COMP6212 Assignment 1 : Shakib-Bin Hamid 25250094 sh3g12

1 Efficient Frontier

Let a portfolio of N assets be $\boldsymbol{\pi}$, whose expected return is $\boldsymbol{\mu}$ and the co-variance is $\boldsymbol{\Sigma}$.

1.1 Efficient Frontier with 3 Assets

According to the paper [1], the expected return of the portfolio, $E = \sum_{i=1}^{N} \pi_i \mu_i = \boldsymbol{\pi}^t \boldsymbol{\mu}$. The risk is analogous to the variance of the returns, i.e. $V = \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{ij} \pi_i \pi_j = \boldsymbol{\pi}^t \boldsymbol{\Sigma} \boldsymbol{\pi}$.

Given $\boldsymbol{\mu} = \boldsymbol{m}$ and $\boldsymbol{\Sigma} = \boldsymbol{C}$ for a 3 assets, we can generate 100 random portfolios, where each portfolio $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3)^t$ s.t. $\mathbf{1}^t \boldsymbol{\pi} = 1$ by `y=randn(3,1); y=y/norm(y,1)`. Then we can calculate $E - V$ for each of the portfolios by `E=y'*m; V=sqrt(y'*C*y)`.

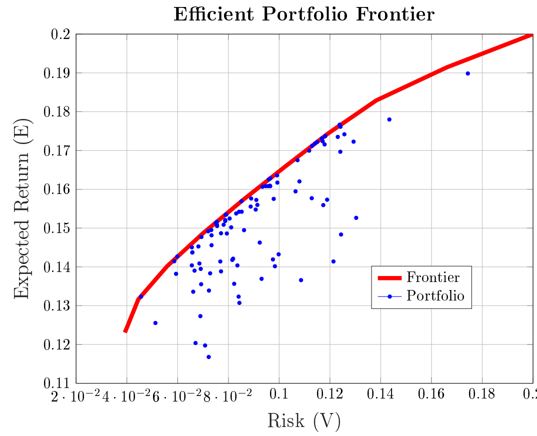


Figure 1: Efficient Portfolio

Finally I make the scatter plot and on the same figure I plot the efficient frontier using `estimateFrontier` function. As expected all the random portfolios were on the correct one side of the frontier.

1.2 Efficient Frontier with 2 Assets

To prepare three 2 asset porfolios, I removed the data points that are not necessary, i.e. that has the third asset. First I plotted random returns generated by the 2 asset mean and variance using `mvnrnd`. As can be noticed from Figure 2, asset 2 and 3 are almost uncorrelated, asset 1 and 2 are negatively correlated, asset 1 and 3 are positively correlated.

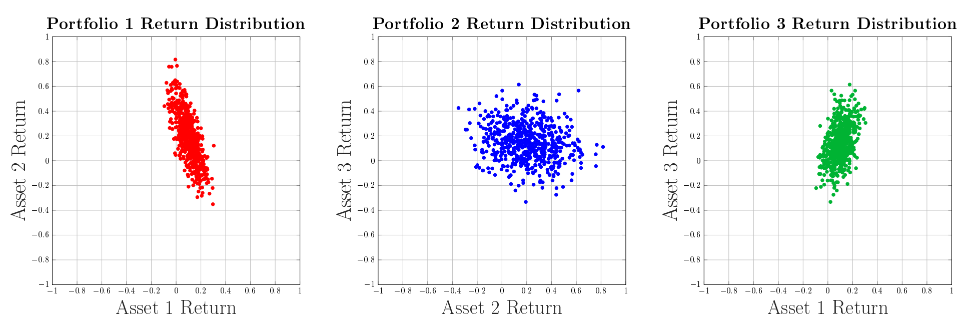


Figure 2: Distribution of 2 Asset Returns

As done previously with all three assets, I generate 100 random portfolios for each of the three 2 asset combinations and plot the $E - V$ scatters along with the efficient frontiers. Notice that in case of 2 asset portfolios, every portfolio construction is efficient and the frontier has a bend, i.e. risk increases for the lowest returns. The reason for all portfolios lying on the efficient porfolio is as follows -

Represent porfolio for two assets as $\boldsymbol{\pi} = (\pi, 1 - \pi), \boldsymbol{\mu} = (\mu_1, \mu_2), \boldsymbol{\Sigma} = (\sigma_1, \sigma; \sigma, \sigma_2)$. Then $E = \pi\mu_1 + (1 - \pi)\mu_2, V = \pi^2(\sigma_1 + \sigma_2) + 2\pi(\sigma - \sigma_2) + \sigma_2$. The second derivative of both are devoid of π, because of this reduction of degree of freedom. So, the $E - V$ will always be the extremum regardless of the portfolio.

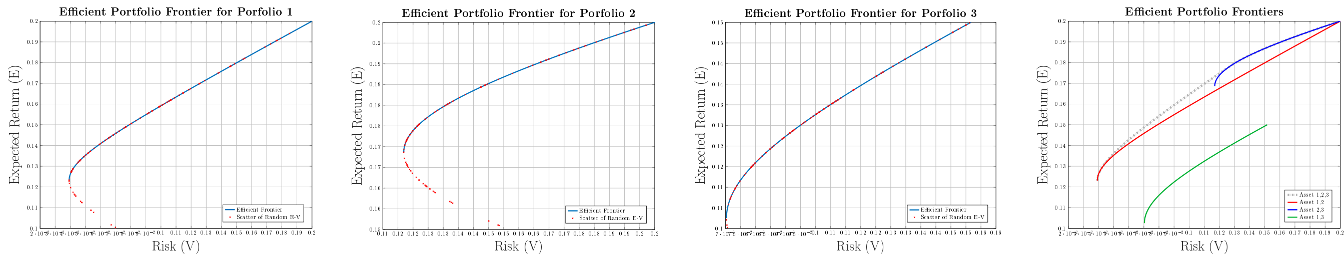


Figure 3: Efficient Frontier for 2 Asset Portfolios

1.3 Use of `linprog` in NaiveMV

In order to calculate the efficient frontier, I needed two extreme points - maximum return for a portfolio regardless of the risk and minimum risk regardless of the return. For the first case, $E = \underset{w}{max}(\boldsymbol{\pi}^t\boldsymbol{\mu})$ s.t. $\mathbf{1}^t\boldsymbol{\pi} = 1$ which gives the portfolio that maximises the return regardless of the risk. We can then calculate $E - V$ for this portfolio, thus giving us the top corner of the $E - V$ graphs here. This is a linear equation of $\boldsymbol{\pi}$. Matlab's `linprog` function can solve linear equation of the following form -

$$\underset{x}{min}(f^t x) s.t. \begin{cases} A.x \leq b \\ A_{eq}.x = b_{eq} \\ lb \leq x \leq ub \end{cases}$$

```
f=-ERet;A=[];b=[];Aeq=ones(1,N);beq=1;lb=0;ub=1
```

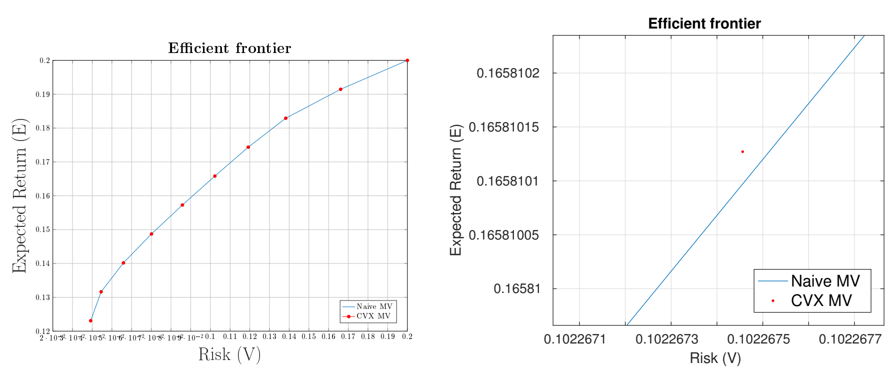
However, to calculate the portfolio that minimises the risk we need to solve a quadratic equation of $\boldsymbol{\pi}$, $\underset{w}{min}(\boldsymbol{\pi}^t\boldsymbol{\Sigma}\boldsymbol{\pi})$ s.t. $\mathbf{1}^t\boldsymbol{\pi} = 1$. In this case we use the `quadprog` function. Finally, we choose N expected portfolio returns between the two extreme points and calculate the porfolio that minimises the risk while achieving the chosen expected returns. Thus the efficient portfolio is created.

1.4 Efficient Frontier : NaiveMV vs CVX

Using the CVX tool we can declarively perform the convex optimisations we performed earlier with `linprog` and `quadprog`, as follows -

```
cvx_begin quiet
        variable w(N,1)
        minimize( -ERet'*w )
        subject to
                ones(1,N)*w == 1;
                w >= zeros(N,1);
cvx_end
```

```
cvx_begin quiet
        variable w(N,1)
        minimize( 0.5*w'*ECov*w + zeros(N,1)'*w)
        subject to
                ones(1,N) * x == 1;
                x >= zeros(N,1);
cvx_end
```



(a) Similarity (b) Difference

Figure 4: NaiveMV vs Using CVX

The results are extremely similar, since differences only show up in 10^{-7} scale. However, the CVX tool was noticeably slower.

2 Markowitz vs Naive $1/N$ Strategy on FTSE 100

2.1 Getting FTSE 100 Data

I downloaded the FTSE 100 index and the top 30 most traded companies' data over the last 3 years using a `bash` script. Then I used a `ruby` script to fill in the missing days (which may be weekends) with the previous day's data. As a result all data had the same number of rows. The returns will be calculated based on the adjusted close values. Figure 5 plots the data.

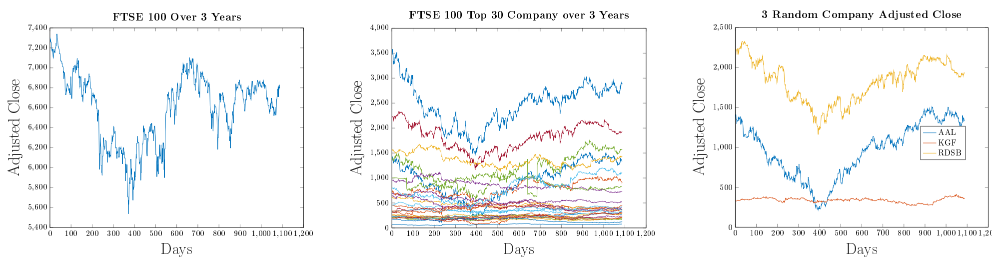


Figure 5: FTSE 100, Top 30 Most Traded Company and Selected 3 Company Adjusted Close over 3 Years

2.2 Returns and Efficient Portfolio of 3 Random Assets

I started by calculating the returns as percentage. I defined the return in two ways -

- $return(i+1) = (val(i+1) - val(i))/val(i+1)$, return based on daily investment
- $return(i+1) = (val(i+1) - val(1))/val(i+1)$, return based on first investment

I expected portfolio returns in first definition to be jittery in a small window since daily returns go up and down. Whereas the second definition would be smoother, although it implicitly imposes a correlation. Figure 6 and 8 build a portfolio out of **all** 30 assets, whereas Figure 7 and 9 select **3** random assets, which were controlled using `rng(1)`. The resultant assets were AAL, KGF and RDSB. Their adjusted closing values were plotted in Figure 5.

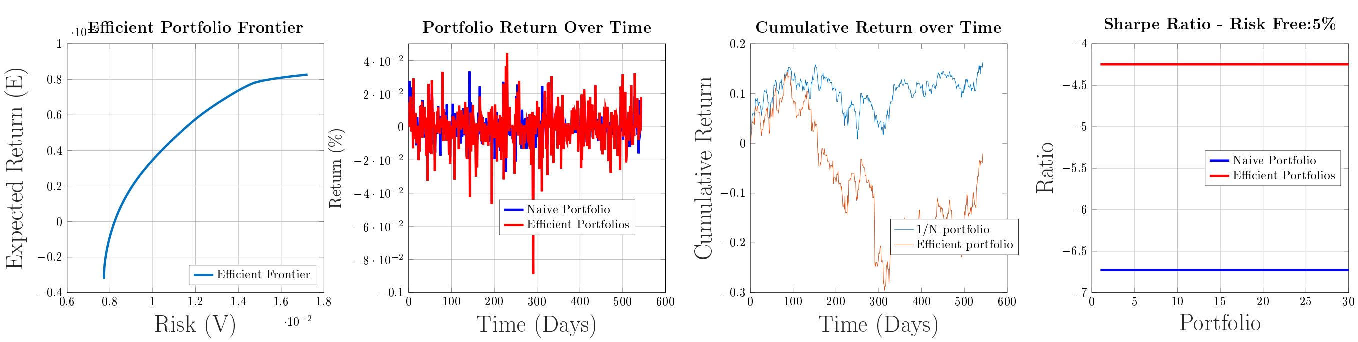


Figure 6: Portfolio of **all** 30 assets analysis where returns are based on **daily** change

As expected, the portfolio returns over time were moving between ± 0.02 if we calculated returns based on daily investment. It was smoother when returns were calculated based on the first investment. The efficient frontier in both cases were of expected shape.

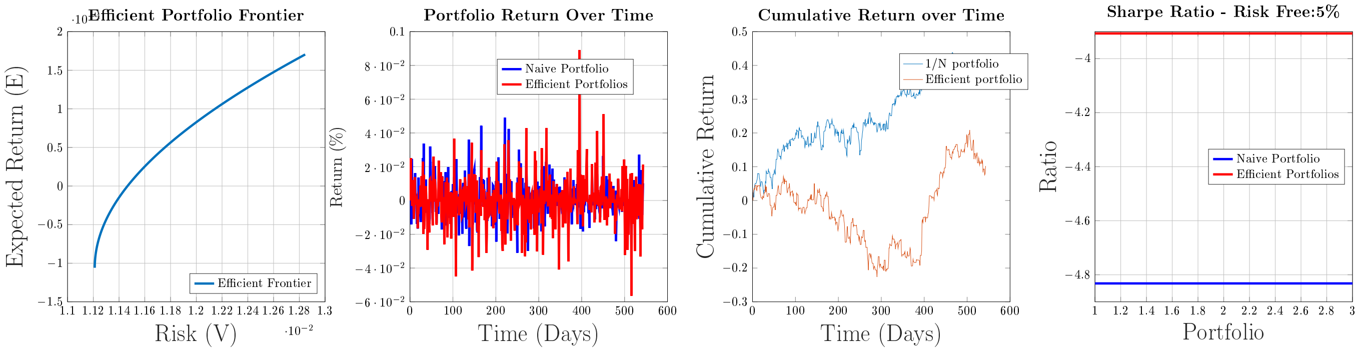


Figure 7: Portfolio of **3 random** assets analysis where returns are based on **daily** change

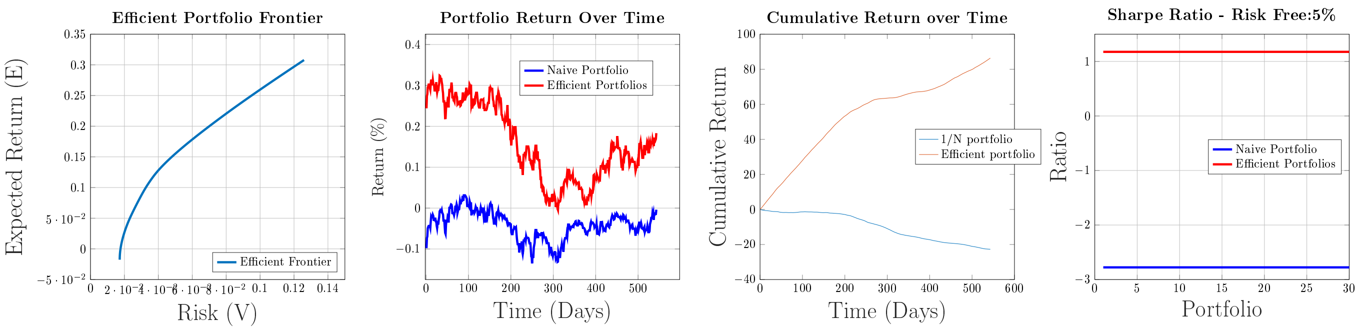


Figure 8: Portfolio of **all** 30 assets analysis where returns are based on **first return**

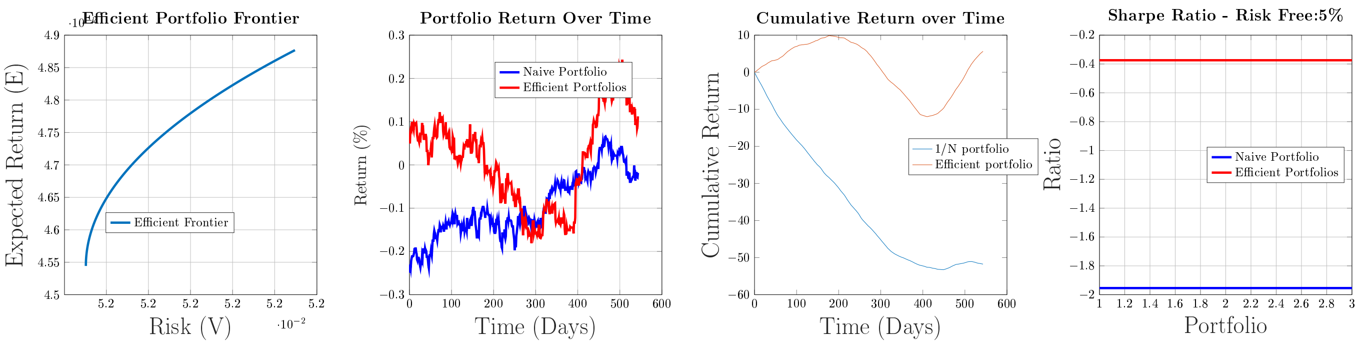


Figure 9: Portfolio of **3 random** assets analysis where returns are based on **first return**

2.3 Comparison with Naive Strategy

After chronologically sorting the data, I picked the first half of the data as the training set and the rest as the validation set. Using the first half, I calculated the Expected Return as the mean and the Risk as the covariance using `m = mean(returnsTrain);C = cov(returnsTrain);`. Then Financial Toolbox calculated the efficient frontier. Then I picked $V_{max}/2$ to be max risk and calculated the weights, which I picked as the Markowitz portfolio for comparison. `1/N*ones(N,1)` was the naive 1/N strategy portfolio.

I started the comparison by calculating the returns over the validation set. I used the Markowitz portfolio selected from the training data and the 1/N portfolio stated above -

```
effReturn = returnsTest * efficientWeights';
naiveReturn = returnsTest * naiveWeights';
```

I plotted them in the second subfigures in the figures in this section. In this particular selection of assets, the markowitz portfolio beat the naive one. However, it was not the case every time. On some runs of the experiment, the naive strategy was giving better returns on the graph.

Then I calculated the sharpe ratio, defined as $r = (m - r_0)/\sigma$, where r_0 is the risk free return. I set `riskFree = 0.05` and calculated the following -

```
naiveSharpe = (mean(naiveReturn) - riskFree)/std(naiveReturn);
effSharpe = (mean(effReturn) - riskFree)/std(effReturn);
```

The sharpe ratio is plotted in the 4th subfigures in the figures in this section. Surprisingly the sharpe ratios in both cases were often negative, meaning that risk free assets, if it returned 5%, would be better. The reason for this is that the assets actually performed poorly in adjusted close values during this time, as seen in Figure 5. Unsurprisingly, if we set $r_0 = 0$, i.e. no returns are risk free, then the sharpe ratios are positive. Also, a larger sharpe ratio can suggest a better performing portfolio. So, whichever portfolio showed better returns on the 2nd subfigures had the larger sharpe ratio on the 4th subfigures. I think the sharpe ratio is a reasonable measure of performance, but it depends largely on the estimate of the risk free assets like - government bonds etc.

3 Index Tracking

Since the index is improving over the years, a passive investor will likely want to invest in the index as a whole. However, investing in a large number of companies comes with the problem of tackling transaction costs. So, it is desirable to invest in a subset of the index assets that, as combination, closely matches the performance of the index itself. I investigated two ways of selecting one fifth of the N assets that represent the index. I first selected =the first half of the data as training data and the rest as the validation data.

3.1 Greedy Forward Search Asset Selection

During each iteration of the greedy algortithm, I go through each of the unselected assets and try to find the most effective linear combination containing it and the assets already chosen. The Root Mean Squared Error (RMSE) is then recorded for that asset combination. At the end of the iteration, I choose the asset combination that produced the least RMSE. Finally, after $N/5$ iterations, I find the overall desired subset of assets. The algorithm is described in Algorithm 1.

Algorithm 1 Greedy Asset Selection for Index Tracking

S = {} // the desired subset of assets
for $i \in \{1, ..., N/5\}$ **do** // N = number of total assets
  $e = \{\}$ // RMSE vector
  for $j \in \{1, ..., N\}$ **do**
    if $j \in \{S\}$ **then**
      $e(j) = \infty$ // asset already chosen
    else
      $s = S \bigcup \{j\}$ // temporary subset containing current asset
      $\boldsymbol{R_t} = \boldsymbol{R}(:, s)$ // training set asset return
      $\boldsymbol{w} = \min_{\boldsymbol{w}}(||\boldsymbol{w}'\boldsymbol{R_t} - \boldsymbol{y_t}||_2) s.t. \mathbf{1}^t\boldsymbol{w} = 1, \boldsymbol{w} \geq \mathbf{0}$ // $\boldsymbol{w}$ = weights, $\boldsymbol{y_t}$ = training set index return
      $e(j) = \sqrt{mean(||\boldsymbol{r_t} * \boldsymbol{w'} - \boldsymbol{y_t}||_2^2)}$
  $idx = minIdx(e)$ // index of the portfolio with minimum RMSE in this iteration
  $S = S \bigcup \{idx\}$

During the asset selection algorithm, the RMSE is calculated based on the training set, since we want to only fit the training data. Once the assets are determined, we find the effective portfolio based on only the training data and apply it on the validation data to see how well it performs. As can be seen from Figure 10, the representative index closely matches the training index, and deviates more on the validation data.

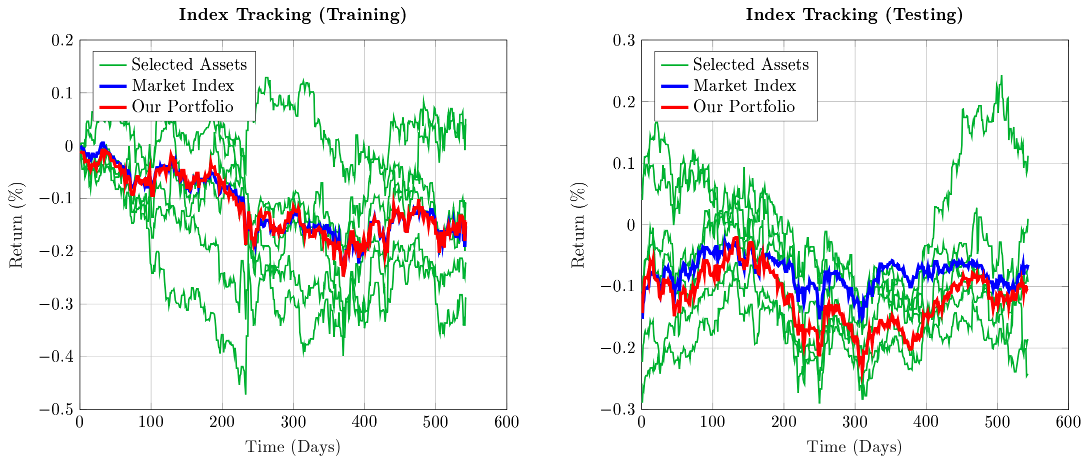


Figure 10: Greedy Index Tracking

The RMSE for the portfolio chosen by the greedy algorithm is plotted in Figure 13b . Selected assets and weights can be found in Figure 13b.

3.2 Sparse Portfolio for Index Tracking

We perform the greedy search so that we don't have to search the entire combinatorial space. However, any markowitz porfolio construction can be ill conditioned, if there is significant correlation between the assets. So, I introduce a l_1 regulariser term $\tau||\boldsymbol{w}||_1$ with the alternative, but equivalent optimisation problem, $\underset{\boldsymbol{w}}{min}(||\boldsymbol{w}'\boldsymbol{R} - \rho\mathbf{1}||_2)$ in order to stabilise the optimisation problem. Then the top $N/5$ most contributing assets are chosen and the optimisation is run again to redistribute the weights over the chosen assets. As before, the weights are chosen to fit the training index and then applied to the validation set to compare with the actual validation index. The results are as expected as seen in Figure 11. It is interesting that if we do not redistribute the weights over the assets, and rather only invest a fraction of our total wealth, it produces better performance on the validation set (Figure 12 and 13a)!

Algorithm 2 Sparse Asset Selection for Index Tracking

$\boldsymbol{w} = \underset{\boldsymbol{w}}{min}(||\boldsymbol{w}'\boldsymbol{R_t} - \boldsymbol{y_t}||_2 + \tau||\boldsymbol{w}||_1) s.t. \mathbf{1}^t\boldsymbol{w} = 1, \boldsymbol{w} \geq \mathbf{0}$

$S = sortIdx(w)$

$S = S(1 : N/5)$

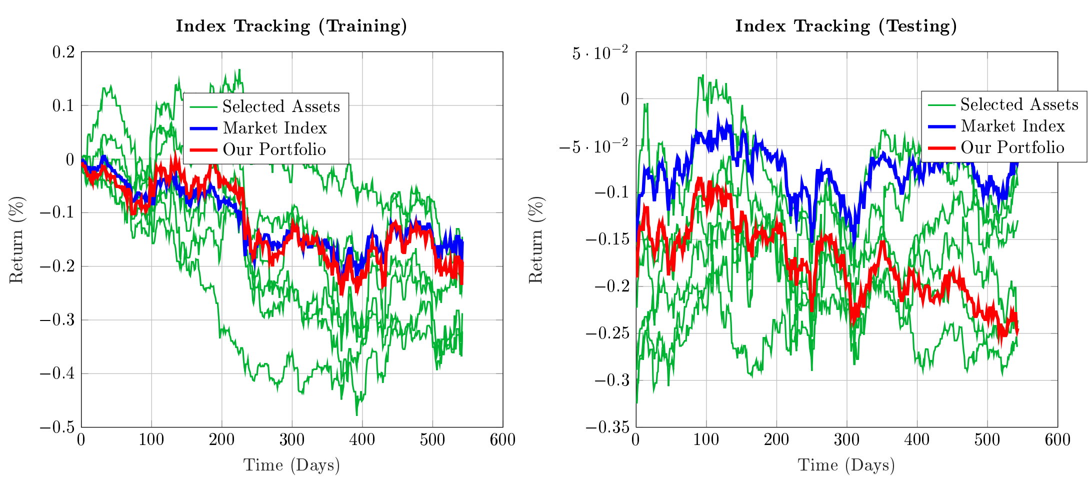


Figure 11: Sparse Index Tracking

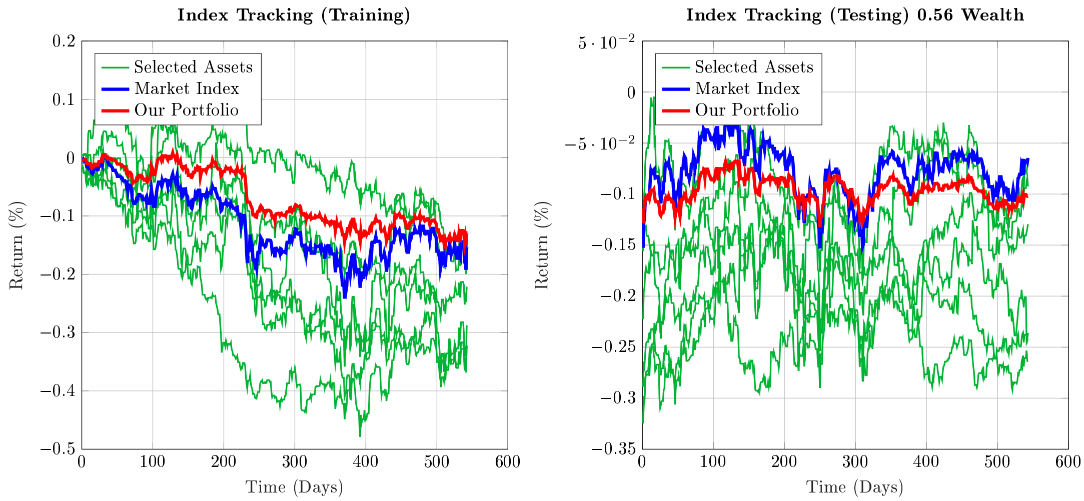
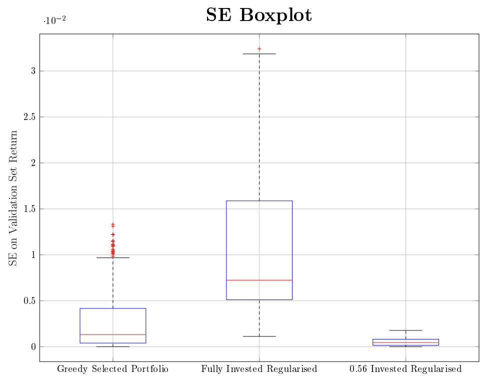
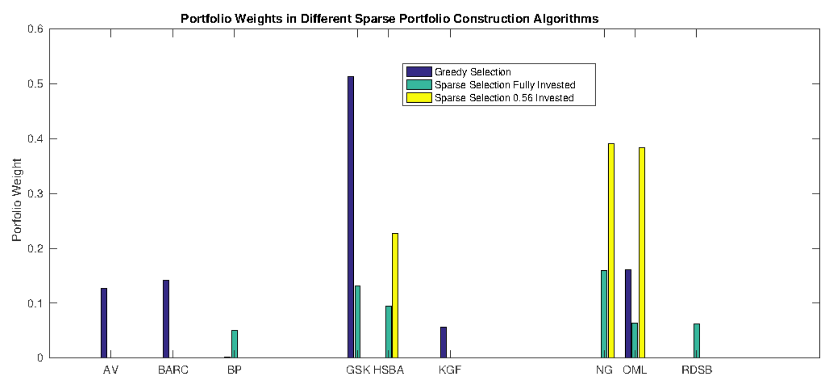


Figure 12: Sparse Index Tracking without Redistribution after Asset Selection

The squared error for the portfolio chosen by the regulariser is plotted in Figure 13a. Selected assets and weights can be found in Figure 13b.



(a) SE for Different Portfolio Selection Processes



(b) Portfolio Weights in Sparse Portfolios

Figure 13: Greedy and Regulariser Asset Selection Analysis

Whether the sparse portfolio using the regulariser is preferred over using the greedy algorithm is more a matter of art and prior knowledge. The idea is that a sparse portfolio is desired, but the way to achieve it can be different. Tuning τ is crucial and here we only tune it to match the number of selections to the number chosen by the greedy algorithm so that we can compare on the same grounds. However, in this exercise it was quite unexpected that changing τ to very high or very low numbers did not affect the weights much and indeed I had to choose the most contributing assets by sorting the weights. In this exercise, it seems that l_1 regulariser where only 56% of the assets were invested produces the best predictions on validation set.

4 Discussion on Lobo (et al.)'s Paper

Lets say, the current holding or wealth of n assets is $\boldsymbol{w} = (w_1, w_2, ..., w_n)$. $\boldsymbol{x}^+ = (x_1^+, x_2^+, ..., x_n^+)$ is the dollar amount bought for each asset and $\boldsymbol{x}^- = (x_1^-, x_2^-, ..., x_n^-)$ is the dollar amount sold for each asset. This $\boldsymbol{w} + \boldsymbol{x}^+ - \boldsymbol{x}^-$ asset is held for a period. After the period the resultant wealth is W and returns are $\boldsymbol{a} = (a_1, a_2, ..., a_n)$. The expected return after the period is $\bar{a}^t(\boldsymbol{w} + \boldsymbol{x}^+ - \boldsymbol{x}^-)$, which we will want to maximise. This is the objective function in the example in the paper's Section 1.6.

Buying and selling the assets will incur transaction costs, which lets say is $\phi(\boldsymbol{x})$, where $\boldsymbol{x}$ represents the amount sold or bought. The exchange of assets will generate $\mathbf{1}^t(\boldsymbol{x}^+ - \boldsymbol{x}^-)$, which will cover some of the transaction costs, if not all (self-financing). If the transaction cost is a linear function of $\boldsymbol{x}$, i.e. $\sum_{i=1}^{N}(\alpha_i^+ x_i^+ + \alpha_i^- x_i^-)$, then we can express the constraint as $\mathbf{1}^t(\boldsymbol{x}^+ - \boldsymbol{x}^-) + \sum_{i=1}^{N}(\alpha_i^+ x_i^+ + \alpha_i^- x_i^-) \leq 0$. This is the first constraint in the example. $x_i^+ \geq 0, x_i^- \geq 0$ represents that buying and selling amount is, obviously, non-negative.

It may also be desirable to limit the amount of short selling on each asset when we are readjusting our portfolio. If such short selling bounds on each asset is $\boldsymbol{s} = (s_1, s_2, ..., s_n)$, then we can express the constraint as $w_i + x_i^+ - x_i^- \geq s_i$, as seen in the example.

Finally, we might also want to reduce the probability of bad or disasterous outcomes after the investment period. If we suppose that the returns after the period is a gaussian $a \sim (\bar{a}, \Sigma)$ and we want that the total wealth after the investment period to be higher than some W^{low} with a probability of more than η. In fact, we may want more than one such constraint. For example: more than 80% probability that the wealth will not be lower than 0.9, more than 97% probability that it will not be lower than 0.7 etc. Define Φ^{-1} as the inverse cumulative distribution function of η which, given a probability shows us the cut-off return. In the paper these shortfall risk constraints are formulated as $\Phi^{-1}(\eta_j)||\Sigma^{1/2}(w + x^+ - x^-)|| \leq \bar{a}^t(w + x^+ - x^-) - W_j^{low}, j = 1, 2$.

In this example, the authors are solving the optimisation problem of maximising the return after a readjustment of the portfolio. The constraints are to tackle transaction cost, bound the short selling and reduce probability of undesirable outcomes after the investment period. It has been shown that this particular example is a convex optimisation problem. The shortfall risk constraints put two bounds on the efficient frontier. In fact, only one of the shortfall constraints is the effective one and the optimal solution to this problem is shown in the circle in Figure 3 of the paper.

In order to implement this, I will have to set $\boldsymbol{w}$ to some weights, as done in the paper (which uses $1/N$ strategy). Then set $\boldsymbol{s}$ to some short selling bounds and $\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-$, e.g. 1%. Similar for some (η_j, W_j^{low}) combinations. $\bar{a}$ would be found from historical data as done already and it will produce Σ. We can use the CVX toolbox, since it is a convex optimisation problem.

References

[1] H. Markowitz *Portfolio Selection.* The Journal of Finance, vol. 7, no. 1, pp. 77-91, 1952.

[2] P. Brandimarte *Numerical Methods in Finance and Economics* Wiley, 2006.

[3] V. DeMiguel, L. Garlappi, and R. Uppal *Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy?* The Review of Financial Studies, vol. 22, no. 5, pp.1915 1953, 2009

[4] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris *Sparse and stable Markowitz portfolios* PNAS, vol. 106, no. 30, pp. 12 267 12 272, 2009.

[5] M. Lobo, M. Fazel, and S. Boyd *Portfolio optimization with linear and fixed transaction cost* Annals of Operations Research, vol. 152, no. 1, pp. 341365, 2007.