The Use of Hierarchical Risk Parity (HRP) Approach in Portfolio Optimizations

Zehao Dong, Xinyu Guo, Yukang Zhou, Run Qiao, Peicheng Wang

ABSTRACT

In this project, we mainly research on the use of Hierarchical Risk Parity (HRP) approach in portfolio construction. HRP applies machine learning techniques to build a diversified portfolio based on information contained in covariance matrix through its three main steps: Tree Clustering, Quasi-Diagonalization and Recursive-Bisection. In our empirical tests, HRP fails to stand out in-sample, but outperforms other traditional portfolio construction models out-of-sample. Besides, HRP algorithm is more stable and efficient in portfolio optimization so it would be more feasible in real-world investment process.

Keywords: Hierarchical Risk Parity, machine learning, portfolio optimization

Background

Investment portfolio construction is probably the most frequent financial problem. For many years lots of investment managers have been struggling with building a profitable and diversified portfolio. So far many methods have been used in portfolio construction, the most popular of which are Markowitz's Effective Frontier and Risk Parity method. On a daily basis, investment managers aim to build a portfolio that combines their perceptions and forecasts of risk and returns. Here's the original question that 24-year-old Harry Markowitz tried to answer over 6 decades ago. His great insight is the recognition of various levels of risk associated with different optimal portfolios in terms of risk-adjusted returns, so "Effective Boundary" (Markowitz, 1952). It is difficult to optimally allocate portfolio with the highest expected returns. Instead of that, we should reconsider across the relevance of alternative investments to build a diversified portfolio.

Risk Parity

Risk parity in general is a portfolio allocation strategy using risk to determine allocations across various components of an investment portfolio. In other words, Risk Parity is about having each asset contributing in the same way to the portfolio overall volatility. By using Risk Parity method rather than traditional portfolio construction we can reduce risk and fees. The Risk Parity approach attempts to equalize risk by allocating funds to a wider range of categories such as stocks, government bonds, credit-related securities and inflation hedges. But traditional Risk Parity method has one big flaw that it does not take into account the correlation between different assets.

IVP (Inverse Variance Portfolio)

When we compare the results of the data, we also used a method of calculating weights called the Inverse Variance Portfolio, in which the weight of each asset is inversely proportional to its variance. IVP's mathematical calculation formula is shown as below:

$$w^{IVar} = \frac{1/\sigma^2}{\sum_{i=1}^{n} 1/\sigma^2}$$

Markowitz Method: Global Minimum Variance Portfolio

The initial milestone in this field was set by Harry Markowitz more than 60 years ago, when he first proposed the idea of "effective frontier", which is the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return, drawing up a fundamental framework for further portfolio optimization research.

One feasible approach to Markowitz's ideology is by computational solution, in which simulations are designed to generate randomly weighted portfolios. It should be noted that we generate random weights by a uniform distribution from 0 to 1, then unify them so that they add up to 1, which satisfies the full-investment-constraint. A few more possible distributions have been tested including uniform distribution on -1 to 1, or standard normal distribution, in which situations the portfolios are greatly more diversified in their volatilities and returns.

The generated portfolios form a pattern as the simulated points get multiplied, and the left bound is by definition the effective frontier generated by simulation shown in Exhibit 1.

Nevertheless, the more straight-forward approach would be deriving the effective frontier from optimization methods. To formulate the effective frontier, we need to solve:

$$\begin{gathered} max \quad -<\omega, C\omega> \\ s.t. \quad <1, \omega> = 1, \qquad <R, \omega> = \tilde{R} \end{gathered}$$

In order to compare with other methods involved in this research, we need to add long constraints to this solution:

$$\begin{gathered} max \quad -<\omega, C\omega> \\ s.t. \quad <1, \omega> = \vec{1}, <R, \omega> = \tilde{R}, \omega \geq \vec{0} \end{gathered}$$

The area under constrained effective frontier has shrunk greatly than the unconstrained one shown in Exhibit 2, due to the limitation that only long positions of assets are possible now.

By now, the strengths of traditional Markowitz's method have been evident: first it is more efficient than simulation methods and exact solutions are derived. Furthermore, it provides an overview of all possible portfolios and can be set as a benchmark for other methods.

On the other hand, the weakness manifests itself conspicuously, which comes in the name of "Markowitz's curse": the more correlated the assets are, the more we need to diversify in our portfolio construction, but on the other hand, the invertibility of covariance matrix is in worse conditions, thus we will receive more unstable solutions. And the inversion is prone to large errors when the covariance matrix is numerically ill-conditioned. Hence it is of highly need of a method that requires no invertibility of the covariance matrix.

Hierarchical Risk Parity

HRP applies Graph theory and machine learning techniques to build a diversified portfolio based on the information contained in the covariance matrix. However, unlike quadratic optimizers, HRP does not require the invertibility or positive-definitiveness of the covariance matrix.
The algorithm operates in three stages: Tree clustering, quasi-diagonalization and recursive bisection.

Stage 1: Tree Clustering

In this stage, we group similar investments into clusters based on their correlation matrix. Having a hierarchical structure helps us to improve stability issues of quadratic optimizers when inverting the covariance matrix.

Consider a $T * N$ matrix of observations X, such as returns series of N variables over T periods. We would like to combine these N column-vectors into a hierarchical structure of clusters, so that allocations can flow downstream through a tree graph.

First, we compute a $N * N$ correlation matrix with entries $\rho = \{\rho_{i,j}\}_{i,j=1,\ldots,N}$, where $\rho_{i,j} = \rho[X_i, X_j]$. Then distance measure is defined by $d: (X_i, X_j) \subset B \to \mathbb{R} \in [0,1]$, $d_{i,j} = d[X_i, X_j] = \sqrt{\frac{1}{2}(1-\rho_{i,j})}$, where B is the Cartesian product of items in$\{1, \ldots, i \ldots, N\}$. This allows us to compute a $N * N$ distance matrix $D = \{d_{i,j}\}_{i,j=1,\ldots,N}$. Matrix D is a proper metric space, in the sense that $d[X,Y] \geq 0$ (non-negativity), $d[X,Y] = 0 \Leftrightarrow X = Y$ (coincidence), $d[X,Y] = d[Y,X]$ (symmetry), and $d[X,Z] \leq d[X,Y] + d[Y,Z]$ (sub-additivity).

Second, we compute the Euclidean distance between any two column-vectors of D, $\tilde{d}: (D_i, D_j) \subset B \to \mathbb{R} \in [0, \sqrt{N}]$, $\tilde{d}_{i,j} = \tilde{d}[D_i, D_j] = \sqrt{\sum_{n=1}^{N} (d_{n,i} - d_{n,j})^2}$. Note the difference between distance metrics $d_{i,j}$and $\tilde{d}_{i,j}$. Whereas $d_{i,j}$ is defined on column-vectors of X, $\tilde{d}_{i,j}$ is defined on column-vectors of D (a distance of distances). Therefore, $\tilde{d}$ is a distance defined over the entire metric space D, as each $\tilde{d}_{i,j}$ is a function of the entire correlation matrix (rather than a particular cross-correlation pair).

Third, we cluster together the pair of columns (i^*, j^*) such that $(i^*, j^*) = argmin_{(i,j)}\{\tilde{d}_{i,j}\}$,and denote this cluster as $u[1]$.

Fourth, we need to define the distance between a newly formed cluster $u[1]$ and the single (unclustered) items, so that $\{\tilde{d}_{i,j}\}$ may be updated. In hierarchical clustering analysis, this is known as the "linkage criterion". For example, we can define the distance between an item i of $\tilde{d}$ and the

new cluster $u[1]$ as $\dot{d}_{i,u[1]} = \min\left[\{\tilde{d}_{i,j}\}_{j \in u[1]}\right]$ (the nearest point algorithm).

Fifth, matrix $\{\tilde{d}_{i,j}\}$ is updated by appending $\dot{d}_{i,u[1]}$ and dropping the clustered columns and rows $j \in u[1]$.

Sixth, applied recursively, steps 3-5 allow us to append $N-1$ such clusters to matrix D, at which point the final cluster contains all of the original items and the clustering algorithm stops.

Stage 2: Quasi-Diagonalization

This stage reorganizes the rows and columns of the covariance matrix, so that the largest values lie along the diagonal. This quasi-diagonalization of the covariance matrix (without requiring a change of basis) renders a useful property: similar investments are placed together, and dissimilar investments are placed far apart. The algorithm works as follows: we know that each row of the linkage matrix merges two branches into one. We replace clusters in $(y_{N-1,1}, y_{N-1,2})$ with their constituents recursively, until no clusters remain. These replacements preserve the order of the clustering. The output is a sorted list of original (unclustered) items.

Stage 3: Recursive Bisection

Stage 2 has delivered a quasi-diagonal matrix. The inverse-variance allocation is optimal for a diagonal covariance matrix, the proof is as follow: consider the standard quadratic optimization problem of size *N*,

$$\min_{\omega} \omega' V \omega$$
$$\text{s.t.}: \omega' a = 1_I$$

with solution $\omega = \frac{V^{-1}a}{a\prime V^{-1}a}$ for the characteristic vector $a = 1_N$, the solution is the minimum variance portfolio. If V is diagonal, $\omega_n = \frac{V_{n,n}^{-1}}{\sum_{i=1}^{N} V_{i,i}^{-1}}$. In the particular case of $N = 2, \omega_1 = \frac{\frac{1}{V_{1,1}}}{\frac{1}{V_{1,1}}+\frac{1}{V_{2,2}}} = 1 - \frac{V_{1,1}}{V_{1,1}+V_{2,2}}$, which is how this stage splits a weight between two bisections of a subset.

We can take advantage of these facts in two different ways: a) Bottom-up, to define the variance of a continuous subset as the variance of an inverse-variance allocation; b) top-down, to split allocations between adjacent subsets in inverse proportion to their aggregated variances. The following algorithm formalizes this idea:

1. The algorithm is initialized by:
   a. setting the list of items: $L = \{L_0\}$, with $L_0 = \{n\}_{n=1,\dots,N}$
   b. assigning a unit weight to all items: $w_n = 1, \forall n = 1, \dots, N$
2. If $|L_i| = 1, \forall L_i \in L$, then stop
3. For each $L_i \in L$ such that $|L_i| > 1$:
   a. bisect L_i into two subsets, $L_i^{(1)} \cup L_i^{(2)} = L_i$ where $\left|L_i^{(1)}\right| = \text{int}\left[\frac{1}{2}|L_i|\right]$, and the order is preserved
   b. define the variance $L_i^{(j)}, j = 1,2$, of as the quadratic form $\tilde{V}_i^{(j)} \equiv \tilde{w}_i^{(j)\prime} V_i^{(j)} \tilde{w}_i^{(j)}$, where $V_i^{(j)}$ is the covariance matrix between the constituents $L_i^{(j)}$ of the bisection, and $\tilde{w}_i^{(j)} = \text{diag}\left[V_i^{(j)}\right]^{-1} \frac{1}{\text{tr}\left[\text{diag}\left[V_i^{(j)}\right]^{-1}\right]}$, where diag[.] and tr[.] are the diagonal and trace operators
   c. compute the split factor: $\alpha_i = 1 - \frac{\tilde{V}_i^{(1)}}{\tilde{V}_i^{(1)}+\tilde{V}_i^{(2)}}$, so that $0 \leq \alpha_i \leq 1$
   d. re-scale allocations w_n by a factor of $\alpha_i, \forall n \in L_i^{(1)}$
   e. re-scale allocations w_n by a factor of $(1 - \alpha_i), \forall n \in L_i^{(2)}$
4. Loop to step 2

Step 3.b takes advantage of the quasi-diagonalization bottom-up, because it defines the variance of the partition $L_i^{(j)}$ using inverse-variance weightings $\widetilde{w}_i^{(j)}$. Step 3.c takes advantage of the quasi-diagonalization top-down, because it splits the weight in inverse proportion to the cluster's variance. This algorithm guarantees that $0 \leq w_i \leq 1, \forall i = 1, \dots, N,$ and $\sum_{i=1}^{N} w_i = 1,$ because at each iteration we are splitting the weights received from higher hierarchical levels. Constraints can be easily introduced in this stage, by replacing the equations in steps 3.c-3.e according to the user's preferences.

This concludes a first description of the HRP algorithm, which solves the allocation problem in deterministic logarithmic time, $T(n) = O(\log_2 n)$. Next, we will put to practice what we have learned, and evaluate the method's accuracy out of sample.

Backtest

We select the most liquid ETF tracking the major index in US markets as our testing data source. Exhibit 3 displays the tickers of selected ETFs and corresponding tracked index.

Daily adjusted close price of ETFs from 2007/01/01 - 2019/12/06 are extrapolated from Yahoo Finance for our final backtesting. For the convenience of recording results, we use some acronyms for each portfolio optimization method. Exhibit 4 shows the acronyms and meanings.

In-sample test

We apply optimization model on the performance of ETFs from 2012/01/01 - 2016/01/01 to output the optimal allocation of portfolio for testing and comparison. Results show that random weighted portfolio(RDM) outstands in annualized return and volatility, while Minimum-Variance portfolio(MVP) earns the highest Sharpe ratio. In the graph of Markov's efficient frontier, Hierarchical Risk Parity is closest to MVP. Exhibit 5(a) -5(c) show the comparison of annualized return, volatility and Sharpe ratio for all models.

Out-of-sample test

In the out-of-testing period, we adjust our portfolio allocation on the first trading day every month. The approach that we rebalance complies with the output (optimized weight) of models, fed with daily performance of ETFs in last 63 trading days. Then we just keep rebalancing throughout the whole period and calculate the cumulative return and volatility.

Stable period testing

We set the testing period from 2012/04/30 to 2019/12/06. Generally, a business cycle lasts about 8 years which matches our testing time horizon.

Results show that UMVP, allowing short position and leverage, earns the highest Sharpe ratio, followed by HRP, excluding the short and leverage position. It is unfair to directly compare them with different constraints. Exhibit 6(a) - 6(b) displays the comparison of annualized volatility and Sharpe ratio for all models.

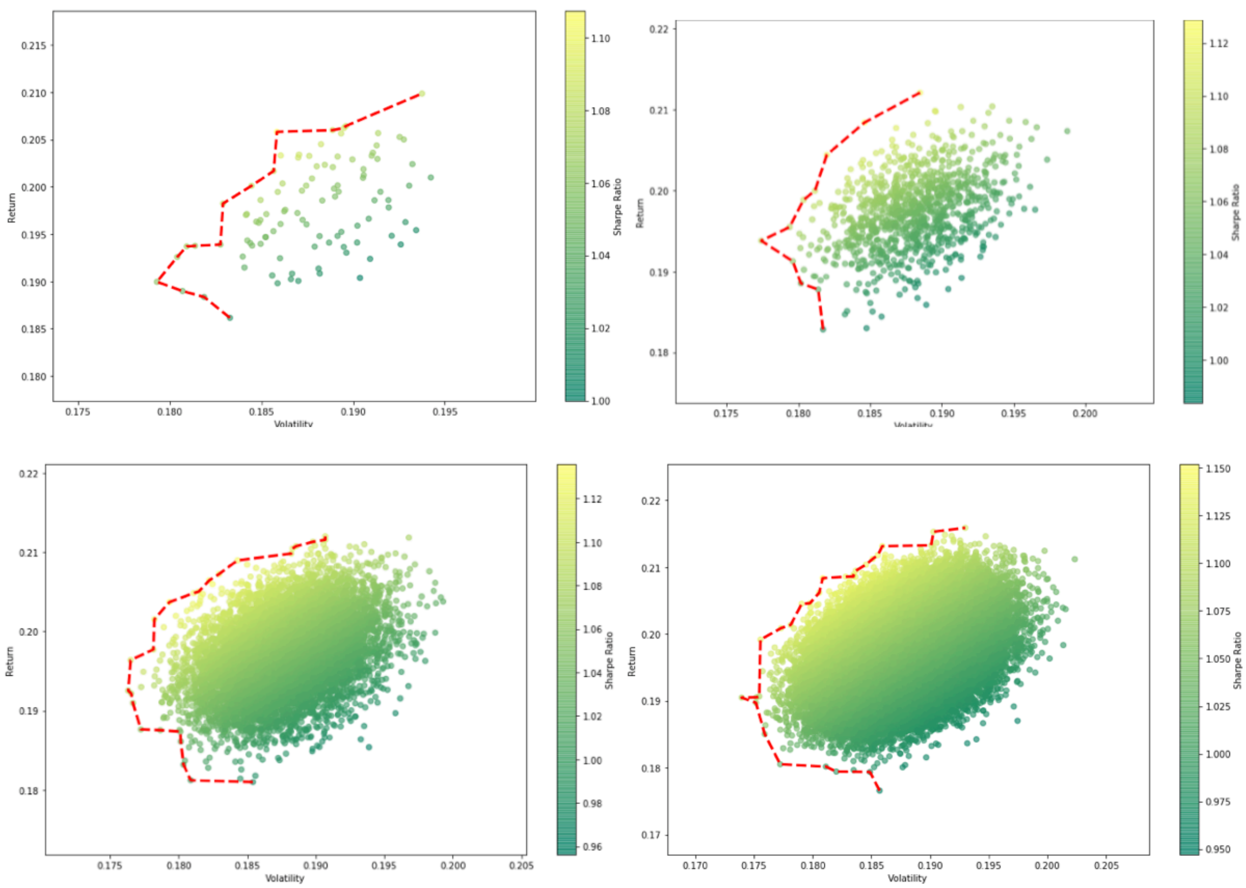
In the aspect of specific portfolio allocations, MVP generates portfolios far from what we defines as diversification, as it keeps betting all on one certain ETF, which had the lowest variance at that time. Additionally, there are some numerical problems with optimization caused by the instability of inverse covariance matrix. However, Inverse Variance, Hierarchical Risk Parity both allocate most weights to a single ETF, which is still not desirably diversified. Exhibit 7(a) - 7(d) shows the time series of allocations for each model.

Conclusion

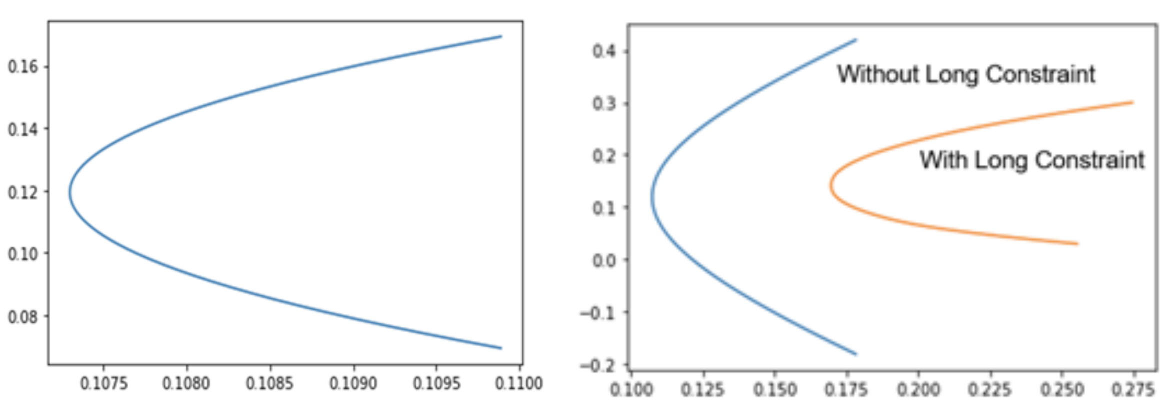
By now we have developed a machine learning technique of HRP which avoids the requirement of invertibility on covariance matrix. HRP model provides a more stable and efficient algorithm in optimizing portfolio construction. Though UMVP has better in-sample and out-of-sample results, it involves highly frequent rebalance and extremely high leverage which is undesirable in real practice. HRP generates a less favorable result compared with MVP in-sample, but outperforms MVP and most of other methods greatly out-of-sample in terms of return, volatility, Sharpe ratio and accumulated P&L. Furthermore it produces portfolios that are more stable in weights which lowers the possible transaction costs and trading fees.

There are a few more features that can be improved in this method: firstly under HRP's scenario only long positions are possible which neglects the potential lucrative opportunities of short positions. Also, the HRP method still creates portfolios that are not sufficiently diversified that weights can be greatly centralized on certain assets in periods of time.

Appendix



[EXHIBIT 1 HERE]



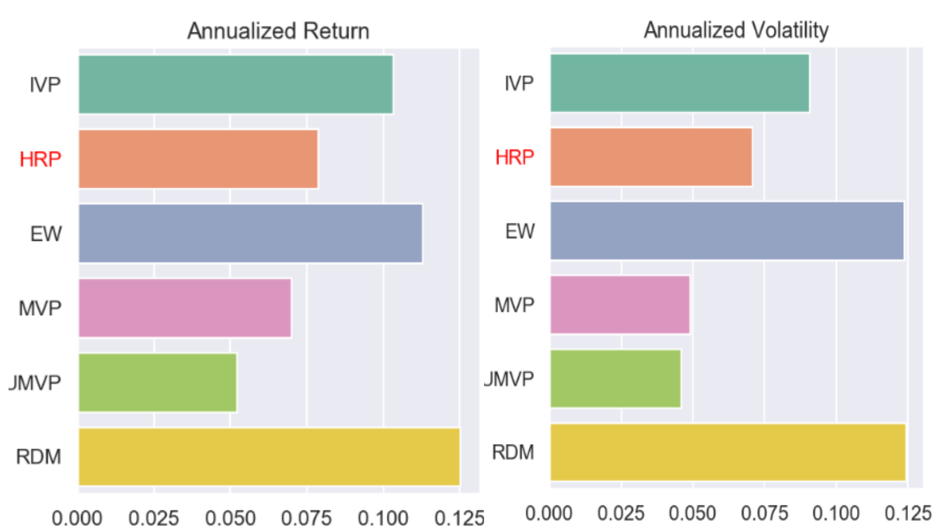
[EXHIBIT 2 HERE]

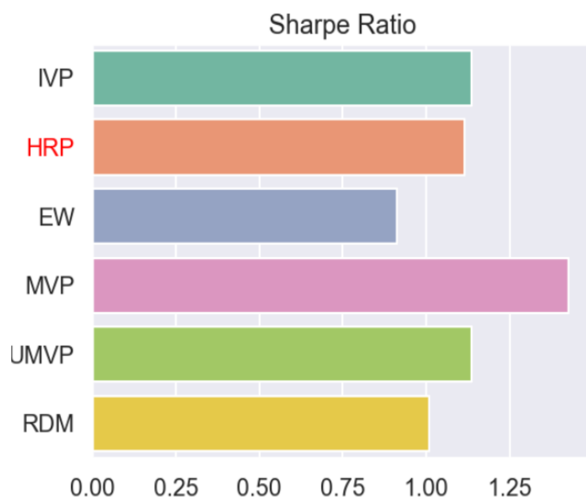
Symbol	Underlying Index
PFF	CE Exchange-Listed Preferred & Hybrid Securities Index
DIA	Dow Jones Index
FEZ	EURO STOXX 50 Index
IJH	S&P MidCap 400 Index
IWD	Russell 1000 Value Index
IWO	Russell 2000 Growth Index
IWF	Russell 1000 Growth Index
IWM	Russell 2000 Index
IWN	Russell 2000 Value Index
MDY	S&P 400 MidCap 400 Index
OEF	S&P 100 Index
QQQ	NASDAQ-100 Index
VTI	CRSP US Total Market Index
SPY	S&P 500 Index

[EXHIBIT 3 HERE]

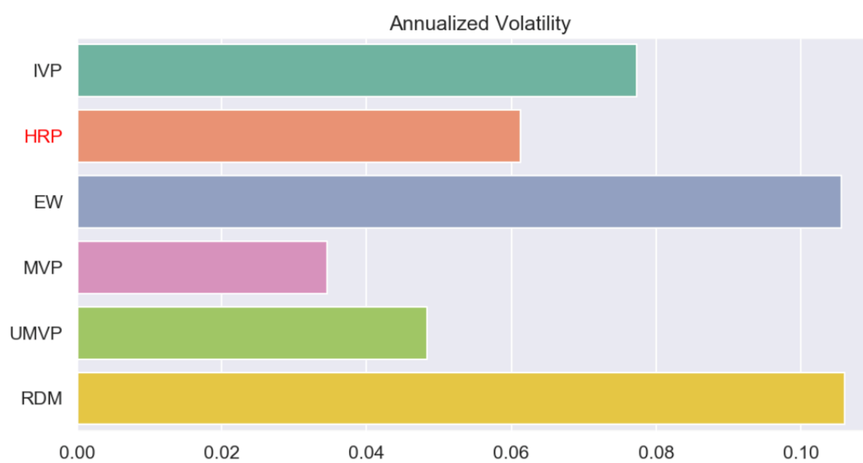
Acronyms	Model Name
IVP	Inverse Variance Portfolio
HRP	Hierarchical Risk Parity Portfolio
EW	Equal Weighted Portfolio
MVP	Minimum Variance Portfolio
UMVP	Unconstrained Minimum Variance Portfolio
RDM	Random Weighted Portfolio

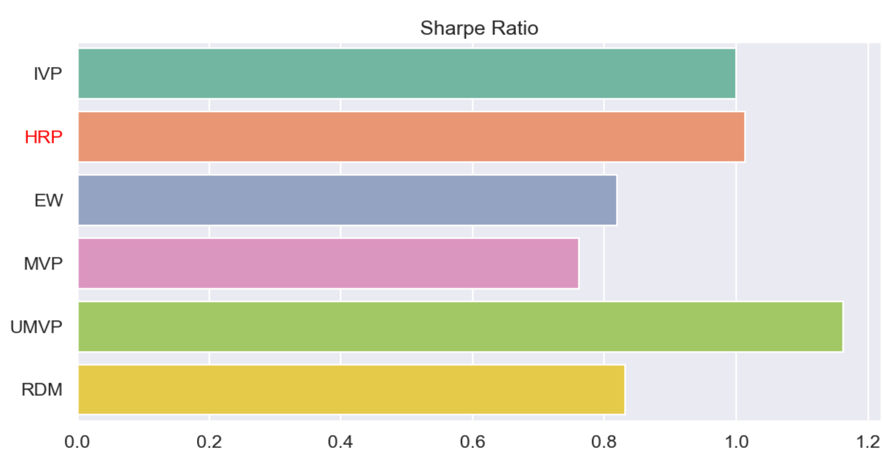
[EXHIBIT 4 HERE]



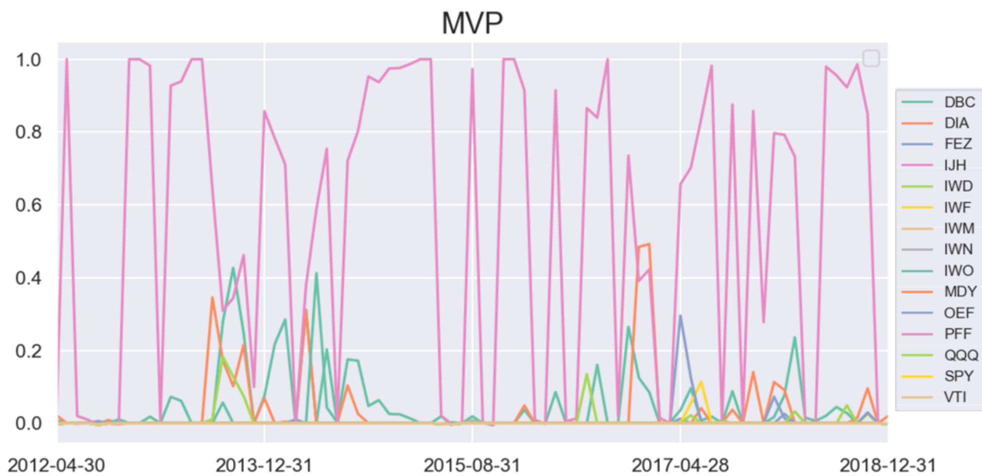


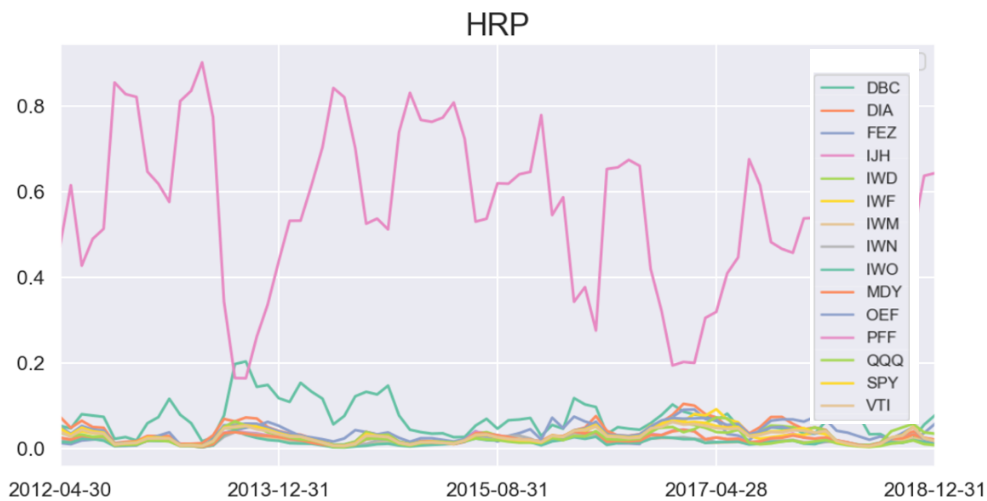
[EXHIBIT 5(a) – 5(c) HERE]

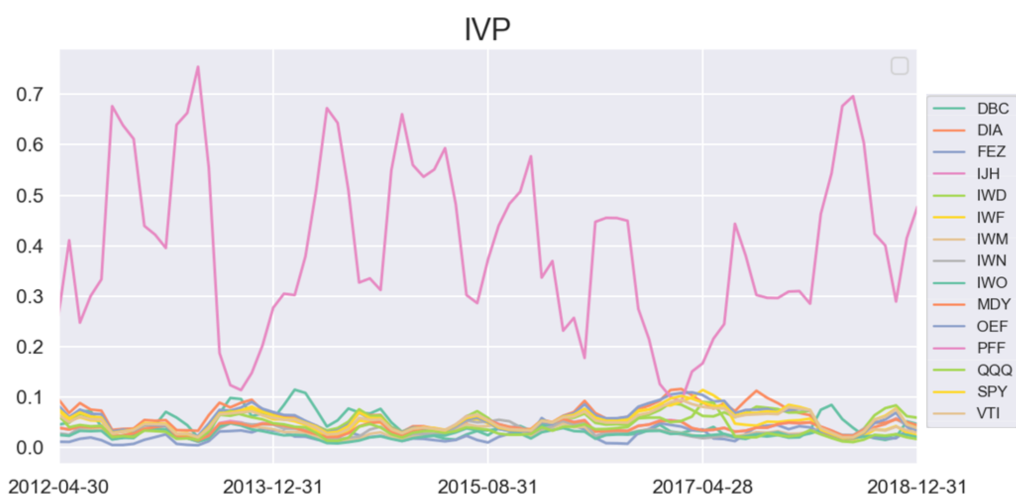


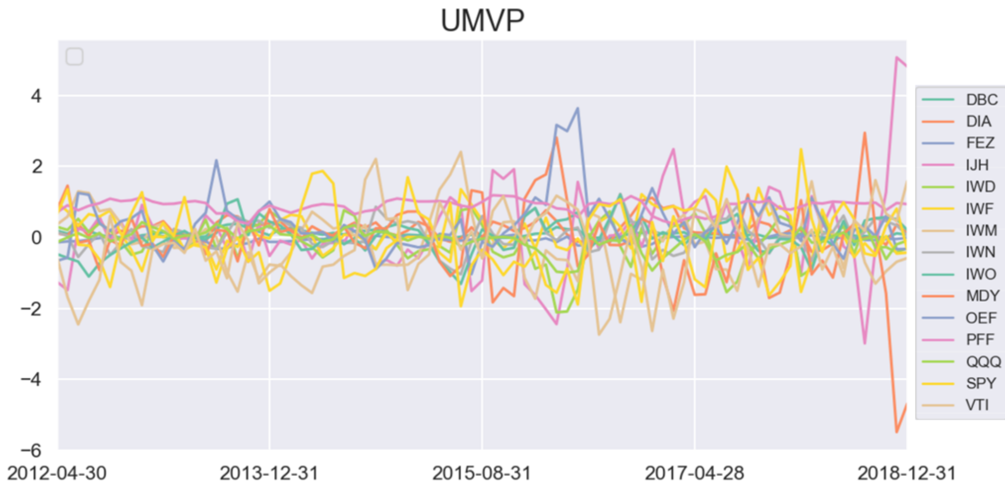


[EXHIBIT 6(a) – 5(b) HERE]


MVP
1.0
0.8
0.6
0.4
0.2
0.0
2012-04-30
2013-12-31
2015-08-31
2017-04-28
2018-12-31
DBC
DIA
FEZ
IJH
IWD
IWF
IWM
IWN
IWO
MDY
OEF
PFF
QQQ
SPY
VTI


HRP
0.8
0.6
0.4
0.2
0.0
2012-04-30
2013-12-31
2015-08-31
2017-04-28
2018-12-31
DBC
DIA
FEZ
IJH
IWD
IWF
IWM
IWN
IWO
MDY
OEF
PFF
QQQ
SPY
VTI


IVP
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2012-04-30
2013-12-31
2015-08-31
2017-04-28
2018-12-31
DBC
DIA
FEZ
IJH
IWD
IWF
IWM
IWN
IWO
MDY
OEF
PFF
QQQ
SPY
VTI



[EXHIBIT 7(a) – 7(d) HERE]

Reference

[1] de Prado M L. Building diversified portfolios that outperform out of sample[J]. The Journal of Portfolio Management, 2016, 42(4): 59-69.

[2] Bailey D, López de Prado M. An open-source implementation of the critical-line algorithm for portfolio optimization[J]. Algorithms, 2013, 6(1): 169-196.