Algorithmic Trading: Momentum

Daniel Sinkin
danSinkin@protonmail.com
https://github.com/OneAndOnlyDaniel

March 19, 2023

Contents

1 Introduction

Investment portfolios are an important tool for growing and managing one's financial assets. However, with so many investment strategies and options available, it can be challenging to know where to start. One such strategy is a momentum-based approach, which focuses on investing in assets that have shown strong recent performance. In this article, we will outline a program for building a portfolio using a momentum strategy. Specifically, we will use the quality momentum, which combines four different momenta into a single quantity based on user-defined weights. By implementing this program, individuals can begin to build a diversified portfolio with a focus on recent growth. This approach can be a valuable tool for those looking to begin their investment journey and showcase their understanding of both investment principles and software development skills.

2 Data

2.1 WebScraping: S%P500 Tickers

To get the tickers for the S%P500 we use the `Request` package to obtain the HTML text of the wikipedia page listing the S%P500 and then extracting them from the table.

2.2 API: Yahoo Finance

To obtain relevant stock information we use the `yFinance` package to query Yahoo finance.

3 The Investment Strategy

The basic idea of momentum based investing is that we look at the past behaviour of a stock, and focus on stocks which have a positive and high **quality growth** trajectory, assuming that if we can choose the right type of **quality growth** then we can benefit from the historical behaviour of the stocks.

For the purpose of this article we will be using a linear approach, given changes in value $\mu_i, 1 \leq i \leq n$ we define the **high quality momentum** by

$$\mu := \frac{\sum_{i=1}^{n} w_i \mu_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

where $w_i \geq 0, 1 \leq i \leq n$ such that for at least one $w_j \neq 0$. We call the w_j weights and the number

$$\Omega := \sum_{i=1}^{n} w_i$$

the total weight and call weights **normalized** if $\Omega = 1$. In that case 1 reduces to

$$\mu = \sum_{i=1}^{n} w_i \mu_i$$

which can be efficiently calculated as the dot product

$$\mu := \underline{w} \cdot \underline{\mu}$$

where $\underline{w} = (w_1, w_2, \ldots, w_n), \underline{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$.

This procedure gives us a number $\mu^{(i)}, 1 \leq i \leq N$ for every stock, summing over them yields

$$M := \sum_{i=1}^{N} \mu^{(i)}$$

and the percentage amount a stock would make up of the whole portfolio is

$$\frac{\mu^{(i)}}{M},$$

if the current costs of the stocks are $c_i, 1 \leq i \neq N$ then we'd buy

$$n_i := \operatorname{floor}\left(\frac{\mu^{(i)}}{M} \cdot \frac{C}{c_i}\right)$$

where C is our capital to be invested.

The total amount invested is

$$\Sigma := \sum_{i=1}^{N} c_i \cdot n_i$$

and the amount remaining is the error term

$$\mathcal{E} := C - \Sigma.$$

4 Backtesting

To evaluate the value of this strategy we have to compare the actual results one would have obtained if the strategy had been run in the past. Suppose that the amount of stocks to be bought at the previous time are denoted by n_i and their cost at the time buying is $c_i', 1 \leq i \leq N$. The current costs are denoted by c_i and so the performance of the portfolio is

$$\Delta := \Sigma - \Sigma'$$

where $\Sigma' = \sum_{i=1}^{N} c_i' n_i$ is the money originally invested.