Portfolio Rebalancing and Optimization

Austin Hochman, Kaya Ito Alpturer, Tanvi Kapoor

May 2021

1 Objective

Our project's goal was to identify the optimal portfolio rebalancing strategy in order to maximize returns.
To do this, we planned on researching popular rebalancing algorithms and selecting one to use in statistical tests. We planned on running tests with different rebalancing intervals and during periods of notable stock market growth and decline. Furthermore, we aimed to incorporate transaction costs into our model in order to provide a more accurate answer to the question: "how often should I rebalance my portfolio?"

2 Data Source

We made the decision to use ETFs to train our algorithm. This is because ETFs are inherently more stable than stocks. They are less prone to dramatic increases or decreases in their value. This means that there are fewer confounding variables at play, and any increase or decrease in a portfolio's value is more likely to be a reflection of the rate at which that portfolio was rebalanced.
We made the decision to use the standard 11 sectors of the stock market, and ETFs that represent them, as our data source. These 11 sectors are: Communication services, Consumer Discretion, Consumer Staples, Energy, Financials, Healthcare, Industrials, Information Technology, Materials, Real Estate and Utilities. We found data for S&P 500 sector indices with daily prices from the previous 10 years (2011-2021). This would provide the necessary data in order to derive the data to run our algorithm and to run our statistical tests.

3 Algorithm

The first rebalancing theory we considered was the traditional Markowitz portfolio theory. Markowitz portfolio theory utilizes a covariance matrix of returns to make its recommendations. However, it was discarded for the following reasons:

- A small deviation in the forecasted returns would cause it to return a very different portfolio, which is undesirable since returns are never fixed and may vary slightly in short periods of time. This happens because the method relies on the inversion of the positive-definite covariance matrix, so it fails if the matrix is ill structured
- It is subject to Markowitz' curse, according to which, the more correlated the investments, the greater the need for diversification, and yet the more likely we are to receive unstable solutions.
- A matrix only accounts for geometric relations, but is unable to account for hierarchal relations between assets. For example, Markowitz theory may assign Goldman Sachs'

assets and a holding in the Caribbean as substitutes of one another based one their correlation, which is an inaccurate representation. Thus, the assignment of weights is highly unreliable, which leads to its instability

To overcome these shortcomings, we decided to explore another algorithm:

Hierarchical Risk Parity (HRP) Algorithm

This algorithm has all the benefits of the Markowitz Portfolio Theory. It also does not need the covariance matrix to be invertible or positive-definite, and it accounts for hierarchal relations between investments
The output of this algorithm is in the form of a recommended portfolio distribution, i.e. it gives the proportion (or weights) of each sector that should be contained in the portfolio to maximize returns, as shown in the graph below:

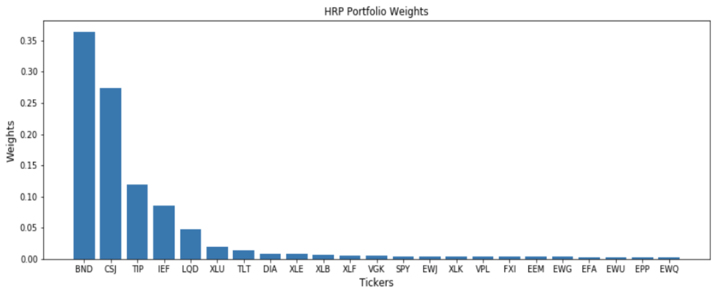

Figure 1: Representation of HRP Algorithm

Structure of the Algorithm

- Hierarchical Tree Clustering
- Matrix Seriation
- Recursive Bisection

1. <u>Hierarchical Tree Clustering</u>

The idea behind hierarchical clustering is that at each iteration, items with minimum statistical differences are combined into clusters. This continues until all the data is merged into one single cluster. This merging can be visualized in the dendrogram below:

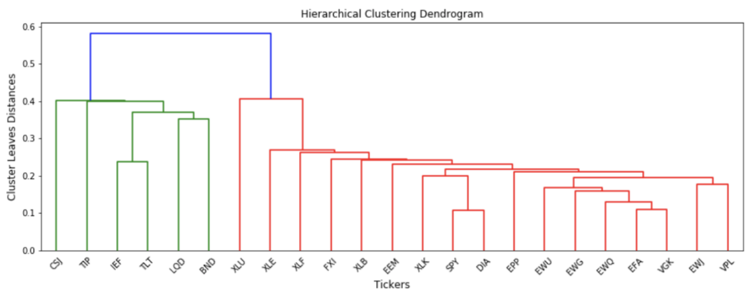

Figure 2: Dendrogram representing Hierarchichal Tree Clustering

Here, to begin with, a correlation matrix ρ of the ETFs of the 11 sectors is calculated, from a matrix that comprised the price of the ETFs over time. From this correlation matrix, a correlation-distance matrix D is calculated such that $D(i,j) = \sqrt{0.5 * \rho(i,j)}$. Next, the Euclidean distance between all columns is calculated in a pair-wise manner to determine the similarity for clustering. This produces a matrix of the form $\bar{D}$, where $\bar{D}(i,j) = \sqrt{\sum_{n=1}^{N}(D(n,i) - D(n,j))^2}$. Using $\bar{D}$, we start forming clusters recursively, by identifying the minimum Euclidean distance, i.e. the first cluster is $U[1] = argmin_{i,j}(\bar{D}(i,j))$, where U is the set of clusters. Finally, $\bar{D}$ is updated to reflect this clustering, i.e. if a and b are the 2 stocks combined in the previous step, then $\bar{D}$ is updated as $\bar{D}(i, U[1]) = min(\bar{D}(i,a), \bar{D}(i,b))$. This continues until a single cluster is formed.

2. Matrix Seriation

Matrix seriation is a rearranging technique that is used to place investments with greater covariance closer to one another, so as to clearly demarcate investments that are more similar from the ones that are more dissimilar. This is shown in the figures below. The algorithm uses the order of clusters formed above for this grouping.

Figure 3: Representation of Matrix Seriation, indicating demarcation of assets

As seen in the graphs above, the distribution of colors in the matrix is random in the 'Unclustered correlations' diagram, indicating that the distribution of assets is not affected by their covariance. In the 'Clustered correlations' diagram however, the assets with lighter colors are closer to the diagonal, indicating a clustering structure, grouped according to covariance.

3. Recursive Bisection

This step is the final step, assigning weights to the assets in the portfolio. To do this, it uses the property of the diagonal covariance matrix determined in the step above, which is that inverse-variance allocations of a diagonal covariance matrix are optimal. It assigns weights in the following way:
The hierarchical tree clustering step above forms binary trees, with sub-clusters V_1 and V_2. For each of these sub-clusters, the variance is calculated, by the formula $V_{adj} = w^T V w$. Here, $w = \frac{diag[V]^{-1}}{trace(diag[V]^{-1})}$. This computation is justified by the property mentioned above. Next, a weighing factor is determined based on these values of V_1 and V_2: $\alpha_1 = 1 - \frac{V_1}{V_1+V_2}$ and $\alpha_2 = 1 - \alpha_1$. Finally, the weights of the stocks in the respective clusters are updated by doing $W_1 = \alpha_1 * W_1$ and $W_2 = \alpha_2 * W_2$. These steps are repeated recursively on all of the subclusters determined in hierarchical tree clustering, until weights are assigned to all the assets.

4 Results and Analysis

We ran the initial analysis for a period of 9 years from October 2012 to March 2021. The HRP algorithm is fed 6 months of prior market data for every rebalancing action. In the end, the results can be seen in Figure 4. Our observations confirm the hypothesis that rebalancing more often is more optimal as increasing rebalancing period decreased average return.

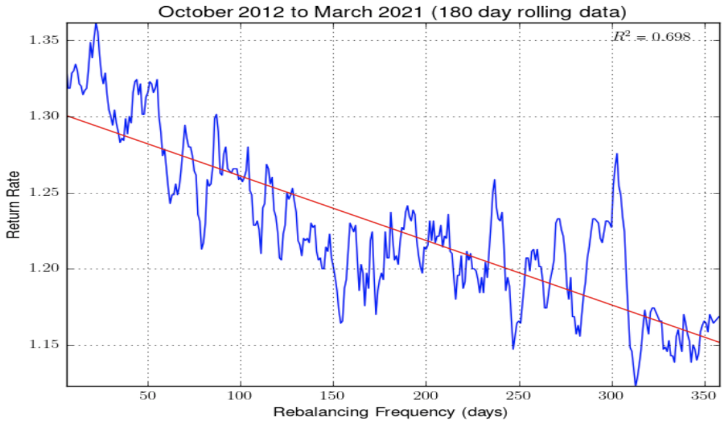

Figure 4: Rebalancing over 9 years

The next analysis was run on a period of two and a half years from December 2012 to May 2015 with 6 months rolling data fed to the algorithm in Figure 5. We chose these dates to evaluate rebalancing frequencies on a period where the market has seen steady growth. With steady growth, rebalancing more frequently was hypothesized to be sub-optimal since the percentage of high growth stocks are reduced in rebalancing to reduce risk. With steady growth, these stocks will continue to perform well and any risk mitigation should decrease returns. However, the opposite was observed since overall growth in the market does not imply that individual stocks are also steadily growing, and the portfolio is still volatile enough to make the act of rebalancing more frequently, optimal.

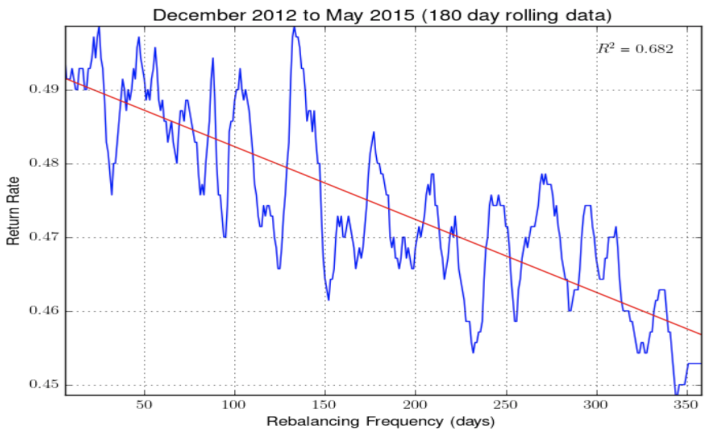

Figure 5: Rebalancing over 2.5 years

The last analysis we ran incorporated transaction costs since they constitute the primary reason for not rebalancing more frequently. Our model incorporated transaction costs as 1% of the absolute change in the price of stocks we buy/sell when rebalancing. For the number of shares s_p prior to rebalancing and the number of shares after rebalancing as s_a, and price of the share

p_s we calculate the transaction cost added as:

$$\frac{p_s|s_p - s_a|}{100}$$

The resulting trend is best described by a 2nd degree polynomial which peaks around the 200 day mark. This analysis clearly demonstrates how the marginal gain from rebalancing early is not comparable to the fees paid for the transaction if the reabalancing is too frequent.

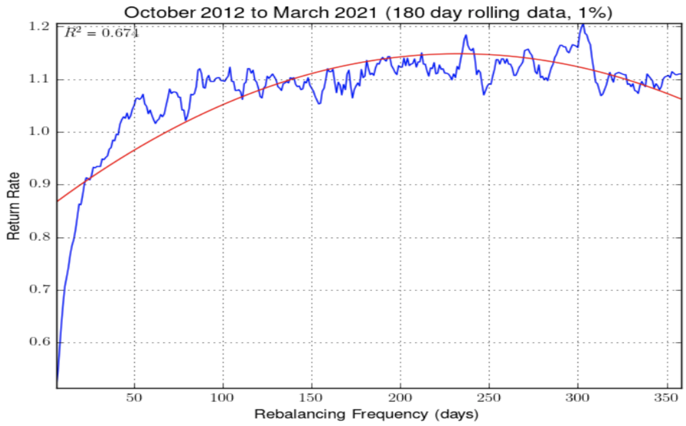

Figure 6: Rebalancing with a Transaction fee model

5 Conclusion and Further Scope

Thus, based on our algorithm, we can conclude that even for an asset such as the chosen indices that are not nearly as volatile as a portfolio directly consisting of stocks, the market is still volatile enough to make frequent rebalancing more optimal. This controlled experiment on SP indices indicates that higher returns can be generated by frequent rebalancing. The last model that included transaction costs demonstrates why this optimal behavior cannot be attained in real life and shows that analysis on determining the optimal frequency is required on a portfolio and market specific context.

Limitations

Our algorithm looks at indices instead of stocks for asset representation. However, portfolio rebalancing is more effective when done on stocks, since stocks are more volatile whereas indices inherently account for some amount of risk mitigation. Thus, the performance of the rebalancing algorithm may not be as good on ETFs.

Further Scope

- We presented a generalized model for transaction cost, but different platforms have different transaction costs and different ways of calculating them, so we can modify the model to suit each individual platform
- As mentioned above, we can try running the rebalancing algorithm on stocks instead of ETFs, to see if the performance is better, or if the insights gained are more beneficial

- While we only looked at one rebalancing algorithm (HRP) in this paper, we can try out different rebalancing algorithms to then determine the optimum, possibly dependent on the data source used
- We can also use this algorithm and the rebalacing strategies to look at different economic periods (inflation, deflation, recession etc.), and then determine the best strategy and behavior for each such period

6 References

Datasources:

1. Sectors - Samp;P Dow Jones Indices. (n.d.). https://www.spglobal.com/spdji/en/landing/investment-themes/sectors/.

Algorithm and Images:

2. The Hierarchical Risk Parity Algorithm: An Introduction. Hudson amp; Thames. (2021, January 11). https://hudsonthames.org/an-introduction-to-the-hierarchical-risk-parity-algorithm/.
3. Lopez de Prado, M. (2019, July 17). Building Diversified Portfolios that Outperform Out-of-Sample. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract$_id = 2708678$.