QFin Trading Team 1

Range Trading

—

Semester 2 2022

Conor Bennett, Zachary Ching, Luke Osborne

Contents

1 Range Trading

1.1 What is Range Trading?

Range trading is a form of high-frequency trading which relies on exploiting temporary fluctuations in the price of an asset while the market "ranges"; i.e. is not trending but moving sideways. By assuming that the price of the asset will remain within the established ranges most of the time, we can conclude that when the price is near the lower bound, we should enter a long position as the price is likely to increase (as it cannot decrease much further without breaking out of the bounds), and vice versa for when the price is near the upper bound. Of course the price may break out of the established ranges, for which case stop losses sitting outside the ranges are employed.

In order to successfully range trade, there are two key elements necessary:

- identifying when the market is ranging or trending
- when the market is ranging, identifying the bounds between which it is ranging

How our team implemented these two aspects of the problem will be discussed in further sections of the report.

1.2 Indicators

1.2.1 Average True Range

Average True Range (ATR) is a technical analysis indicator used to view market volatility. Here, a security with a high level of volatility has a higher ATR, and vice versa. ATR is simple to calculate from historical price data and is a useful tool to be added to filter through different trade opportunities.

ATR has a wide range of uses, namely, it can be used to enter and exit trades while also giving traders an indication on what size trade to make in a derivatives market. However, limitations do exist, ATR is a subjective measure which means that no single ATR value will provide much valuable information. This falling can be compensated through comparisons against earlier readings to understand a trend's strength or weakness. Additionally, ATR only measures volatility and not the direction of the asset which may result in mixed signals, especially when markets are experiencing pivots or turning points in trends.

1.2.2 Bollinger Bands

Bollinger Bands are technical analysis tools that create a pair of trendlines along the price data that indicate significant deviations from the simple moving average (SMA) price. As the price approaches and passes the lower band, it provides an indication that the security is being oversold, and hence a significant decrease in price occurs, providing opportunity to purchase at a below-market rate. Conversely, as the price of the security approaches the upper band, it indicates that the security is being overbought, driving up the price and allowing for a profit on short positions on return to the SMA price.

Bollinger bands are used when the security is no longer ranging horizontally, and is most effective at capitalising on market fluctuations outside the defined standard deviation boundaries. The profitability of systems using this indicator relies on the assumption that the price will return to within the bands and the SMA over time, and therefore persistently rising and falling prices present an issue. In many cases, however, this can be accounted for by restricting positions made through Bollinger bands to only be created when a momentum indicator (eg. average directional index) is below a threshold value. This is certainly an area our algorithm can improve upon and utilise for future improvement.

1.2.3 Donchian Channels

Donchian Channels are an indicator we considered before building our algorithms. The indicator works to determine bullish and bearish trends and examine higher and lower breakouts.

The channels are determined by the Highest High in the training period, typically 14 periods and the Lowest Low in the same training period. Additionally, a median exists which is the mean of the upper and lower bounds.

1.3 Trading Strategy

Our trading strategy was to experiment different ideas using a range of different indicators to implement logic in Python. Through rigorous trial and experimentation we determined our key indicators and how we would utilise them as primary parts of our solutions.

2 Nature of Data

Our algorithm was developed to run on data of stocks from the S&P500 with 15-minute intervals consisting of the open, close, low and high prices, as well as the volume of stock traded during that 15-minute period. During development, our algorithm was tested against TSLA and AAPL stock data from 2020, and will be finally evaluated against a selection of random stocks from the S&P500, with data of a similar format.

3 Trading Algorithm

Throughout our report we narrowed down our trading strategy to two key algorithms based around Bollinger Bands and Range Trading separately.

Bollinger bands create a simple rolling average price over a fixed time period, and then define an upper and lower band set a fixed number of standard deviations above and below this average respectively. The standard deviation used to calculate this width is also calculated using a fixed width rolling interval of time series data.

In our algorithm, we decided to use three different Bollinger Bands to analyse our data, with varying parameters for 'short', 'medium' and 'long' term bands. Most standard Bollinger Bands rely on a time frame of 20 points to calculate SMA and standard deviations, but in our examples, we used 10, 20 and 50 for our short, medium and long bands respectively. Similarly, we also implemented different 'widths', ie. the number of standard deviations above and below the SMA each of our bands sits. The typical value is ± 2, but we chose ± 1.5, ± 2 and ± 2.5 respectively.

As the security price moves outside the boundary created by the Bollinger bands, a long or short position is entered based on if it passed the lower or upper bound respectively. Once the price returns to within ± 0.1 standard deviations of the SMA, the position is closed. At a crude level, the more bands the price passes, the more heavily weighted the positions taken are. In future, the different bands should be assigned different weightings with respect to the positions entered when passing them. As the long-term band is wider and relies on a greater data set to create, it follows that more purchasing power should be weighted to enter these positions than other potentially less reliable options. To refine this algorithm further, it could theoretically be simplified to one band, with a variable buy-value function that adjusts the amount bought proportionally to the distance from the boundary.

The first step to utilising Range Trading lied in the name and was to determine when the data was in a range. To solve this, we utilised Average Directional Index (ADX), an extension of Average True Range. Utilising ADX, we considered our data to be ranging when ADX was below 25.

As prices were ranging, we determined our upper and lower trading bounds which included a standard deviation buffer. This would allow us to take advantage of any prices that dropped below a certain price or rose above, and take advantage of any reversals (commonly seen in ranges). To take profits sell signals were implemented, with additional sell signals implemented to accommodate for stop losses.

4 Results and Findings

While both algorithms performed decently on our historical data, key differences between laid in the fact that Range Trading only traded while the market was moving horizontally whereas, Bollinger bands traded at any time whenever the entry conditions were met. This resulted in Bollinger Bands trading more frequently which resulted in higher returns despite the additional fees from trading more frequently. The higher return however, could be attributed to the difference in logic implementation and trading strategy rather than the base ideas in range trading and bollinger bands.

5 Future Improvements

The current method of parameter sweeping is extremely inefficient as it manually iterates over all possible combinations of parameters. This results in parameter sweeping requiring a large amount of time to process. A much more efficient optimisation technique that could be employed is gradient descent; an algorithm by which the parameters are treated as variables of a multivariable cost or reward function, and the grad of the function is used to determine the direction for iterative steps to minimise or maximise this function. This allows for a much quicker traversal of the parameter space, but is a much more heuristic method than a brute force parameter sweep, which can pose issues. For example, gradient descent algorithms can get stuck at local minima or maxima, and can have convergence issues. These problems are not insurmountable, however require more sophisticated modifications to the gradient descent algorithm to be overcome.