{1}^{\top} + \mathbf{B}\mathbf{F}^{\top} + \mathbf{E}$$

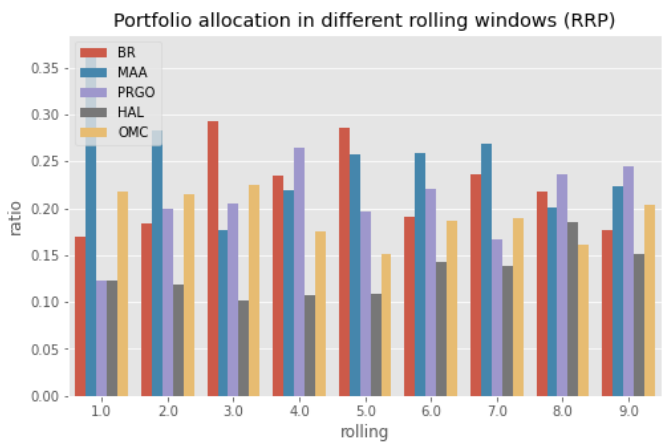
where the loadings are a matrix of betas. Use all data in `"dataset.csv"`. **Print** the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

Find the five stocks with the highest α.

(b)

With the five stocks you find in Q3.a, we now implement walk-forward (WF) process based on rolling windows. We divide the 1000 days into 10 windows of length 100 denoted as $[T_0, T_1, \ldots, T_9]$ and in each $T_n, n = 1, \ldots, 9$ we will stick to a certain portfolio $w_n, n = 1, \ldots, 9$, calculated based on the data only in T_{n-1}.

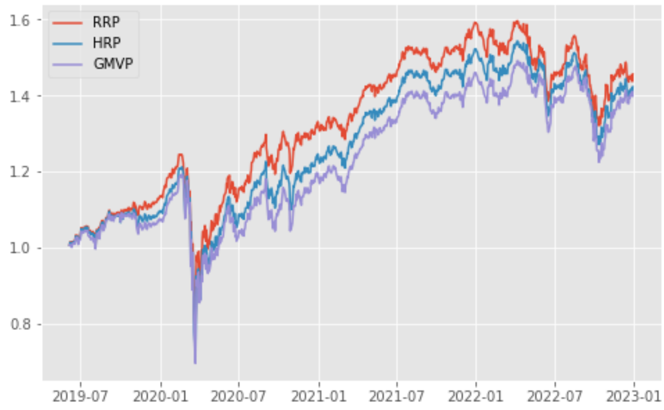
Apply the RPP (Q1), HRP (Q2) and GMVP portfolio. For RRP portfolio, use the budget vector [0.2,0.2,0.2,0.2,0.2]. **Create** grouped bar charts of the allocation for each portfolio like the following figure.



(c)

Compute the simple return (not compounded) in this backtest.

Plot the return path like the following figure.



(d)

Redo the Q3.c with the shrinkage estimator as follows:

$$\hat{\mathbf{\Sigma}}^{\text{SH}} = (1 - \rho)\hat{\mathbf{\Sigma}} + \rho\mathbf{T},$$

where $\rho = 0.3$, $\hat{\mathbf{\Sigma}}$ denotes the sample covariance matrix, and the target is $\mathbf{T} = \frac{1}{N}\text{Tr}(\hat{\mathbf{\Sigma}}) \times \mathbf{I}$ (scaled identity).

Observation #1: make sure to write your code in a modular, readable way. Code organization will be taken into account for the grading. Use meaningful names for variables, create functions to organize your code as much as possible. In case of doubt on coding best practices, take a look at the Google's style guide. Note that you don't have to strictly follow that style, you can develop your own, but make sure it is understandable and you use it consistently.

Observation #2: submit your code via canvas in zipped file (`.zip`) containing a Jupyter notebook (`.ipynb`) and also its exported version in the HTML (`.html`) format.

---

1. The figures are just for illustration and data for figures is generated by code. The real data can hardly generate such a neat correlation matrix.↩