SIAM-Sandra's Notes

Sandra Ng

February 26, 2021

1 Introduction

Solving the risk aversion portfolio optimisation problem with short-selling and leverage constraints.

Maximise the loss function

$$\mathcal{L}_T^\lambda = \mathbb{E}[R_T(w_{0:T-1})] - \lambda\mathbb{V}[R_T(w_{0:T-1})]$$

subject to constraint

$$\boldsymbol{w_t} \in \Delta^d := \{\boldsymbol{x} \in \mathbb{R}^d : x^i > 0 \text{ and } \sum x^i = 1\}$$

Commonly written in terms of one-period returns: Maximise

$$w'\mu - \frac{\gamma}{2}w'\Sigma w$$

subject to

$$w'1_p = 1, \qquad w_i \geq 0$$

where $\gamma = 2\lambda$ is the arrow-pratt meausure of risk aversion and typically takes values between 1 and 10, so we will evalute our strategy over $\lambda \in [0.5, 5]$

This can also be interpreted as maximising the risk-adjusted return.

2 The Market Model

Log-returns are given by

$$\mathbf{S_{t+1}} - \mathbf{S_t} = \boldsymbol{\mu} + \kappa(\Delta\mathbf{w_{t-1}}) + \mathbf{M}\boldsymbol{\xi_t}$$

Market impact function

$$\kappa(x_i) = (c_i sign(x_i)\sqrt{|x_i|})_{i=1}^d$$

Key observations about the market model

- M is a low-rank matrix resulting in highly correlated asset returns
- from the market impact function, $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ hence given two assets with highly correlated returns, transaction costs are lower if we invest in one of them than both of them.

3 cVaR Optimisation

cVaR optimisation is a strategy that seeks to minimize the conditional value at risk, also known as expected shortfall. The cVaR at the $100(1-\beta)\%$ level is the expected loss (magnitude of negative daily return) amongst the worst $(1-\beta)$ outcomes over the trading period. For example, $\beta = 0.95$ implies that we are considering the average of losses that occur in the worst 5% of cases.

Consequently, cVaR optimisation identifies assets with negative drift and takes their weights to 0. This works because in our market model, each asset has a constant drift, μ

We implement *thisalgorithm* for choosing portfolio weights to minimise cVaR, subject to the short-selling and leverage constraints. This reduces the problem into a linear optimisation.

The trade-off between risk and return can be pinned down by the time interval between portfolio rebalancing, henceforth referred to as the 'rebalancing period'. Idea:

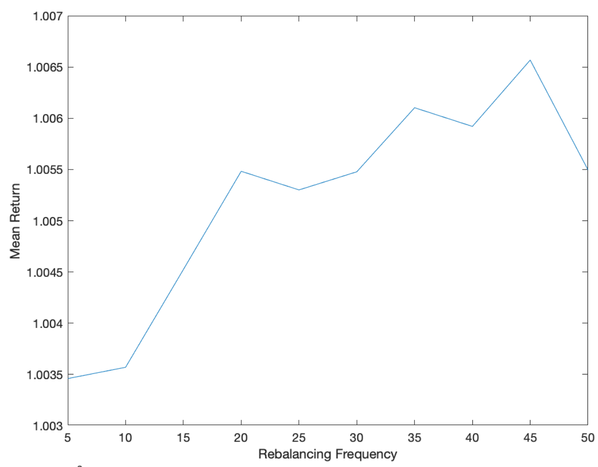
- More frequent rebalancing incurs higher transaction costs, which reduces expected returns
- On the other hand, more frequent rebalancing results in more stable returns as new information is incorporated more quickly, reducing the chances of sudden unexpected loses between rebalancing.

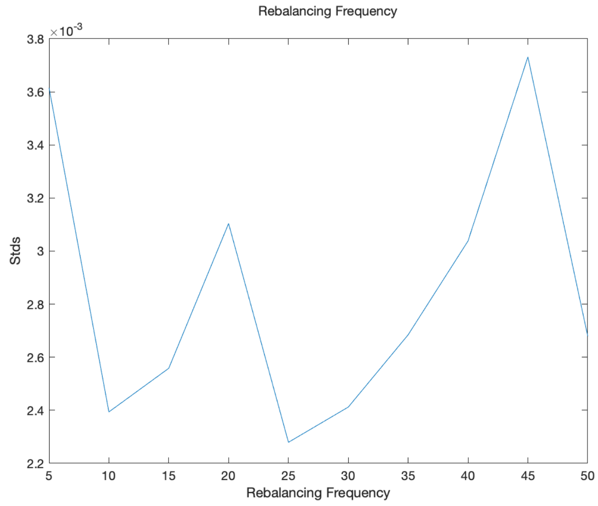
- More risk-averse investors with higher λ are willing to accept lower expected returns in order to have lower returns volatility, which can be achieved by more frequent rebalancing. Hence, we want to set the rebalancing periods as a decreasing function of λ;

$$\text{rebalancing period} = f(\lambda), \qquad f'(\lambda) < 0$$

Before determining the function f, we first verify that the relationship between rebalancing period, mean returns and variance of returns is indeed consistent with the proposed reasoning above.

Using the given parameters $T = 500, d = 50, \eta = 0.0002$, and setting $\beta = 0.99$ and calculating the loss function using 50 simulations for each value of rebalancing period $\in \{5, 10, \dots, 50\}$





Over a suitably chosen range of rebalancing frequencies, the results are indeed as expected.

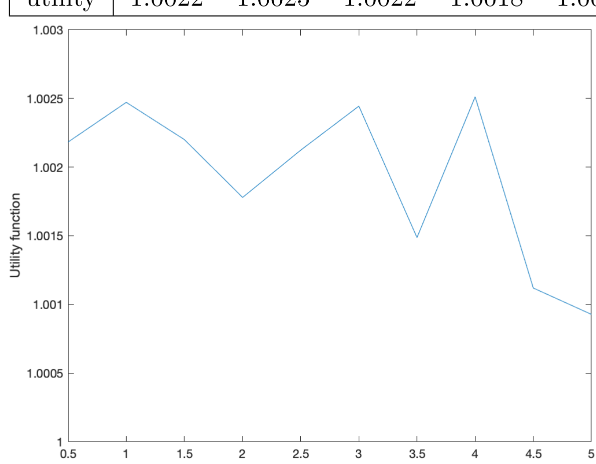
Next, we try the naive strategy of setting f as a decreasing linear function of λ, arbitrarily choosing

$$f(\lambda) = \lfloor 0.2T - (\lambda - 0.5)0.15T/4.5 \rfloor$$

where T is the number of trading days

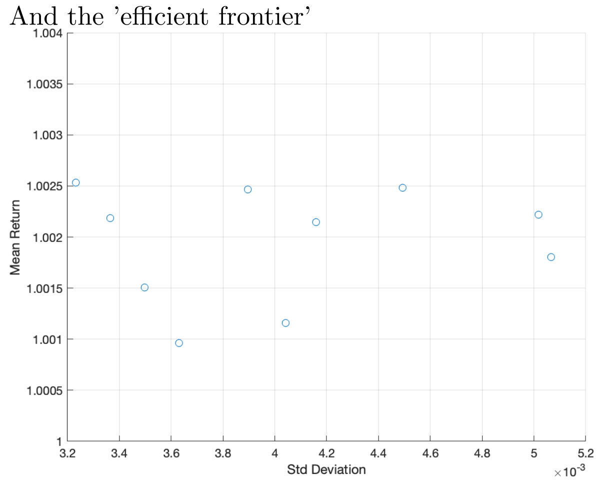
Varying λ over $[0.5, 5]$, we evaluate the utility function

λ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
utility	1.0022	1.0025	1.0022	1.0018	1.0021	1.0024	1.0015	1.0025	1.0011	1.0009



Fairly stable, but observe that our strategy performs better for lower values of λ. Maybe the rebalancing periods were reduced by too much for higher λ, in which case setting f as a decreasing convex function of λ might work better (try another time...)

And the 'efficient frontier'



There is room for improvement but looks promising...ideally we want this to be an increasing function...consistent with our evaluation of the utility function, the results are not so great for lower std, which is probably when rebalancing periods were shorter

Next we should also choose an optimal value of β, the (1-VaR) level

4 Sparse Principal Components

The conventional solution to the mean-variance optimisation problem is to first estimate the covariance matrix Σ using principal component analysis. This reduces the dimensionality of the problem. We then maximise the utility function by quadratic programming

Each principal component (PC) is a linear combination of all assets in the market. However, given the square root market impact function, putting positive weights in every single asset is likely to result in excessive transaction costs.

We use the SpaSM toolbox which calculates the sparse PCs, whereby each PC is now a linear combination of only some of the assets. We can specify the number of sparse PCs, k and the parameter 'stop', which is the number of non-zero weights to include in each PC. The SpaSM toolbox implements the variable selection via LASSO regression.

After estimating Σ, the next step is to maximise the utility function subject to the given constraints. We used quadprog from Matlab's optimisation toolbox to minimize the negative of the utility function.

However, running the quadratic optimisation multiple times is computationally expensive and (in my opinion it is not clear that the function we are minimizing is convex, so quadprog might not actually be returning the global minimum but I have no idea how to justify this). The results were very bad, even for different choices of the parameters k and stop

5 Next Steps

Possible further strands of development (discussed with Pearl)

- cVaR optimisation: further optimise the function f (choosing rebalancing frequency), test how the strategy performs over different specifications of the market model
- Sparse PCA optimisation: find an alternative to quadratic programming
- Hierarchical risk parity: how to incorporate λ (since the lambda-free version of the strategy actually performed pretty well)

Another thought:

- In terms of Sharpe Ratio, cVaR optimisation was clearly superior to momentum strategies due to the inherent volatility of momentum strategies.
- However, we are being evaluated on risk-adjusted returns, not Sharpe Ratio. (I think) the momentum strategies had higher expected returns. Would be interesting to see how the utility function looks like.