

# COMP6212 Assignment 1

Junjun Shi

March 10, 2017

## 1 Efficient Mean-Variance frontier

### 1.a

Generate 100 random portfolios and plot their expected return and variance.

The whole procedure could be described as:

- Construct 100 vectors  $X$  whose length is 3, and the elements  $x_1, x_2, x_3$  satisfy  $\sum_{i=1}^3 x_i = 1$  and  $x_i \geq 0$ .
- For each vector:
  - Calculate the expected return  $E = \sum_{i=1}^3 X_i \mu_i$ ,
  - Calculate the expected variance  $V = \sum_{i=1}^3 \sum_{j=1}^3 X_i X_j \sigma_{ij}$ .

Plot a scatter of diagram, we can get the picture like figure 1.

### 1.b

Matlab no longer support the function `frontcon` in version R2016b, I used the `Portfolio` object as replacement according to the official documnetation.

From the figure 2 we can find that the `frontier` gives a left bound of portfolios.

The figure 3 shows the three two-asset models, the all exist on a single line, and the frontier is also part of the line.

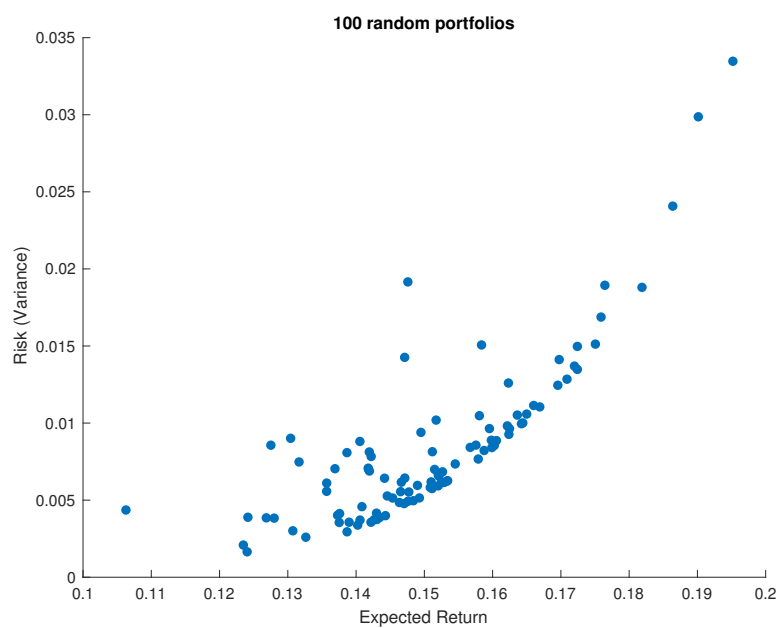


Figure 1: E-V plot of 100 randomly selected portfolios

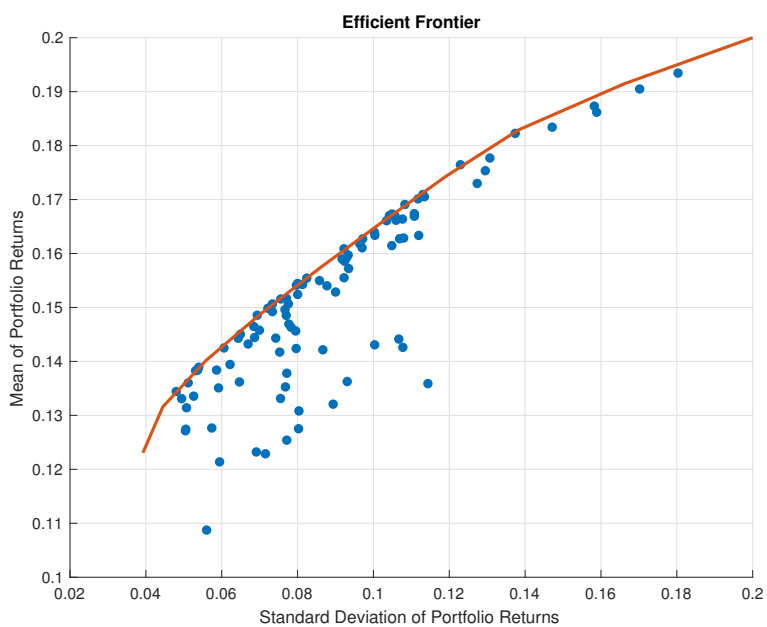


Figure 2: Efficient frontier for three-asset model

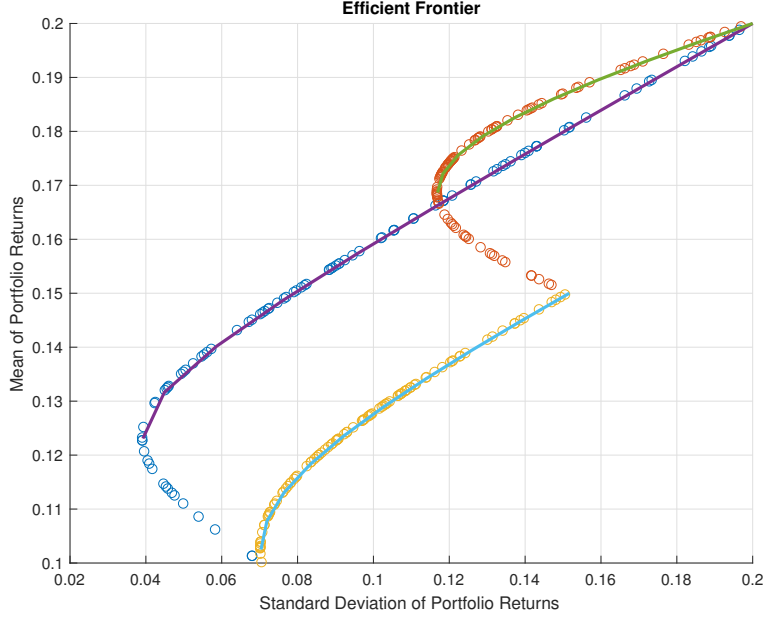


Figure 3: Efficient frontiers for two-asset models

### 1.c

The `NativeMV` function calculate the status of Maximum return and Minimum risk, divide the range of return according to the given number, and calculate the minimum risk of each state. The `linprog` function is used by `NativeMV` to find out the maximum return at first.

The maximum problem:

$$\max \mathbf{w}^T \bar{\mathbf{r}} \text{ subject to } \sum_{i=1}^N w_i = 1, w_i \geq 0$$

be rewritten as:

$$\min -\mathbf{w}^T \bar{\mathbf{r}} \text{ subject to } \sum_{i=1}^N w_i = 1, w_i \geq 0$$

to use the `linprog` function to get the answer.

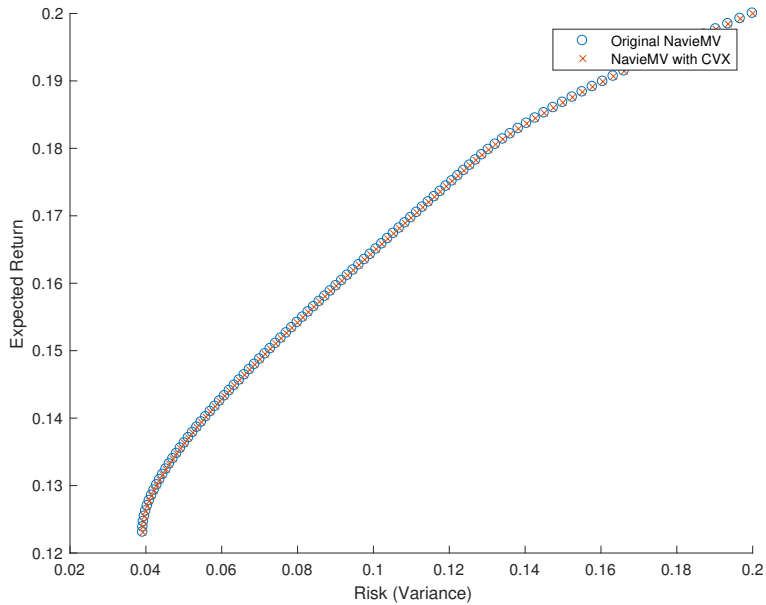


Figure 4: Performance of CVX

### 1.d

The figure 4 shows the result of using original `NativeMV` with `linprog`, `quadprog` and the result of using `CVX` are identical. This is because that the problem they are solving are identical, just using different methods. `CVX` is suitable for solving more generic problems. However, `CVX` cost much more time, which could be easily observed while running code.

## 2 Evaluation of performance

Downloaded the three-year data from **Yahoo! Finance**, using the close price to stand for the price of every single day, the days with no data are. For some situations like no transaction, I used the same price as the previous day standing for that day will have no return if you invest on it. Used MATLAB `tick2ret` function to get the return ratio.

Calculated the mean return of the stocks got, and choosed three positive return stocks to do the next steps.

Based on the data generated and the works done in **Question 1**, generate the frontier of the portfolios. Using the function `estimateFrontier` get a

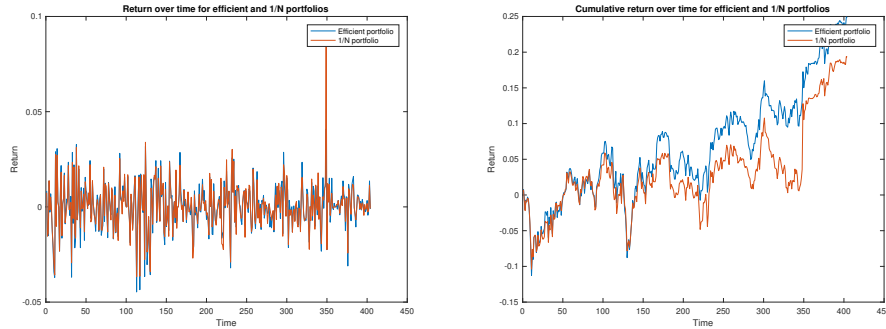


Figure 5: Compare of Mean-Variance and Navie  $\frac{1}{N}$  strategy

set of efficient portfolios and used `sharpe` function to find out the one with the highest **Sharpe Ratio**.

With the portfolio, we can compare it with the simple  $\frac{1}{N}$  portfolio. As shown in the the figure 5, the figure on the left repressing daily return can not show which is better, for the **cumulative** one we can find that the efficient portfolio performs better than the navie one for a long window.

### 3 Greedy and sparse index tracking

#### 3.a

For the greedy strategy,

#### 3.b

### 4 Discussion of Lobo et al.