

1 Efficient Frontier

Let a portfolio of N assets be π , whose expected return is μ and the co-variance is Σ .

1.1 Efficient Frontier with 3 Assets

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

Given $\mu = m$ and $\Sigma = C$ for a 3 assets, we can generate 100 random portfolios, where each portfolio $\pi = (\pi_1 \pi_2 \pi_3)^t$ s.t. $\mathbf{1}^t \pi = 1$ 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)`.

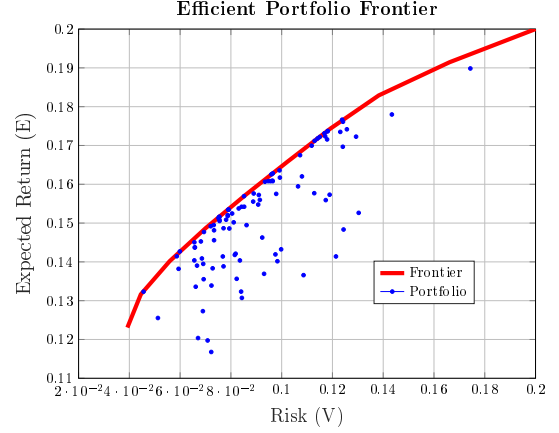


Figure 1: Efficient Portfolio

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 portfolios, we remove 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.

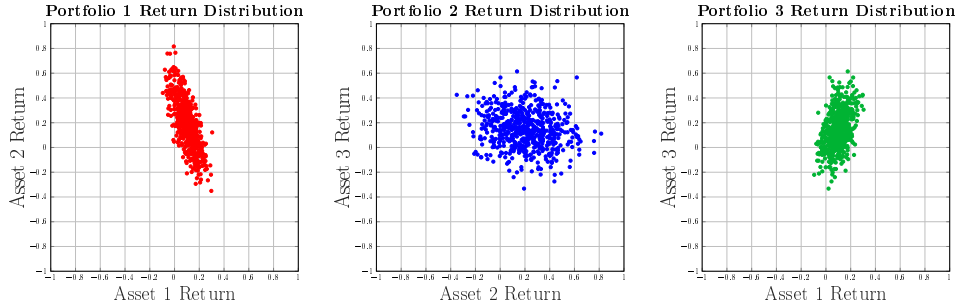


Figure 2: Distribution of 2 Asset Returns

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.

1.3 Use of linprog in NaiveMV

In order to calculate the efficient frontier, we need two extreme points - maximum return for a portfolio regardless of the risk and minimum risk regardless of the return. For the first case, we have $E = \max_w (\pi^t \mu)$

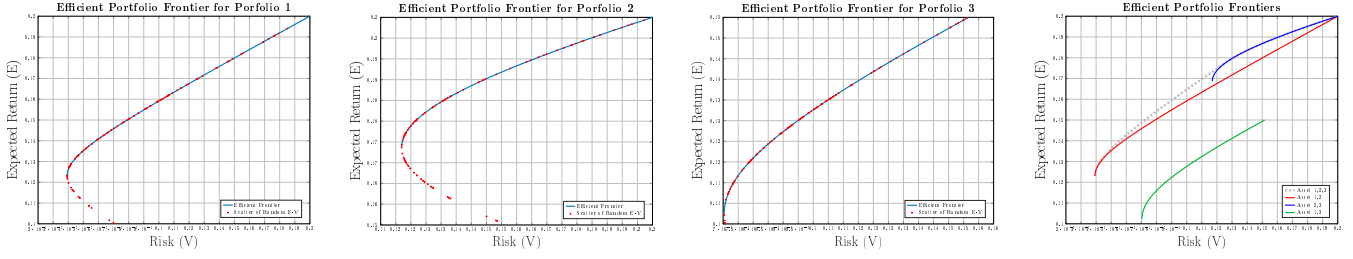


Figure 3: Efficient Frontier for 2 Asset Portfolios

s.t. $\mathbf{1}^t \pi = 1$ 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 π . Matlab's `linprog` function can solve linear equation can solve such equations of the following form -

$$\min_x (f^t x) \text{ s.t. } \begin{cases} A \cdot x \leq b \\ A_{eq} \cdot x = b_{eq} \\ lb \leq x \leq ub \end{cases} \quad f = -ERet; A = []; b = []; Aeq = \text{ones}(1, N); beq = 1; lb = 0; ub = 1$$

However, to calculate the portfolio that minimises the risk we need to solve a quadratic equation of π , $\min_w (\pi^t \Sigma \pi)$ s.t. $\mathbf{1}^t \pi = 1$. In this case we use the `quadprog` function. Finally, we choose N expected returns between the two extreme points and calculate the portfolio 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 declaratively 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
```

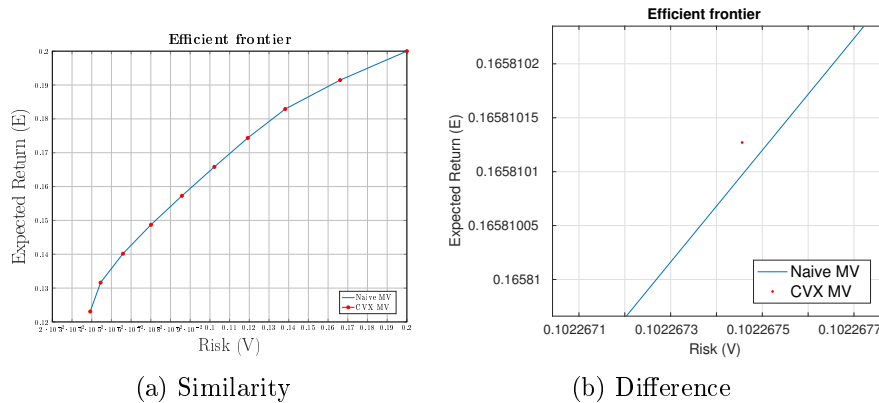


Figure 4: NaiveMV vs Using CVX

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

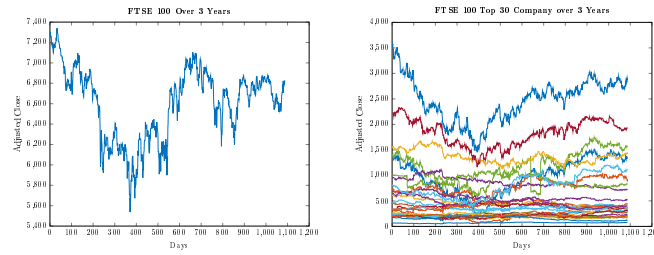


Figure 5: FTSE 100 and Top 30 Most Traded Company Adjusted Close over 3 Years

2.2 Returns and Efficient Portfolio of 3 Random Assets

2.3 Comparison with Naive Strategy