COMP6212 Coursework 1

ZHIHAN WANG
Data Science
zw3u18@soton.ac.uk

March 1, 2019

1 Markowitz efficient frontier

Efficient frontier shows the relationship between two variables, the expected value of a portfolio and the corresponding variance of this portfolio. Each point on the frontier represents the lowest variance in a given expected value or the highest expected value with a given variance. In the first question, there are 2 assets and the mean and covariance matrix are as below.

$$m = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}, C = \begin{bmatrix} 0.005 & 0.0 \\ 0.0 & 0.005 \end{bmatrix}$$

In order to calculate Markowitz efficient frontier, firstly we need to find out all possible portfolio. i.e. available weights for these two assets. To obtain the optimal weights, we want the expected value of the portfolio the highest and, at the same time, the risk the lowest. To describe this thing in a maths way, the equations are constructed as below.

$$\begin{aligned} &\max_w w^T m \\ &\min_w w^T C w \\ &s.t \;\; w^T \mathbf{1} = 1 \end{aligned} \tag{1}$$

m is the expected mean vector, while C is the covariance matrix representding the risk of each asset and the relationship between any two assets. w is the weight vector and also our target. Different w can produce different assets combination. The lager value of components from the weight vector means more holding on the corresponding asset. However, in order to normalize, the sum of the value of all the components from the weight vector should be equal to 1. In the question 1, according to the equation 1, we can derive a weight vector, w=$[0.5 \;\; 0.5]^T$. As there is only one weight vector available, the frontier would be one point. The standard deviation and the expected value are 0.05 and 0.1 respectively.

2 Markowitz efficient frontier

100 random portfolios can be achieved by randomly generating 100 weight vectors corresponding to 3 assets. In this part, I am going to use two MATLAB internal financial toolbox 'portopt' and 'portstats' to draw the efficient frontier and plot a scatter diagram[2].

2.1 Three-asset model

Firstly, randomly generating 100 weight vectors with 3 components in each vector. Then normalize these vectors to make the sum of 3 components from 100 weight vectors all equal to 1. In the next step, using 'portstats' to plot scatter histogram, in which x axis represents the risk of portfolios and y axis represents the mean of portfolio returns. Finally, using 'portopt' to add the efficient frontier to the scatter plot.

Figure 1 shows the final result. There are 100 portfolios in red points and a blue frontier curve. In the figure, all the red points are scattered on or below the blue efficient frontier. Each point on the frontier corresponds the biggest value of expected return with regard to a given standard deviation value(risk). At the same time, the points on the frontier always correspond the smallest investment risks when the expected returns values are given.

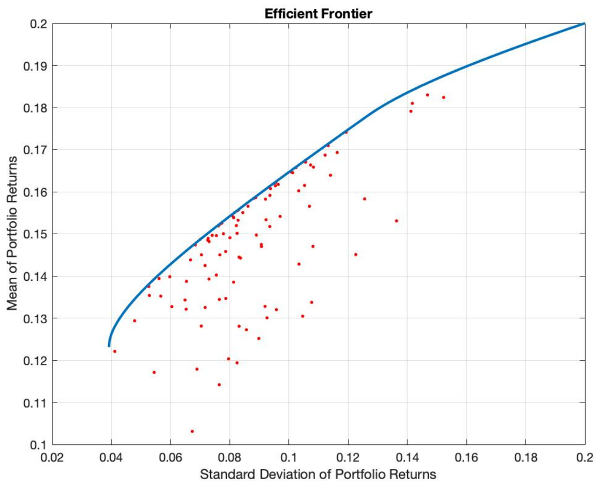

Figure 1: Red points represent 100 random portfolios. In each portfolio, there are 3 assets. Each point represent a portfolio corresponding the investment risk and expected return. The blue curve is the efficient portfolio frontier. It contains all portfolio with the best performance. When the risk is given, the points on the curve are with the highest return. When the return is given, the points on the curve are with the lowest risk. The portfolio points will not exceed the curve all the time.

2.2 Two-asset model

To compare the different performance of the whole-asset portfolio with subset-asset portfolios, draw efficient frontiers for all of them respectively. Figure 2 shows 4 curves, including 3 portfolio frontiers of

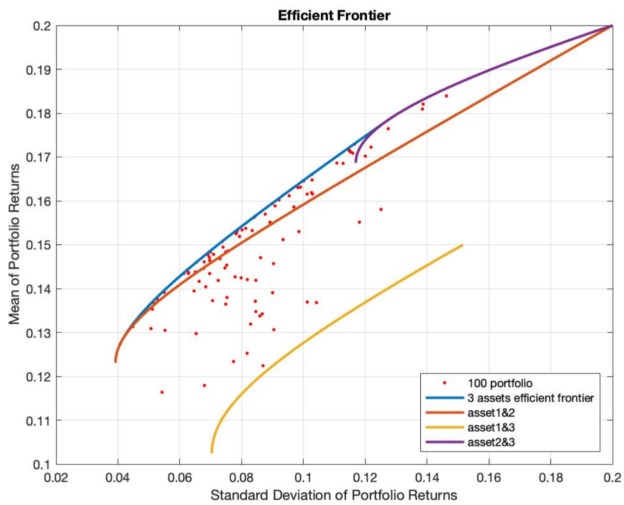

Figure 2: two-asset combinations and three-asset portfolios.

two-asset combinations and one from the whole assets. From this figure, we can see three-asset model performs better than any two-asset models. Although part of the frontier curve of the combination of asset 2&3 overlaps with the whole asset portfolio, it does not exceed the blue frontier. Therefore, all the two-asset combinations perform poorer than the three-asset model.

3 NaiveMV w/o CVX

3.1 NaiveMV without CVX

We are going to find the portfolio with the maximum expected return and the minimum return variance similar to the question 2. But in this question, we are going to write optimization function NaiveMV in MATLAB by ourselves. At first, there are two optimization we need to implement in NaiveMV. They can be converted to maths denotation same as equation 1.

In the NaiveMV function, 'linprog' is used to optimize the highest expected return and 'quadrog' is used to optimize the lowest variance which can also be called risk. They can be explained as equation 2 and equation 3.

$$\min_x f^T x \ such\ that \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ lb \leq x \leq ub \end{cases} \tag{2}$$

$$\min_x x^T Hx + f^T x \ such\ that \begin{cases} Ax \leq 0 \\ A_{eq}x = b_{eq} \\ lb \leq x \leq ub \end{cases} \tag{3}$$

The corresponding programming languages are as below.

```
x=linprog(f,A,b,Aeq,beq,lb,ub)
x=quadrog(H,f,A,b,Aeq,beq,lb,ub,x0)
```

3.2 NaiveMV with CVX

CVX is a Convex Programming toolbox which can be used to replace the optimization steps in cvx style.

To replace 'linprog', the matlab code is as below.

```
    """
    Find the maximum expected return
    """
   cvx_begin
       variable MaxReturnWeights(NAssets);
      MaxReturn = MaxReturnWeights.' * ERet;
      minimize( -MaxReturn );
       subject to
           V0 <= MaxReturnWeights;
           MaxReturnWeights.'*V1.' == 1;
    cvx_end
    MaxReturn = MaxReturnWeights.' * ERet;
```

To replace 'quadrog', the matlab code is as below.

```
    """
    Find the minimum variance return
    """
    cvx_begin
       variable MinVarWeights(NAssets)
       minimize( MinVarWeights.'* ECov * MinVarWeights );
```

```
    subject to
        MinVarWeights.'*V1.' == 1;
        V0 <= MinVarWeights;
cvx_end
MinVarReturn = MinVarWeights.' * ERet;
MinVarStd = sqrt(MinVarWeights.'* ECov * MinVarWeights);
```

Finally, to compare the performance of two programming code, I plot 2 efficient frontiers using 2 code respectively. The data used here are from question 2. Figure 3 shows the comparison, from which we can see that the two methods have the same result.

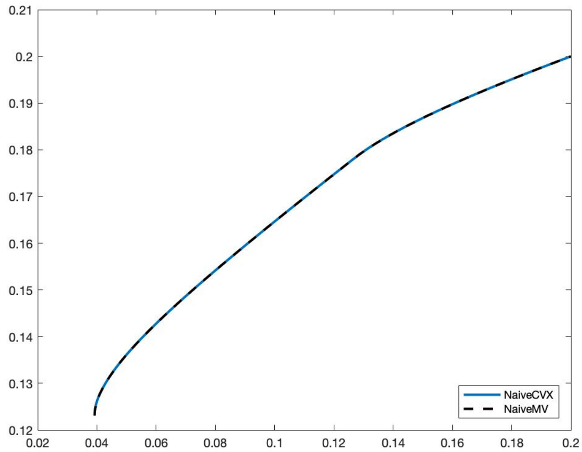

Figure 3: Comparison between NaiveMV function using CVX toolbox and not using CVX toolbox. The two efficient frontier curves totally overlap.

4 3 stocks

In the question 4, I select 30 stocks from FTSE 100 randomly in Yahoo Finance. The data are for the past 3 years. There are some missing values in the three-year period of these 30 stocks. Therefore, firstly, I fill the missing data with the previous recorded data in MATLAB.

To compare the performance of the portfolio using mean-variance method with simple $\frac{1}{N}$ portfolio, I make 10-time choices to ensure the result much more persuadable. For each time, I choose 3 stocks from 30 stocks randomly and implement mean-variance model to find the optimal combination(i.e.weight vector). 'The optimal combination' refers to the combination with the highest sharpe ratio. The Sharpe ratio is calculated by subtracting the risk-free rate (eg.the interest rate on a three-month U.S. Treasury bill for U.S. investors)[2] from the return of the portfolio and dividing that result by the standard deviation of the portfolios excess return. It can be used to evaluate the performance of a portfolio considering the expected return and investment risk[2]. A higher Sharpe ratio can be regarded as a higher return with lower risk.

Then compare the sharpe ratio of our mean-variance optimal combination with the portfolio using simple $\frac{1}{N}$ method. To be more clearly perceived, the portfolio scatter points are connected to two lines. Figure 5 shows the comparison. From this figure, we can see simple $\frac{1}{N}$ portfolio performs better than the portfolio using mean-variance model in much more time.

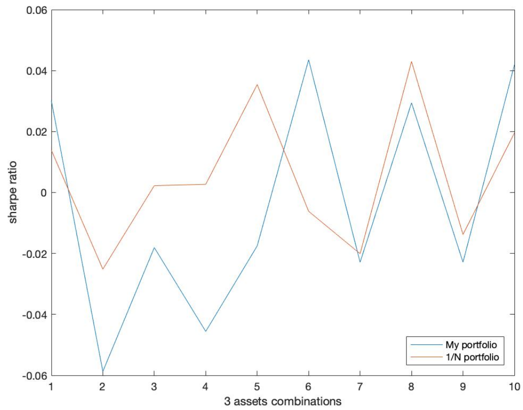

Figure 4: Performance comparison of portfolio using mean-variance model and simple $\frac{1}{N}$ method. Each point in x axis(integer from 1 to 10) is a three-asset combination which is randomly chose from 30 stocks. Y axis refers to the value of 'Sharpe ratio'. For each point in the blue curve, it corresponds to the optimal portfolio of one three-asset combination and the Sharpe ratio of this combination. For each point in the red curve, it corresponds to the simple $\frac{1}{N}$ portfolio of one three-asset combination and the Sharpe ratio value as well. From this figure, more points in the red curve correspond higher Sharpe ratio values than in the blue curve.

5 Enhancement

As the last digit of my student ID is 4, the enhancement method I am going to implement is MacKinlayPastor model(Eqn.7).

The difference between the enhancement method and the traditional method is the way to get estimated mean and covariance matrix. The traditional method derives the mean and covariance matrix directly from the data. This brings a problem. When some factors are not observed, the mean would have a poor explanation to the model and some information(eg. mispricing) is contained in the covariance matrix. MacKinlay and Pastor(2000) found a way to help to alleviate this problem. By constructing an equation $\Sigma = v\mu\mu^T + \sigma^2 I_N$, the covariance matrix Σ can be divided into two parts, the mean part and deviation part, paired with 2 parameters. Then using maximum likelihood method, v, σ^2 and μ can be derived and are put into the optimization problem to get the optimal weight. By using this method, the covariance matrix is well divided. More correlationship information between assets tends to be contained in μ. Therefore, we can make the mean estimator more stable and reliable than the traditional method.

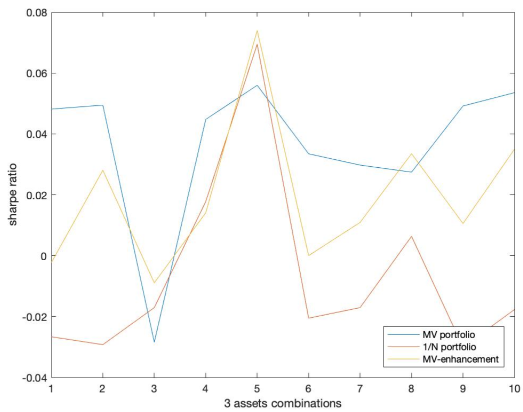

Figure 5: Comparison among 3 models.

To be more clear, firstly, I tried a few values pair, v and σ. The performance of the portfolio with enhancement tends to be better than the simple mean-variance model as a rise of the Sharpe ratio can be seen. The comparison among naive $\frac{1}{N}$ portfolio, simple M-V portfolio from question 4, and portfolio with enhancement are shown in figure 5. From the figure, we can see that the enhancement method always performs better than naive model and better than original mean-variance portfolio in some times.

6 Index tracking

In this part, I am going to compare two strategies for Index Tracking. Both of them are used to solve an optimization problem shown in equation 4, finding a portfolio that has minimal variance for a given expected return ρ.

$$\hat{w} = \arg\min_{w} w^T C w, \ \textit{such that } w^T \mu = \rho, w^T \mathbf{1}_N = 1 \tag{4}$$

For better implementations, this equation can be converted to equation 5.

$$\hat{w} = \arg\min_{w} \frac{1}{T} \|\rho \mathbf{1}_T - Rw\|_2^2, \ \textit{such that } w^T \mu = \rho, w^T \mathbf{1}_N = 1 \tag{5}$$

The 'performance' is evaluated by the value of MSE. MSE stands for mean squared error and can be used to evaluate the distance between the estimated value and the real value. A lower MSE value refers to a better explanation of the model, therefore brings us an accurate prediction and return. But it will introduce some transaction fees in realistic circumstance since we have to invest on a larger number of assets to meet the demand of a low MSE value. A reasonable way to solve this problem in some extent is to introduce a penalty term on the weight of the assets so as to decreasing the quantity of the selected assets.

6.1 Greedy forward selection

Greedy algorithm breaks a big problem into many sub-problems and gives the best result in each step by adding assets to produce the lowest value $\|\mathbf{y} - \mathbf{R}\mathbf{w}\|_2^2$. However, it easily gets struck in the local optima. The hope of finding the global optima cannot be met all the time[1]. In the question 6, firstly we use the greedy forward selection algorithm to find a fifth of 30 stocks.

The result of asset selection using greedy selection is shown in table 1.

Table 1: Asset selection

Run	asset
1	a_6
2	a_6, a_8
3	$\mathrm{a}_6, a_8, a_{22}$
4	$\mathrm{a}_6, a_8, a_{22}, a_{16}$
5	$\mathrm{a}_6, a_8, a_{22}, a_{16}, a_{20}$
6	$\mathrm{a}_6, a_8, a_{22}, a_{16}, a_{20}, a_{15}$

In the sixth run, we find a combination of the assets with the best performance. The corresponding weight is [0.1111 0.1040 0.2200 0.4513 0.0317 0.0819]. The MSE value of this combination is 0.1217.

6.2 L1 regularization

To reduce the transaction cost brought by a larger number of assets, introduce a weight penalty to the original equation 4. Then it would be converted into equation 6.[3]

$$\hat{w} = \arg\min_{w} [\frac{1}{T} \|\rho \mathbf{1}_T - Rw\|_2^2 + \tau \|w\|_1], \ \textit{such that } w^T \mu = \rho, w^T \mathbf{1}_N = 1 \tag{6}$$

$\tau\|w\|_1$ is a penalty term and τ can be regarded as a multiplier to regulate the strength of the penalty. In other words, if τ is big, then the strength of the penalty is big; therefore, there is a smaller number of assets in the investment combination. However, if τ is small, there are more assets in the combination. The sparsity is lower. The relationship between value of multiplier τ and the number of non-zero coefficient is in figure 6. In this part, by tuning τ, finding a combination with 6 assets same as the number of

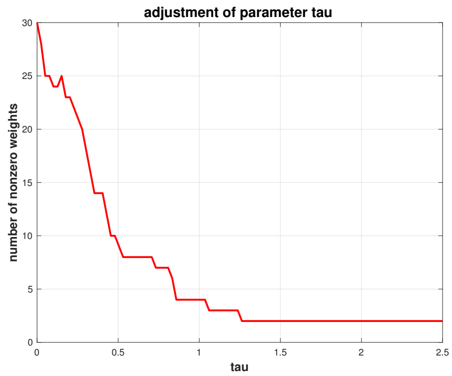

Figure 6: The relationship between τ and sparsity. A higher τ corresponds a smaller number of assets.

assets in greedy algorithm. τ is 0.6 and the asset combination is $[a_6, a_{13}, a_{12}, a_{22}, a_{16}, a_{20}]$, with the weight vector[0.1497 0.0004 0.2080 0.1534 0.2366 0.2483]. The MSE value is 0.2095. The MSE value is a bit higher than that in greedy algorithm, even though both models are with 6 assets. Therefore, in the 6-asset portfolio, greedy algorithm performs better. But if we increase the value of τ to strengthen penalty so as to increase sparsity, we might get a better investment combination using L1 regularization. When τ is set up to 1, the combination only contains 4 assets with the MSE value 0.2353, slightly bigger than 0.2095. This means if we consider the number of assets in the combination significantly, sparse index tracking portfolio using L1 regularization is much better.

7 Lobo paper

Compared with the fundamental portfolio selection problem, Lobo's paper discussed a more complex circumstance, in which the portfolio would be adjusted by performing a number of transactions. However, the primary theory has not changed. The investor's goal is still to maximize the expected wealth at the end of the period, while satisfying a set of constraints[4].

In the fundamental portfolio model, we only need to maximize the expected return with regard to constraining risk to a low value, or minimize the risk with regard to constraining expected return to a high value.

In Lobo's model, the constraints are not as simple as before anymore but still in the same fashion[4]. There are more constraints implemented. An important previous assumption is that all the constraints are convex. Therefore, as the combination of convex functions is still convex, the globally optimal portfolio can be computed rapidely and accurately. The constraints implemented in Lobo's model are shown as below.

Transaction costs constraints This constraint represents the sum of all the transaction cost. A simple maths denotation is as below.

$$\phi(x) = \sum_{i=1}^{n} \phi_i(x_i) \tag{7}$$

x is the dollar amount transacted in each asset and ϕ_i is the transaction cost function for asset i. Instead of making any transaction without considering loss, in a more realistic circumstance, all the costs of transaction would be recorded, including buying and selling.

Diversification constraints This constraint requires the amount invested in each asset i lower than a specific value p_i.

$$w_i + x_i \leq p_i, \ i = 1, ..., n. \tag{8}$$

w_i is the current holding of asset i.

Shortselling constraints This constraint shows the maximum amount of shortselling.

$$w_i + x_i \geq -s_i, \ i = 1, ..., n. \tag{9}$$

Variance constraints This constraint requires the standard deviation of the wealth at the end of the period is less than a value $\sigma_m ax$.

$$(w + x)^T \Sigma (w + x) \leq \sigma_{max}^2 \tag{10}$$

Shortfall risk constraints This constraint shows that the probability of the wealth W at the end of period greater than W^{low} is higher than η.

$$Prob(W \geq W^{low}) \geq \eta \tag{11}$$

Fig.2 is derived from this constraint. It shows the cumulative distribution of the return. In the figure, for each point on the curve, it can be explained as the probability of a return below corresponding z less than the corresponding value in y axis. As the vector of returns is subjected to a jointly Gaussian distribution, the expected return is the same as the median return in the cumulative distribution. When z is in a very small value, the probability of a return lower than it is difficult, so the corresponding value in y axis(probability) is very low. In the contrast, when z value is high, the returns are easily lower than it, so the cumulative probability would be close to 1.

Convert all the constraints referred above to the readily handled equivalent denotation and combine all of them, then we can arrive at the convex portfolio optimization problem in the Section 1.6 example.

References

[1] Mao Chen (2008). A Greedy Algorithm with Forward-Looking Strategy, Greedy Algorithms, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech

[2] P. Brandimarte, Numerical methods in finance and economics, 2nd edition, Wiley, 2006.

[3] Brodie, Joshua, et al. "Sparse and stable Markowitz portfolios." Proceedings of the National Academy of Sciences 106.30 (2009): 12267-12272.

[4] Lobo, Miguel Sousa, Maryam Fazel, and Stephen Boyd. "Portfolio optimization with linear and fixed transaction costs." Annals of Operations Research 152.1 (2007): 341-365.

[5] DeMiguel, Victor, Lorenzo Garlappi, and Raman Uppal. "Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?." The review of Financial studies 22.5 (2007): 1915-1953.