COMP6212 Computational Finance Assignment Part I (1 of 3 from MN)

Zheng Cong Koh 28194578

Question 1

Given $\mu = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 0.005 & 0.0 \\ 0.0 & 0.005 \end{bmatrix}$,

The Markowitz efficient frontier can be found in the E-V space by varying the weights, $\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}$, for the two stocks and finding their corresponding expected return and variances. Equation 1 and 2 are the equations used to compute the expected return and variances for a given value of $\boldsymbol{\pi}$.

$$\text{Expected portfolio return, } r_p = \boldsymbol{\pi}^T \mu \tag{1}$$

$$\text{Variance of portfolio return, } Var_p = \boldsymbol{\pi}^T \Sigma \boldsymbol{\pi} \tag{2}$$

π_1	π_2	r_p	Var_p
1.0	0.0	0.1	0.0050
0.9	0.1	0.1	0.0041
0.8	0.2	0.1	0.0034
0.7	0.3	0.1	0.0029
0.6	0.4	0.1	0.0026
0.5	0.5	0.1	0.0025
0.4	0.6	0.1	0.0026
0.3	0.7	0.1	0.0029
0.2	0.8	0.1	0.0034
0.1	0.9	0.1	0.0041
0.0	1.0	0.1	0.0050

Table 1: Calculated results for plotting the efficient frontier.

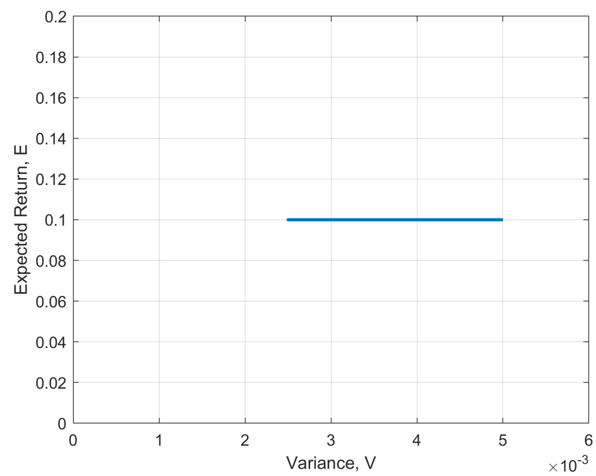


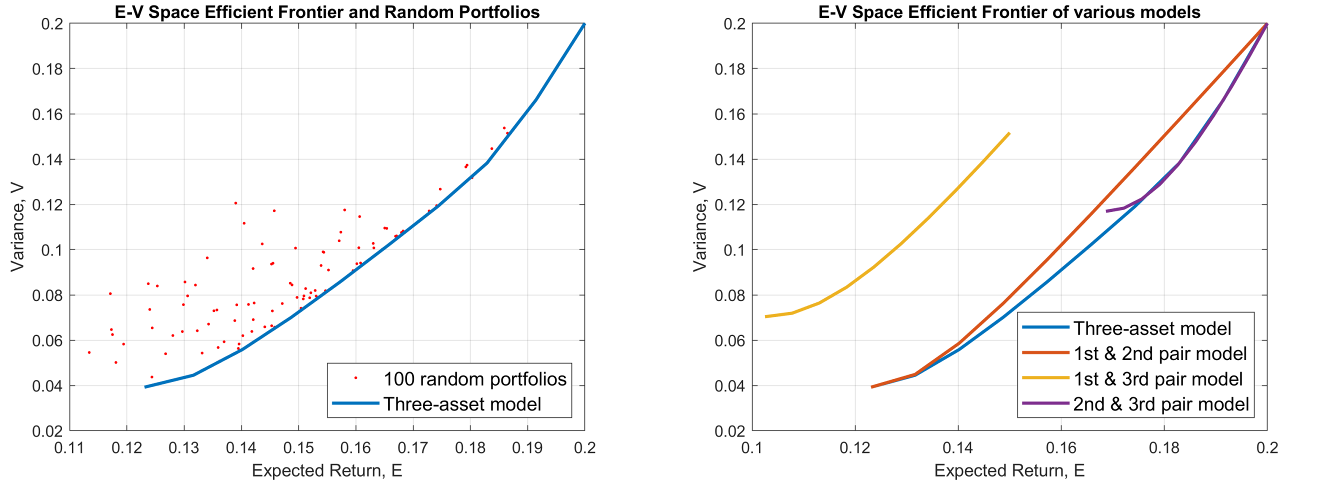
Figure 1: Efficient frontier plot for question 1.

As both components in the given mean are the same values and the variance for each stock is the same, therefore the derived efficient frontier is a straight line where the expected return is 0.1. The variance varies from 0.0025 to 0.005 with the smallest variance (0.005) for when the stocks are evenly split ($\pi_1 = \pi_2 = 0.5$), and the largest variance (0.0025) when only one stock takes up all the weights ($\pi_1 = 1.0,\ \pi_2 = 0.0$ or $\pi_1 = 0.0,\ \pi_2 = 1.0$).

Question 2

Given the expected returns and their corresponding covariances of the three securities, 100 random portfolios can be generated with various combination of their weights, as shown in figure 2a.

From figure 2a, it can be observed that the generated portfolios will never have a higher expected return, E than the efficient frontier, for a given variance, V. This is because the efficient frontier is the set of optimal portfolios that offers the highest E for a defined level of V or the lowest V for a given level of E.



(a) Efficient Frontier for the 3-asset model and 100 randomly generated portfolios.

(b) Efficient Frontier of the three-asset model and three two-asset models.

Figure 2: Generated graphs for the 3 securities.

From figure 2b, it can be observed that by having only the two assets in a portfolio, the portfolio will not perform better than the three-asset portfolio, as the efficient frontier of all the two-asset portfolio are within the efficient frontier of the three-asset portfolio. This is expected as having a more diverse portfolio helps reduce the risk of the portfolio.

Question 3

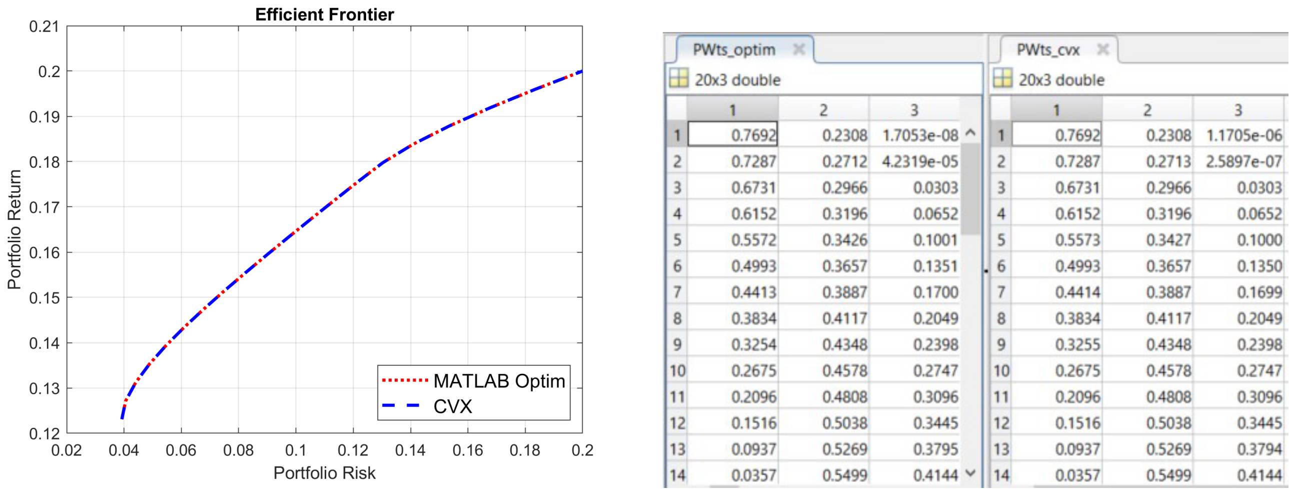
The `NaiveMV` function given in [1] uses functions from the MATLAB optimization toolbox (`linprog`, `quadprog`) to obtain the optimal weights. This can be done similarly using the `CVX` toolbox. Listing 1 provides an example on how the `linprog` function be implemented equally using `CVX`.

Listing 1: Example of changing linprog to CVX.

```
% Find the maximum expected return using linprog
MaxReturnWeights = linprog(-ERet, [], [], V1, 1, V0);

% Find the maximum expected return using CVX
cvx_begin quiet
    variable MaxReturnWeights(3);
    maximise ( MaxReturnWeights' * ERet );
    subject to
    V1 * MaxReturnWeights == 1;
    MaxReturnWeights >= V0;
cvx_end
```

All of the `linprog` and `quadprog` statements in the `NaiveMV` function have been replaced by their equivalent counterparts in CVX and using the same **m** & C from Question 2, the results are compared. From figure 3a, it can be observed that the computed efficient frontier are equivalent for both methods. Figure 3b shows that the weights on the efficient frontier are almost identical from using both methods.



PWts_optim
20x3 double

	1	2	3
1	0.7692	0.2308	1.7053e-08
2	0.7287	0.2712	4.2319e-05
3	0.6731	0.2966	0.0303
4	0.6152	0.3196	0.0652
5	0.5572	0.3426	0.1001
6	0.4993	0.3657	0.1351
7	0.4413	0.3887	0.1700
8	0.3834	0.4117	0.2049
9	0.3254	0.4348	0.2398
10	0.2675	0.4578	0.2747
11	0.2096	0.4808	0.3096
12	0.1516	0.5038	0.3445
13	0.0937	0.5269	0.3795
14	0.0357	0.5499	0.4144

PWts_cvx
20x3 double

	1	2	3
1	0.7692	0.2308	1.1705e-06
2	0.7287	0.2713	2.5897e-07
3	0.6731	0.2966	0.0303
4	0.6152	0.3196	0.0652
5	0.5573	0.3427	0.1000
6	0.4993	0.3657	0.1350
7	0.4414	0.3887	0.1699
8	0.3834	0.4117	0.2049
9	0.3255	0.4348	0.2398
10	0.2675	0.4578	0.2747
11	0.2096	0.4808	0.3096
12	0.1516	0.5038	0.3445
13	0.0937	0.5269	0.3794
14	0.0357	0.5499	0.4144

(a) Computed efficient frontier for both methods. (b) Optimal weights on the efficient frontier for both methods.

Figure 3: Computed outputs from using MATLAB optimisation toolbox and CVX toolbox.

Question 4

The three stocks chosen from FTSE 100 for Question 4 & 5 are MRW, HRGV & GLEN, with the historical data extracted from 15/02/2016 to 13/02/2019. From the first half of the time series, the expected daily return and covariances are computed, where the daily return for a stock is the daily percentage change of the stock's price, which gives $\mu = \begin{bmatrix} 0.000978 \\ 0.000566 \\ 0.003722 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 0.0002 & 0.0001 & 0.0001 \\ 0.0001 & 0.0004 & 0.0002 \\ 0.0001 & 0.0002 & 0.0010 \end{bmatrix}$.

Using the estimated μ and Σ, 1000 random portfolios are generated and plotted on the E-V space to have a better understanding of the three chosen stocks. The Sharpe ratio for each of the portfolio are also calculated, using equation 3. The results are then plotted in figure 4.

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \tag{3}$$

where:

R_p = Return of portfolio
R_f = Risk-free rate (Assumed to be 0)
σ_p = Standard deviation of portfolio

The implemented portfolio strategy is the minimum-variance strategy, which impose restrictions on the estimation of the moments of the asset return [2]. The chosen portfolio minimises the variance of returns, i.e.:

$$\min \quad \boldsymbol{\pi}^T \Sigma \boldsymbol{\pi}, \text{ s.t. } \mathbf{1}^T \boldsymbol{\pi} = 1 \tag{4}$$

This gives the portfolio weights, $\boldsymbol{\pi} = \begin{bmatrix} 0.69 \\ 0.26 \\ 0.05 \end{bmatrix}$. Using the computed weights, the performance of the portfolio can now be compared with that of using the naive portfolio.

To simulate the performance of the portfolios on the second half of the historical data (15/08/17 - 13/02/19), we assume the portfolio size is £10,000 in the first day, illustrated at start of the time series in figure 5. Therefore, the £10,000 is split into the three stocks (MRW, HRGV, GLEN) accordingly, based on the portfolio's weights.

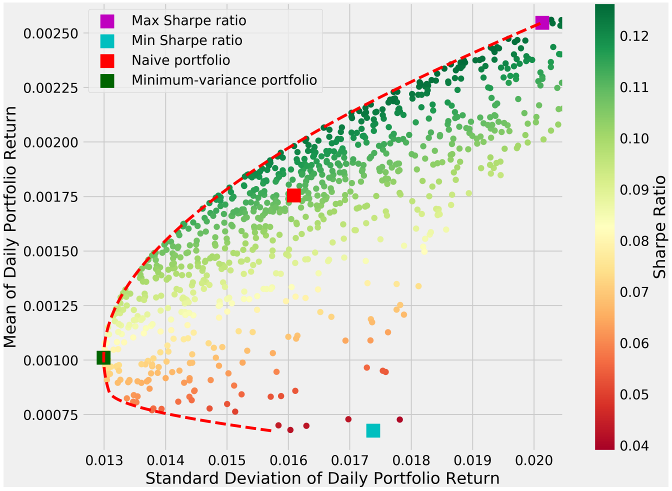


Figure 4: Portfolios on E-V space using estimated μ & Σ.

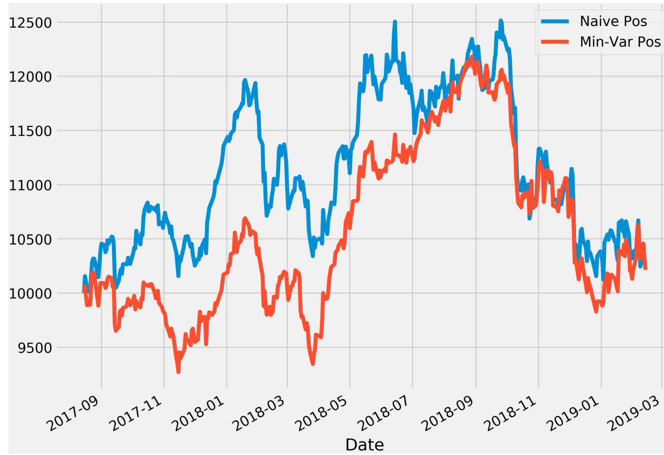
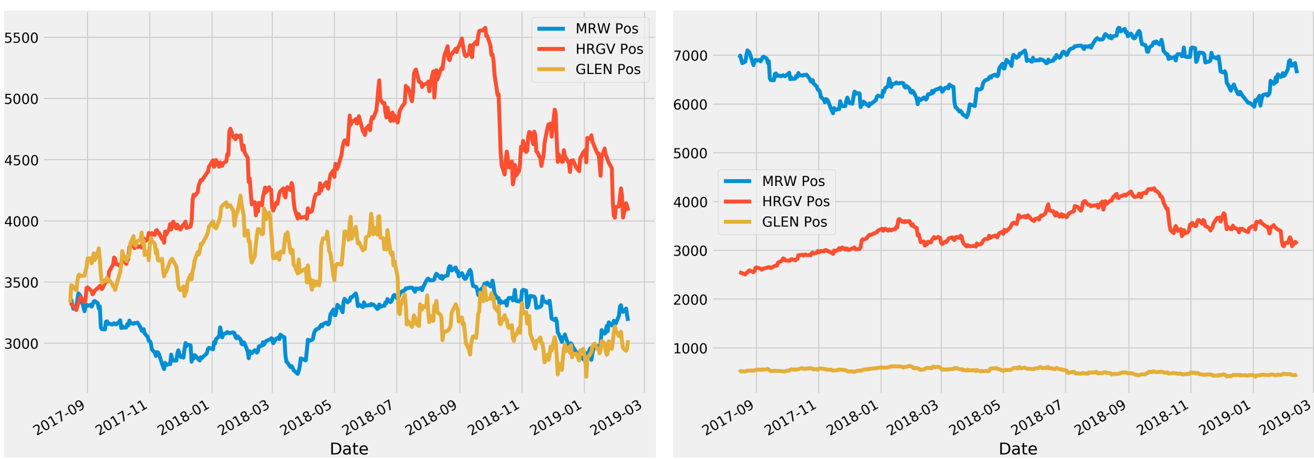


Figure 5: Total position for both portfolios.

For both the naive portfolio and the minimum-variance portfolio, the daily position for each stock is computed, as shown in figure 6a & 6b. The computed Sharpe ratio for the naive portfolio and the min-variance portfolio are 0.01235 and 0.01062 respectively.



(a) Individual stock position for naive portfolio.

(b) Individual stock position for min-variance portfolio.

Figure 6: Performance analysis for naive & min-variance portfolio.

It can be observed from figure 6a & 6b that there is a larger variance on the position for the naive position as compared to that of the min-variance portfolio. This aligns with figure 4 where the min-variance portfolio has the smallest standard deviation.

However, as the min-variance portfolio has the smallest standard deviation, it also have a lower expected return as compared to the naive portfolio. From figure 5, it can be observed that for the majority of the time series the naive portfolio outperforms the min-variance portfolio.

Looking at the Sharpe ratio for both portfolios, it can be observed that the naive portfolio's Sharpe ratio is greater than that of the min-variance portfolio. These findings aligns with the results shown in [2], such that strategies that constrain shortsales perform much worse than the naive strategy for small estimation windows.

Question 5

The method implemented for question 5 is the shortsale-constrained portfolio. The shortsale-constrained portfolio strategy is implemented by imposing an additional non-negativity constraint on the portfolio weights in the optimisation problem to find the optimal weights. As described in the existing literature in [2], imposing such constraints is equivalent to "shrinking" the expected return towards the average. This may be beneficial as it is difficult to estimate the expected return accurately.

Therefore, for the minimum-variance portfolio implemented in question 4, we add the additional constraint $\pi_i \geq 0,\ i = 1, ..., N$ for N weights in the optimisation problem. Implemented in [2] as "g-min-c" is a strategy which is a combination of the $1/N$ policy and the constrained-minimum-variance strategy, in which the problem is formulated as:

$$\begin{aligned} \min \quad & \boldsymbol{\pi}^T \Sigma \boldsymbol{\pi} \\ \text{subject to} \quad & \sum_{n=1}^{\infty} \pi_i = 1 \qquad \text{and } \pi \geq a \\ & a \in [0, 1/N] \end{aligned} \tag{5}$$

To compare the performance against the minimum-variance portfolio and the $1/N$ portfolio, a is arbitrarily chosen to be $a = \frac{1}{2}\frac{1}{N}$, which act as the middle ground between the constrained-minimum variance portfolio and the $1/N$ portfolio. This gives the portfolio weights, $\boldsymbol{\pi} = \begin{bmatrix} 0.64 \\ 0.19 \\ 0.17 \end{bmatrix}$.

As a is set to $\frac{1}{2}\frac{1}{N}$, it forces all the weights to be at least $\frac{1}{2}\frac{1}{N} = 0.16$. Comparing the portfolio weights computed for "g-min-c" against the portfolio weights computed in question 4, it can be seen that the weights given to the third stock (GLEN) is greater, with only minor differences for the other 2 weights.

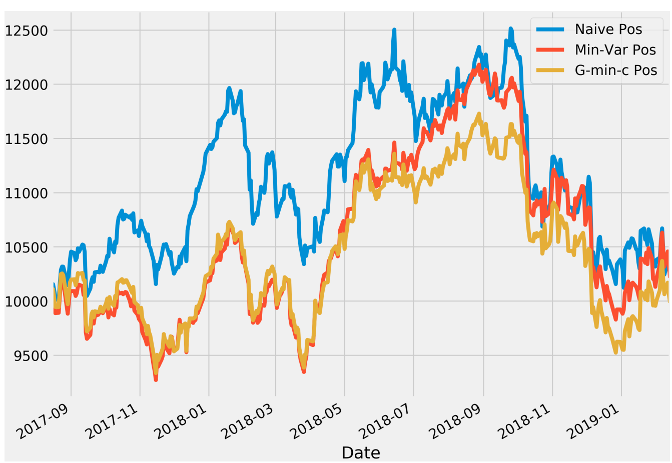


Figure 7: Comparing the 3 portfolios.

Using the portfolio weights on the 3 chosen stocks, it can be observed from figure 7 that it performs similarly to the original min-variance portfolio. This is because there is only a small difference between the two portfolio weights.

Question 6

To implement both the greedy forward selection algorithm and the lasso regression algorithm for index tracking, the normalised rate of return against the first day of the historical data is used instead of their actual values, as the prices of the individual stock and the index value are different. Another reason that the normalised change is used is because the index tracking problem only requires the trend of the index to be tracked, instead of the exact values. This is illustrated in figure 8 and figure 9, where the normalised rate of return provides the trend information regardless of how large the values are.

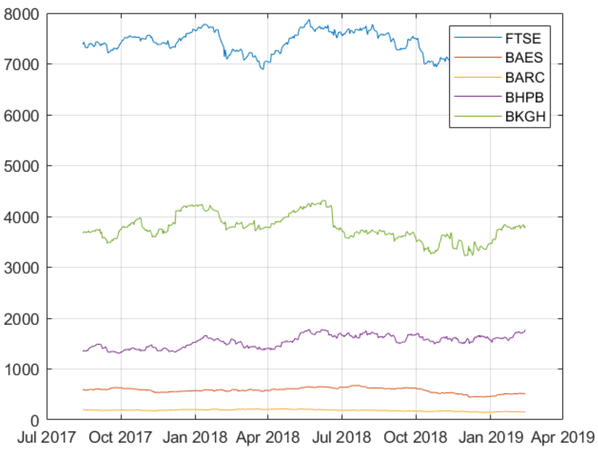


Figure 8: Exact values from historical data.

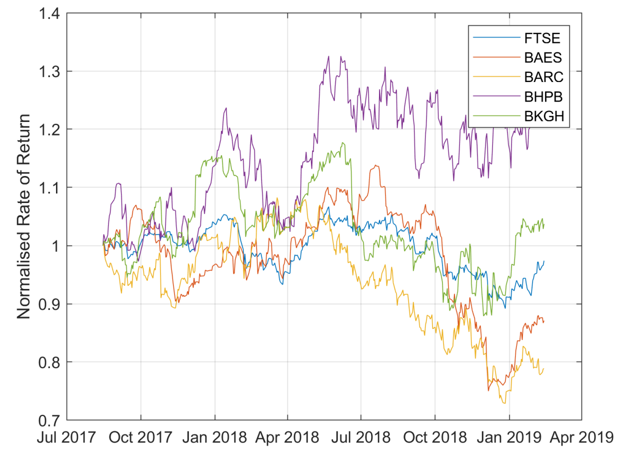


Figure 9: Normalised rate of return from historical data.

To find and select the stocks with their weights for tracking the FTSE 100 Index, only the first half of the historical data is used for both methods, and the second half of the historical data is used to compare the out-of-sample performance.

Greedy Forward Selection

The greedy forward selection algorithm is a feature selection algorithm that chooses the best feature at each iteration. In the context of index tracking, this can be used to choose a subset of stocks that approximates the performance of the index. The following describes the steps taken to implement greedy forward selection for the index tracking problem:

1. Given the 30 stock's normalised rate of return, choose 1 stock which have the smallest difference with the FTSE 100 Index. Save that stock as the current best model.
2. Using the current best model, find the next stock that when combined with the current best model with their optimal weights, gives the lowest sum squared error (SSE). Update the current best model by adding that stock to the model.
3. Repeat step 2 until 6 stocks (a fifth of 30 stocks) are chosen, which best represent the index.

For step 1, the average relative difference between a stock's rate of return (S) and the index's rate of return ($FTSE$) for time series T can be computed using equation 6. This is used to find the first stock that best tracks the index.

$$\text{Average Relative Difference} = \frac{\sum_t^T \sqrt{(S_t - FTSE_t)^2}}{\sum_t^T FTSE_t} \tag{6}$$

For each combination of stocks in step 2, their optimal weights ($\boldsymbol{\pi}$) are computed using `CVX`, which the objective function is given in equation 7, where X is the matrix containing the historical data of the stocks, and Y is the historical data of the index.

$$\begin{aligned} &\text{minimise} \quad && \|X\boldsymbol{\pi} - Y\|_2^2 \\ &\text{subject to} \quad && \sum_{n=1}^{\infty} \pi_i = 1 \quad \text{and } \boldsymbol{\pi} \geq 0 \end{aligned} \tag{7}$$

The obtained results from this algorithm is shown in table 2.

Number	1	2	3	4	5	6
Stocks	SDR	GSK	BP	MRW	HRGV	PRU
Weights	0.199	0.374	0.140	0.105	0.104	0.078

Table 2: Obtained results from greedy forward selection.

Lasso Regression

The sparse index tracking portfolio using l_1 regularization as described in [3] augments the objective function as shown in equation 7 which gives equation 8. This is implemented in MATLAB using the `lasso` function from the Statistics and Machine Learning Toolbox.

$$\boldsymbol{\pi}^{[\tau]} = \arg\min\{\|X\boldsymbol{\pi} - Y\|_2^2 + \tau\|\boldsymbol{\pi}\|_1\} \tag{8}$$

Adding the l_1 penalty helps to promote sparsity, which shrinks the less important feature's weights to zero, thus removing some feature altogether. This works well for feature selection in the index tracking problem. The parameter τ is tuned so that the number of stocks chosen is equal to the number of stocks used in the greedy forward selection. Figure 10 illustrates how the number of stocks varies with parameter τ.

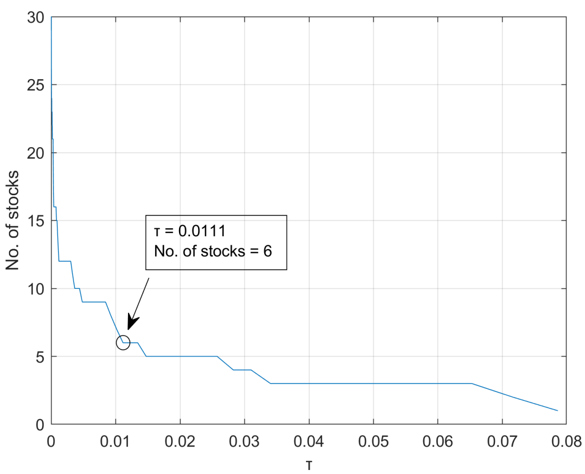


Figure 10: No. of stocks against τ.

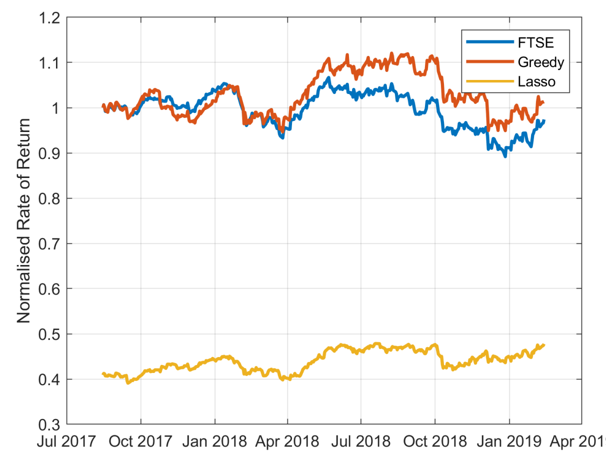


Figure 11: Out-of-sample performance for both methods.

Using the results obtained from figure 10, the value for $\tau = 0.0111$ is chosen and used to find the 6 best stocks. The obtained results from lasso regression is shown in table 3.

Number	1	2	3	4	5	6
Stocks	BP	EXPN	GLEN	HSBA	IHG	REL
Weights	0.012	0.180	0.012	0.048	0.075	0.082

Table 3: Obtained results from lasso regression, with $\tau = 0.0111$.

Comparing greedy forward selection and lasso regression

Greedy forward selection has been known to be a sub-optimal algorithm because the model is built on previous features that it chose. For example, having chosen the first feature, it is fixed on using the first feature in combination with another feature which might not be the combination of two features that produces the least error. The lasso regression considers all the stocks together and finds the stocks that are more general and important.

Looking at the subset of stocks selected by the two methods, it can be observed that no single stock are present in both of the methods. It is important to note that the weights for the result from lasso regression are not summed up to 1. This is why the rate of return for the lasso regression shown in figure 11 is not at the same level as compared to greedy model and the FTSE index itself. Regardless, it is still able to track the trend of the index.

Question 7

Given the portfolio selection problem as described in [4], where the objective function is to maximize the return on the asset, subjected to a budget constraint and a set of feasible portfolios, the problem can be expanded to take into account of transaction costs and additional constraints, as described below and in the example problem of Section 1.6 [4]. This is possible because adding convex transaction costs and convex constraints to an originally convex problem will still results to a convex problem.

$$\begin{aligned}
&\text{maximize} \quad && \bar{a}^T(w + x^+ - x^-) && \text{(9a)}\\
&\text{subject to} \quad && \mathbf{1}^T(x^+ - x^-) + \sum_{i=1}^{n}(\alpha_i^+ x_i^+ + \alpha_i^- x_i^-) \leq 0 && \text{(9b)}\\
& && x_i^+ \geq 0,\ x_i^- \geq 0,\ i = 1, ..., n && \text{(9c)}\\
& && w_i + x_i^+ - x_i^- \geq s_i,\ i = 1, ..., n && \text{(9d)}\\
& && \Phi^{-1}(\eta_j)\|\Sigma^{1/2}(w + x^+ - x^-)\| \leq \bar{a}^T(w + x^+ - x^-) - W_j^{\text{low}},\ j = 1, 2. && \text{(9e)}
\end{aligned}$$

The original objective function $\bar{a}^T(w + x)$ is updated to objective function 9a along with constraints 9b & 9c, to take into account for a linear transaction cost, and constraints 9d & 9e are updated accordingly where x^+ and x^- are the vector of amounts transacted for buying assets and selling assets respectively. A linear transaction costs is where the cost for each transaction is proportional to the amount traded, which is reflected in constraint 9b by α_i^+ and α_i^-, which are the cost rates associated with buying and selling asset i.

Next, shortselling constraints (constraint 9d) are also added which sets individual bounds s_i on the maximum amount of shortselling allowed on asset i. The 2 shortfall risk constraints (constraint 9e) allows the requirement of having the probability of wealth W exceeding an undesired level W^{low} is greater than η, to be imposed on the convex optimisation problem 9. However, this is only possible provided $\eta \geq 0.5$, so that the constraint remains convex, else if $\eta < 0.5$, the constraint becomes concave in x.

For the optimisation problem 9, the plot for the cumulative distribution of the return as shown in Fig. 2 of [4] is obtained due to the fact that the end of period wealth is a Gaussian random variable.

It can be observed from Fig. 2 of [4] that the expected return is a good return as it is greater than 1.0. Having the active constraint which is where $\eta_2 = 97\%$, $W_2^{\text{low}} = 0.7$, means that the optimal portfolio limits the probability of a return below 0.7 to no more than 3%, this also means that the probability of a return above 0.7 is at least 97%.

Therefore, it can be deduced from the figure that for a higher return (where z increases), the probability that the return (R) is less or equal to z increases, suggesting that it is more likely to happen. Similarly, for a lower return (where z decreases), that probability decreases, suggesting it is less likely to happen.

References

[1] Brandimarte, P., 2013. Numerical methods in finance and economics: a MATLAB-based introduction., pg 33-35, John Wiley & Sons.

[2] 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.

[3] 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.

[4] M. Lobo, M. Fazel, and S. Boyd, "Portfolio optimization with linear and fixed transaction costs," Annals of Operations Research, vol. 152, no. 1, pp. 341–365, 2007.