Momentum Trading Algorithm

Thomas Cotter, Dom Sauta, Rachel Nguyen
QFin UWA

A Project Completed by the Trading Team in conjunction with
The University of Western Australia in 2021
Quantitative Finance UWA

Chapter 1

Momentum Trading and Signals

1.1 Momentum Trading

Momentum trading is a technical trading strategy where you analyse assets in the short-term, buy assets that have been showing up-trends, and close the position when the trend starts to lose momentum. In our strategy, Simple Moving Average and Exponential Moving Average of price in a given period are used as technical indicators to provide trading signals.

1.2 Simple Moving Average

To calculate the short-term and long-term trends of the data, we could calculate a simple moving average, which, in our case, is simply the mean of the previous 8 and 24 prices respectively. By comparing the long-term and short-term averages, we can infer general trends in the data. For example, a short-term average greater than a long-term average indicates upward movement of the coin, and vice versa.

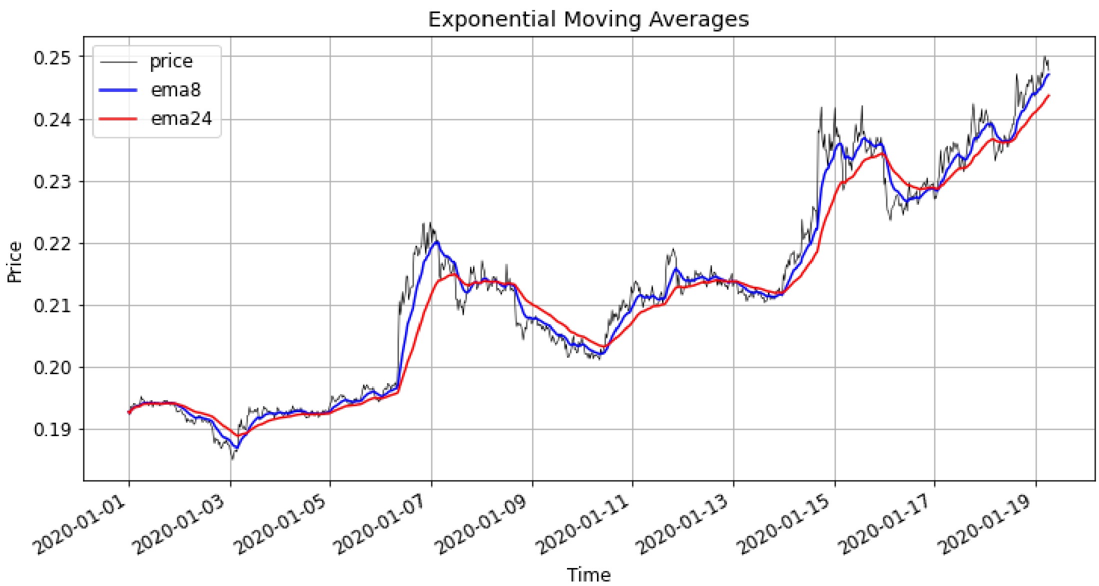
1.3 Exponential Moving Average

The primary drawback of using a simple moving average is that new information is considered to be equally as important as older information when generating a trading signal. In reality, this is not the case, and it is likely, by the efficient market hypothesis, that the most recent price (excluding any noise) contains the most up to date information and should therefore be considered the most important. To capture this idea, we employ an exponential moving average as defined below:

$$\text{EMA}_t(P, \alpha) = \begin{cases} P_0, & t = 0 \\ t\alpha \cdot P_t + (1 - \alpha) \cdot \text{EMA}_{t-1}(P, \alpha), & t > 0 \end{cases}$$

In our case, we specified α in terms of our centre of mass $1/(1 + \text{com})$, where *com* was defined by the number of days we were interested in looking at.

Below, we highlight the differences in the EMAs when changing the COM hyper-parameter. Notice the differences in reactivity to movements in price.



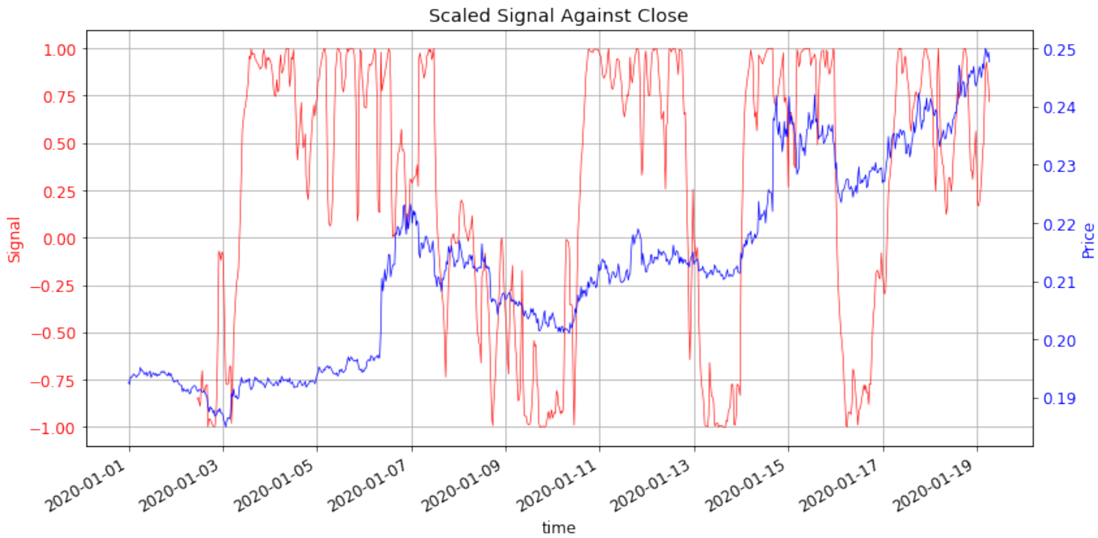
Now that we have the baseline for our signal, the signal is then normalised and created as follows:

```
ema_l = lookback['close'].ewm(com = n_l, adjust = False).mean()[today]
ema_s = lookback['close'].ewm(com = n_s, adjust = False).mean()[today]
sigma_s = lookback['close'].rolling(window = sigma_s_lookback).std()[today]

y = (ema_s - ema_l) / sigma_s
signal_values.append(y)
z = y / np.std(signal_values)

# final signal to be used
u = (z * np.exp(-z**2  /4)) / (np.sqrt(2) * np.exp(-1/2))
```

Notice that was actually transform the signal twice- the first time with a short term volatility measure of the underlying, and the second time with a longer term volatility measure of the signal itself. Due to this we actually lose a lot of trading signals in the warm-up period while we wait for the data to accumulate. This can be seen in the plot below. After all of our transformations, we can make use of the signal detailed below. It is standardised to be between -1 and 1 as per [1], where positive values indicate a long position and negative numbers indicate a short position, and the magnitude represents the relative strength of that signal. Our algorithm however, only assumes long positions.



Now, it is important to construct a threshold for which we buy and sell. This should be subject to optimisation, but we chose to bin into 3 equally sized responses- "long", "short" and "neutral". The signal outputs a long position when the signal is greater than 0.33, and short position when the signal is less than 0.33. Let us define this threshold as η Note that in order to make the algorithm more aggressive, one would simply need to widen the window for buying and selling.

1.4 Testing the Algorithm

Let us examine the results of the algorithm on Cardano, and start tweaking some of the hyper-parameters. Firstly, with default settings as below. Note that these values all have units of '30 minute periods', except η, has no unit as it represents the signal.

n_s	n_l	$\hat{\sigma}(P)$	$\hat{\sigma}(y_k)$	η
8	24	12	168	0.33

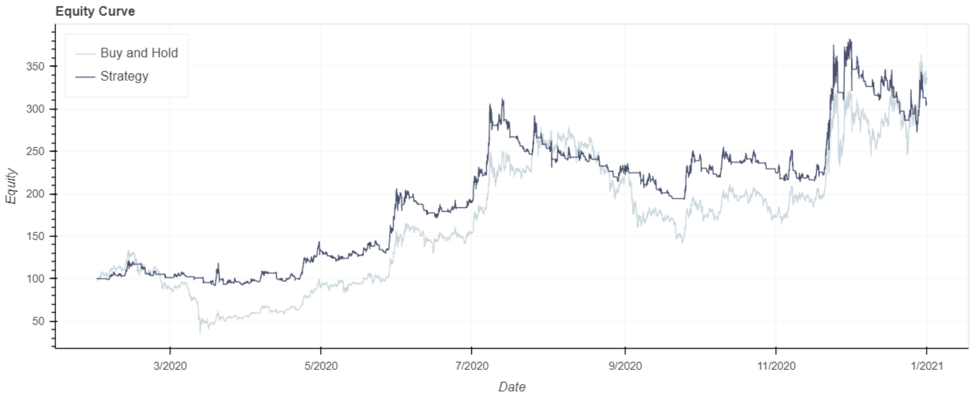


Figure 1.1: Cardano strategy performance with default tuning variable

Let's now demonstrate a more aggressive algorithm with a η of 0.1 for demonstration purposes. Notice that this strategy performs a lot worse- likely a result of more false trading signals. We also trade 326 times- much more than our previous strategy.

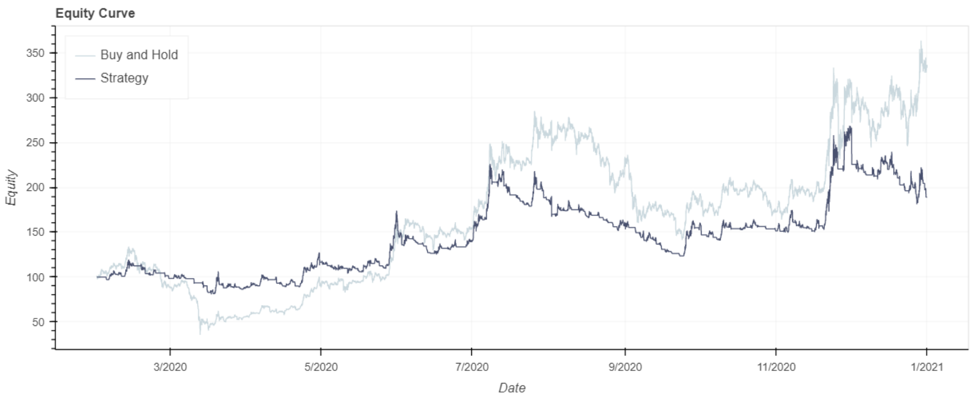


Figure 1.2: Cardano strategy performance with aggressive η

Changing the η value once more to 0.5, we can observe that the algorithm makes 220 trades.

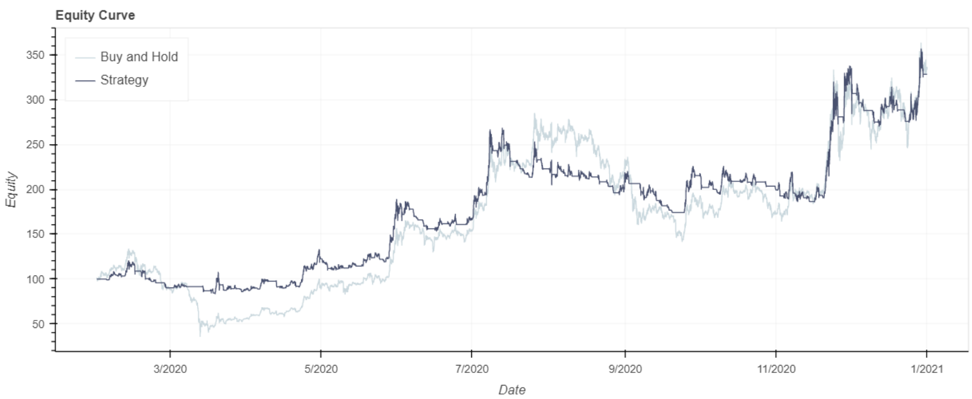


Figure 1.3: Cardano strategy performance with picky η

1.5 Drawbacks

This trading strategy involves a number of arbitrarily defined parameters. These should be subject to optimisation in the form of a grid search, parameter sweep or otherwise. The primary issue with doing that is the risk of over fitting. Past performance is not an indicator of future performance. We also face a great deal of coin-specific risk by trading a sole crypto. This should be dealt with by implementing a portfolio of coins. Fortunately, this strategy does scale well to that idea and signals can be generated simultaneously on multiple coins if desired, with cash allocation simply evenly split between assets.

Bibliography

[1] Jeffrey Chu, Stephen Chan, and Yuanyuan Zhang. High frequency momentum trading with cryptocurrencies. *Research in International Business and Finance*, 52: 101176, 2020.