QFin Momentum Trading Report

Trade Winners

11/2021

1 Momentum Trading

1.1 What is Momentum Trading?

Momentum trading is a widely used trading strategy where it involves trades that aim at capitalising on a potential market trend, whether bullish or bearish, by taking into account several factors such as price, volume, and other fundamentally economic events that would drive the market and create a trend. Thus, all of the trading indicators and tools that are used in our trading strategy are utilised to identify early buy and sell signals so we can spot any potential momentum trades. Therefore, The trading strategy that we have adopted consists of using two main indicators, RSI (Relative Strength Index) and three simple moving averages (SMAs), where these two indicators should be used together in sync so that fake buy/sell signals can be avoided and more reliable signals can be generated.

1.2 Relative Strength Index (RSI)

To illustrate further, RSI indicator is used since it informs us when and where would the market be considered to be overbought/oversold so that buy and sell signals can be generated. The way it is utilised in our trading strategy is that a buy signal is generated whenever the RSI indicator reaches a level of 30 or below, and a sell signal is identified whenever it reaches a level of 70 or above. Moreover, a parameter of 24 is used for the RSI because crypto-currencies operate in the market six days a week, thus a 24 parameter would calculate the RSI indicator over a period of month, and the rationale behind wanting to calculate the RSI indicator over a one-month period is because crypto-currency market is one of the most volatile markets and therefore the wider the period of time specified, the less fake signals we receive from the indicator itself since more data is considered from the market to reflect a clearer movement of the price market. Moreover, RSI can be useful in terms of acting as a stop-loss in cases of sudden crashes in the market, since an exponential moving average (EMA) is embedded in the calculation of the RSI, which means it will react quicker to any crashes because EMA tends to react much quicker to any changes in the market than simple moving average (SMA).

1.3 Simple Moving Averages (SMA)

On the other hand, since moving averages smooth out the data generated by the market price movement over a specified period of time, then using three moving averages together of different parameters (9, 14 and 30) would confirm the signal generated by the RSI indicator itself through utilising a strategy called crossover moving averages strategy where it creates buy/sell signals whenever these three moving averages cross above or below each other in a certain way. That is, any time we have a situation in the market where both 9 and 14 SMAs are above the 30 SMA, and 9 SMA is crossing below 14 SMA, then this gives a sell signal to traders as it is indicating that the current upward trend (bullish) has ended (or is soon ending) and a new potential downward trend (bearish) is beginning (or has already begun). On the other hand, if both 9 and 14 SMAs are below 30 SMAs, and 9 SMA crosses above 14 SMA, then this gives a buy signal to traders.

1.4 Market Strategy: Special Feature

However, the most important concept to note in all of this trading strategy is the added special feature that allows us, not only to identify a potential trend, but also to ride the trend from its onset so that we circumvent two major issues in our trading strategy; the issue of not using any volume indicator (such as ADI) and also the issue of lagging that is embedded in any momentum trading strategy system due to the requirement of waiting for signals to be generated from the indicators, which are solved by the added feature. This added feature is simply setting a rule of ignoring any buy/sell signals generated from both RSI and crossover SMAs strategy up until the point where the first wave of the three SMAs crossovers has lapsed, as this would enable us to know what is the overall market trend direction from its onset, or sometimes even before it starts, before entering a position.

2 The Algorithm

The python code for our trading logic is shown below (try catch block omitted):

```
def logic(account, lookback):

    # get the latest index
    today = len(lookback) -1

    # in case RSI_HISTORY > longterm lookback
    longest_lookback = max(moving_av_lengths['longterm'], RSI_HISTORY)

    # start trading after all moving averages have started
    if (today > longest_lookback):
        invested = account.buying_power <= 0    # we go all-in every time
                                                # only invested when buying power=0

        # get the 3 moving averages
        short_ls = moving_av_lengths['shortterm']
        medium_lookback = moving_av_lengths['midterm']
        long_lookback = moving_av_lengths['longterm']

        shortterm_moving_average = lookback['close'].rolling(window=short_lookback)
    .mean()[today]
        midterm_moving_average = lookback['close'].rolling(window=medium_lookback).
    mean()[today]
        longterm_moving_average = lookback['close'].rolling(window=long_lookback).
    mean()[today]

        # see if longterm is low or it is high
        longterm_is_low  = (longterm_moving_average < shortterm_moving_average) and
     (longterm_moving_average < midterm_moving_average)
        longterm_is_high = (longterm_moving_average > shortterm_moving_average) and
     (longterm_moving_average > midterm_moving_average)

        # -------------------- Moving Average Logic --------------------

        if longterm_is_high and not IN_FIRST_DIP:
            if (not invested):
                if (shortterm_moving_average > midterm_moving_average):
                    account.enter_position('long', account.buying_power, lookback['
    close'][today])
                    IN_FIRST_DIP = True
            else:
                for position in account.positions:
                    account.close_position(position, 1, lookback['close'][today])
                IN_FIRST_DIP = False

        # -------------------- RSI Indicator Logic --------------------

        rsi_score = rsi(lookback)[today]

        if RSI_LOW > rsi_score and not invested:
```

```
44              account.enter_position('long', account.buying_power, lookback['close'][
    today])
45              IN_FIRST_DIP = True
46
47          if RSI_HIGH < rsi_score and invested:
48              for position in account.positions:
49                  account.close_position(position, 1, lookback['close'][today])
50              IN_FIRST_DIP = False
51
52          if longterm_is_low:
53              IN_FIRST_DIP = False
```

The variable *moving_av_lengths* is a dictionary of constants for the window size of each rolling average. At *Line 25* the moving averages have been found, and we define boolean constants *longterm_is_high* which is true if the long term moving average is above the short and mid term averages.*longterm_is_low* is similarly defined.

The condition for buying and selling is, firstly, whenever *longterm_is_high* is true, and secondly when this is a dip we haven't made a buy in. *IN_FIRST_DIP* is a boolean flag that is true when we are in a dip where we have already bought. If *IN_FIRST_DIP* was excluded, the algorithm would sell instantly after buying, as the conditions to sell would be met every time.

At *Line 30*, we define the trade conditions relating to the moving averages - if we are not invested then we buy when the short term average passes over the mid term average, and set *IN_FIRST_DIP* to be true. Else, we sell and set *IN_FIRST_DIP* to be false.

Next, at *Line 42* we use an abstracted RSI method (found online and shown below) to find the current RSI score - *rsi_score*.

```
1  def rsi(df, periods=RSI_HISTORY):
2      """
3      Returns a pd.Series with the relative strength index.
4      source: https://www.roelpeters.be/many-ways-to-calculate-the-rsi-in-python-
       pandas/
5      """
6      close_delta = df['close'].diff()
7
8      # Make two series: one for lower closes and one for higher closes
9      up = close_delta.clip(lower=0)
10     down = -1 * close_delta.clip(upper=0)
11
12     # Use exponential moving average
13     ma_up = up.ewm(com = periods-1, adjust=True, min_periods = periods).mean()
14     ma_down = down.ewm(com = periods-1, adjust=True, min_periods = periods).mean()
15
16     rsi = ma_up / ma_down
17     rsi = 100 - (100/(1 + rsi))
18     return rsi
```

From *Line 44* we define the trading logic for the RSI indicator. Simply, if *rsi_score* is lower than our lower-bound, *RSI_LOW*, then we buy, if we aren't already invested, and set *IN_FIRST_DIP* to true. Else if *rsi_score* is higher than our upper-bound, *RSI_HIGH*, then we sell and set *IN_FIRST_DIP* to false.

Finally, at *Line 54* we set *IN_FIRST_DIP* to be false when we exit a dip.

Below are our instances of the global variables used in the final algorithm:

```
1      # Moving average window sizes
2      DAY_SCALE = 2*24
3      moving_av_lengths = {
4          'shortterm' : 9*DAY_SCALE,
5          'midterm'   : 14*DAY_SCALE,
6          'longterm'  : 30*DAY_SCALE
7      }
8
```

```
    # Thresholds for RSI
    RSI_LOW     = 35
    RSI_HIGH    = 75

    IN_FIRST_DIP = False    # Initially False
```

3 Testing

3.1 A Candidate Algorithm For Benchmarking

A candidate algorithm for benchmarking is buy-and-hold. This is because buy-and-hold has very good returns when the market is booming (like cryptocurrencies have been in recent years). So our focus with momentum trading would be having comparable or better performance but at a vastly better consistency than buy-and-hold.

Momentum trading makes profit over buy-and-hold by exploiting inefficiencies where buy-and-hold would make a loss due to the price dipping and closes its position to cut losses. This means that as long as the market has a long-term upwards trend it makes profit. It also means when there is a period where the price drops (slowly) momentum-trading makes a slight profit over buy-and-hold. It can also check for other factors (oversaturation of the market in our case) in order to gain an advantage over buy-and-hold.

3.2 Comparison With Buy-and-hold

Below are 2 examples of using the original 9-14-30 day parameters vs the higher frequency 25-30-35 half-hour parameters for the look-back period of our moving averages.

Expected in the high-frequency and high-noise market longer windows tend to outperform shorter windows as they are not as susceptible to the random movements present in the market and tend to react to real changes in the market instead, even outperforming buy-and-hold in many instances.

3.3 Final Parameters

Unfortunately due to bugs and time constraints the decision of our parameters had to be based on financial experience rather than a rigorous statistical method. We therefore decided on using 9-14-30 (at a daily scale) as our final parameters for the period of the rolling windows for our three moving averages as it *seemed* to grant the best returns as indicated in the figures below.

Figure 1: Ripplecoin - parameters: (9 days, 14 days, 30 days)

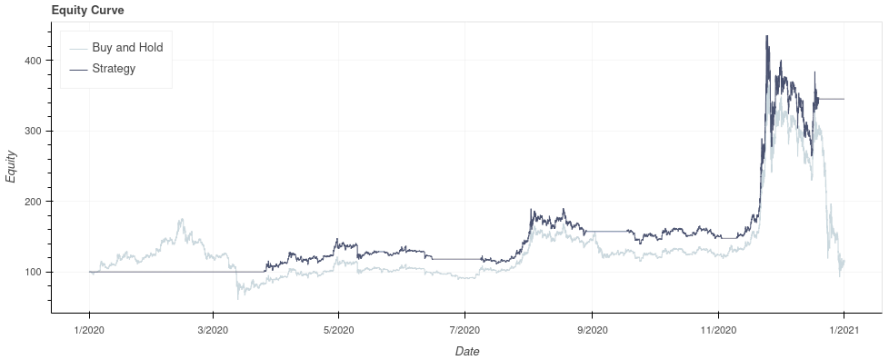


Figure 2: Litecoin - parameters: (9 days, 14 days, 30 days)

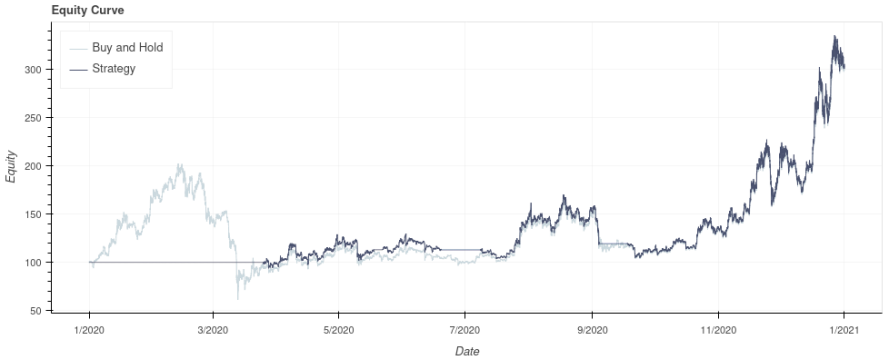


Figure 3: Ripplecoin - parameters: (25 half-hours, 30 half-hours, 35 half-hours)

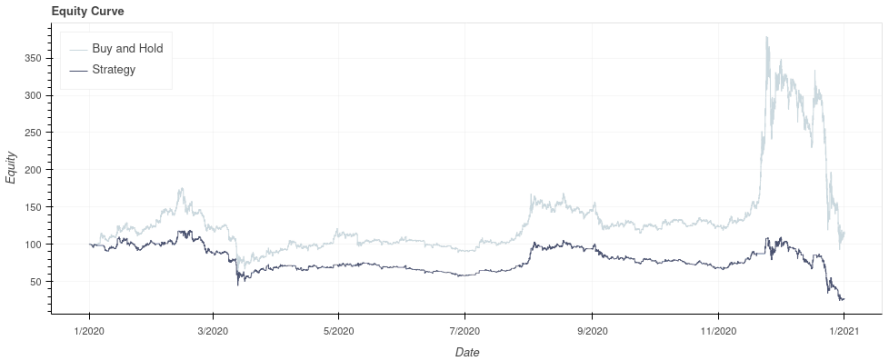
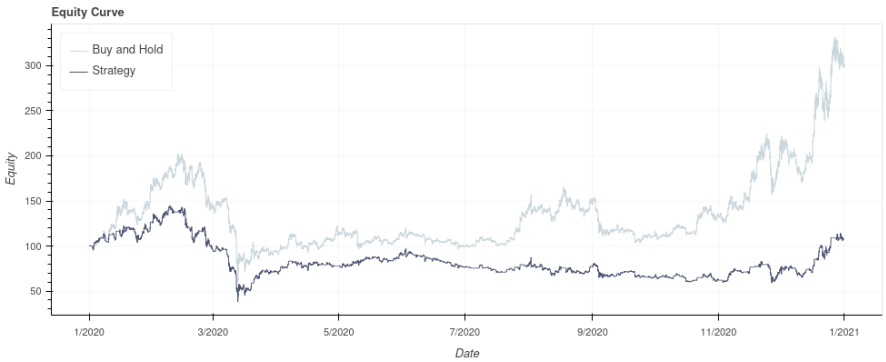


Figure 4: Litecoin - parameters: (25 half-hours, 30 half-hours, 35 half-hours)



4 Results and Expectations

As our strategy is based on momentum then the expected return is also based on momentum. The longer that momentum is high and the price is increasing the more profit that we make. However for our algorithm to make more money than buy-and-hold it has to have many periods where the price decreases in order to capitalize on the inefficiencies of the buy-and-hold strategy by selling for those periods.

As such we will expect at least an incrementally higher return than the buy-and-hold strategy during the testing of our algorithm as cryptocurrencies are not expected to spend long stretches of time slowly declining in price.

5 Limitations

5.1 Crashes

It might be interesting to investigate ways to mitigate crashes. As our shortest moving average window is 9 days in duration and RSI although fast might not respond to near-instantaneous crashes: our algorithm can sometimes take days to catch up with the changing state of the market and so it cannot respond effectively to sudden crashes.

It might be interesting to put something similar to a stop-loss that will cause a withdrawal of the current market position if the price were to suddenly drop.