



Range Trading Algorithm

“Trade the Range and Ride the Trend”

Trading Team 3: Lucas Siciliano, Isaac Bergl and Henry Miao

A 2022 Semester 2 Project Completed by the Trading Team in conjunction with

Quantitative Finance UWA

Introduction

Range trading is **an active trading strategy that identifies a range at which the investor buys and sells at over a short period**. Range trading is best utilised in sideways markets, when prices oscillate between an upper and lower boundary. Conversely, a trending market represents a risk as it would negate the value of a range trading strategy. When market conditions change, a range trading strategy can either adapt or pause trading until the conditions are favourable.

Indicator Selection

There are numerous indicators that can be used to implement a range trading strategy. We decided to build our trading strategy on three specific indicators described below. We then tested different hyperparameter values to try and optimise the performance of our algorithm.

Indicator 1: Average Directional Index

Average directional index (ADX) is an indicator used to determine the strength of a trend. It is a combination of two momentum indicators, the positive directional indicator (+DI) and negative directional indicator (-DI). ADX signals a trendless market when it is below 20 and signals a trending market when it is above 25. This allows us to determine whether the timing is right for trades to be taken, as our range trading strategy is most suited to sideways markets. However, ADX does not help us identify the direction of a trending market, for that we need to use +DI and -DI.

Indicator 2: Bollinger Band

Bollinger band is a price envelope indicator that defines an upper and lower price range level. The band distance is calculated based on a set number of standard deviations above and below a moving average price. Bollinger band allows us to determine the trading ranges for our algorithm and is used as the main confirmation signal for the execution of our trades. We take the lower band to be the support level, and the upper band to be the resistance level as price moves between the two bands.

Indicator 3: Relative Strength Index

Relative Strength Index (RSI) is a momentum indicator that is used to evaluate overvalued or undervalued conditions in stock prices. A security with an RSI below 30 is considered to be oversold and therefore undervalued, while a security with an RSI above 70 is considered to be

overbought and therefore overvalued. We utilise RSI as an additional check for our trade execution in case of price breakouts.

Strategy

Our strategy has four main components:

- i) Identifying a sideways market (ADX)
- ii) Defining a dynamic range (BOLL)
- iii) Using the range to make trades (BOLL + RSI)
- iv) Hold/Sell appropriate position in trending market (+DI/-DI)

Identifying Sideways Market:

Identifying when a market is 'sideways' is an important first step in our strategy. For this, we utilised the Average Direction Index indicator (ADX).

Table 1: Credit Investopedia

ADX Value	Trend Strength
0-25	Absent or Weak Trend
25-50	Strong Trend
50-75	Very Strong Trend
75-100	Extremely Strong Trend

We only traded our strategy when ADX was below 25 to target markets with no trend - the ideal condition for our range trading algorithm to function effectively.

Defining Dynamic Range:

Our trading range was created using Bollinger Bands and ADX. When ADX is low (<10), it is believed there is little to no trend. Thus, as ADX decreased, we tightened our Bollinger bands, as we were more confident the price would trade within a smaller range. Conversely, as ADX increased, we widened our Bollinger bands, as there is more uncertainty regarding the trading price. We used a standard deviation of 2.8 for our Bollinger bands when ADX was less than 10, and 3.5 when ADX was between 10 and 25. Despite our use of a piece-wise

function, realistically we could use a number of decreasing functions for Standard Deviation (as a function of ADX) when optimising our results.

Using the Range to Make Trades:

Once our Bollinger bands were set, our algorithm could now trade within them. We utilised a simple Bollinger band strategy in combination with RSI to find profitable trades. When price moved above our upper Bollinger band, and the security was overbought (indicated by an $RSI > 70$), we entered a short position. Similarly, when price moved below our lower Bollinger band and was oversold ($RSI < 30$), we entered a long position. Using RSI as an additional indicator alongside Bollinger bands allowed us to avoid breakouts and make trades that are more likely to be profitable, which greatly enhanced the algorithm performance.

Hold/Sell Appropriate Position in Trending Market:

As stated earlier, when a market was trending ($ADX > 25$) we didn't want our algorithm to trade. Initially, we made sure our algorithm never traded when ADX was over 25. However, this led to issues where whatever position we were holding when ADX was under 25, wasn't sold until ADX was back under 25. To ensure we would not get caught with a bad position, we decided to add a final feature to our algorithm that re-adjusted our positions when the market was trending and the price was outside our Bollinger bands. This was achieved by utilising +DI and -DI, which were key components in the calculation of ADX. Specifically, when the market was trending downwards ($-DI > +DI$), our algorithm would sell our long positions while maintaining any short positions with the aim of profiting from the downward trend. Conversely, when the market was trending upwards ($+DI > -DI$), our algorithm would sell our short positions while maintain any long positions to profit from the upward trend. This additional specification in our strategy improved our performances across all stocks that we back-tested on.

Results

We tested our model with two different categories of stocks; upwards trending and sideways market. APPL and TSL from 2020-2021 were used as the upwards trending stocks and WMT and NDAQ from 2020-2022 were used as the sideways trending stocks.

The upwards tending stocks consistently underperformed buy and hold at all stages of our strategy, with our strategy losing money on both TSLA and APPL. The sideways market however proved to be well-suited to our strategy as we were able to exceed buy and hold by a significant margin. We beat buy and hold strategies on both WMT and NDAQ by considerable margins.

Discussion

Positives:

- Strategy achieved goal, in performing well in sideways markets
- Strategy is easily implemented and simple to understand
 - It is easy to make changes to indicators and parameters in our range trading strategy
- As a team, we developed skills in Python and gained a greater understanding of markets

Improvements:

- Optimising hyperparameters
 - We did very little optimisation, as our focus was on developing a sound trading strategy first and foremost
- Bollinger bands don't necessarily have to be symmetrical
 - We can adjust Bollinger bands based on how a stock is trending
 - Eg: If a stock is upwards trending, increase the Bollinger bands distance from the mean
- As stock moves and trends, RSI bounds can change, but in our algorithm we use 70 and 30 as fixed values
- Utilising a greater lookback window for variables
 - For ADX we used a 3.5 hour lookback window, perhaps using longer lookbacks would give us a better idea of how the market has previously trended

Appendix: Implementation Details

Our full code is available on our [GitHub page](#).

Mean Average Calculation

For a variable *TRAINING_PERIOD*, set to 14 for our implementation

```
# Calculate Typical Price
df['TP'] = (df['close'] + df['low'] + df['high'])/3
# Calculate Standard Deviation
df['std'] = df['TP'].rolling(TRAINING_PERIOD).std()
# Calculate Moving Average of Typical Price
df['MA-TP'] = df['TP'].rolling(TRAINING_PERIOD).mean()
```

ADX Calculation

For a variable *INTERVAL*, set to 14 for our implementation

(LUCAS - Change this, I changed stuff to get +DI and -DI)

```
# Directional Movement : previous low minus current low
df['-DM'] = df['low'].shift(1) - df['low']
# Directional Movement : current high minus previous high
df['+DM'] = df['high'] - df['high'].shift(1)
df['+DM'] = np.where((df['+DM'] > df['-DM']) & (df['+DM'] > 0), df['+DM'], 0.0)
df['-DM'] = np.where((df['-DM'] > df['+DM']) & (df['-DM'] > 0), df['-DM'], 0.0)
df['TR_TMP1'] = df['high'] - df['low']
df['TR_TMP2'] = np.abs(df['high'] - df['close'].shift(1))
df['TR_TMP3'] = np.abs(df['low'] - df['close'].shift(1))
df['TR'] = df[['TR_TMP1', 'TR_TMP2', 'TR_TMP3']].max(axis=1)
df['TR'+str(INTERVAL)] = df['TR'].rolling(INTERVAL).sum()
df['+DMI'+str(INTERVAL)] = df['+DM'].rolling(INTERVAL).sum()
df['-DMI'+str(INTERVAL)] = df['-DM'].rolling(INTERVAL).sum()
df['+DI'+str(INTERVAL)] = df['+DMI'+str(INTERVAL)] / df['TR'+str(INTERVAL)]*100
df['-DI'+str(INTERVAL)] = df['-DMI'+str(INTERVAL)] / df['TR'+str(INTERVAL)]*100
df['DI'+str(INTERVAL)+'-'] = abs(df['+DI'+str(INTERVAL)] - df['-DI'+str(INTERVAL)])
df['DI'+str(INTERVAL)] = df['+DI'+str(INTERVAL)] + df['-DI'+str(INTERVAL)]
df['DX'] = (df['DI'+str(INTERVAL)+'-'] / df['DI'+str(INTERVAL)])*100
df['ADX'+str(INTERVAL)] = df['DX'].rolling(INTERVAL).mean()

df['ADX'+str(INTERVAL)] = df['ADX'+str(INTERVAL)].fillna(df['ADX'+str(INTERVAL)].mean())
del df['TR_TMP1'], df['TR_TMP2'], df['TR_TMP3'], df['TR'], df['TR'+str(INTERVAL)]
del df['+DMI'+str(INTERVAL)], df['DI'+str(INTERVAL)+'-']
del df['DI'+str(INTERVAL)], df['-DMI'+str(INTERVAL)]
del df['+DI'+str(INTERVAL)], df['-DI'+str(INTERVAL)]
del df['DX']
```

RSI Calculation

```
def calc_rsi(data, periods=14):
    close_delta = data['close'].diff()

    # Make two series: one for lower closes and one for higher closes
    up = close_delta.clip(lower=0)
    down = -1 * close_delta.clip(upper=0)

    ma_up = up.ewm(com=periods - 1, adjust=True, min_periods=periods).mean()
    ma_down = down.ewm(com=periods - 1, adjust=True,
                       min_periods=periods).mean()

    rsi = ma_up / ma_down
    rsi = 100 - (100 / (1 + rsi))
    return rsi
```

Trading Logic (pseudo code)

note: this doesn't include that when $ADX > 25$, depending on +DI and -DI (direction of trend) we sell off specific positions (as said in report)

```
def logic():

    if(before training period):
        don't trade and return

    if  $ADX \geq 25$ :
        don't trade and return

    if  $ADX \geq 20$  and  $ADX < 25$ :
        set Bollinger bands to be 2x as wide
    elif  $ADX > 10$  and  $ADX < 20$ :
        don't change Bollinger bands width
    else:
        set Bollinger bands to be 1/2 as wide

    if(close price < lower Bollinger band):

        if  $RSI >$  upper RSI limit:
            don't trade and return

        close all short positions
        enter long positions at current price

    if(close price > upper Bollinger band):

        if  $RSI <$  lower RSI limit:
            don't trade and return

        close all long positions
        enter short positions at current price
```