Return Predictions From Trade Flow

January 30, 2024

1 Introduction

Here you will assess trade flow as means of generating profit opportunities in 3 cryptotoken markets. We stress the word "opportunity" because at high data rates like these, and given the markets' price-time priority, it is far easier to identify desirable trades in the data stream than it is to inject oneself profitably into the fray.

2 Data

We have preprocessed level 3 exchange messages from the Coinbase WebSocket API for you into a more digestible format of truncated level 2 data.

2.1 Treatment

Load the 2023 data for all 3 pairs from the class website. For each one, split it into test and training sets, with your training set containing the first 40% of the data and the test set containing the remainder.

2.2 Format

The data has the following structure

2.2.1 Trades

received_utc_nanoseconds	timestamp_utc_nanoseconds	PriceMilliionths	SizeBilliionths	Side
1674521267814309000	1674521267874527000	22970120000	87069600	-1
1674521267814046000	1674521267874527000	22970150000	25797600	-1
1674611962312088000	1674611962347434000	22499070000	4801640	-1
1674611962339264000	1674611962375191000	22498910000	1120200	-1

The *Side* is actually a sum of trade sides at the same price and time.

2.2.2 Book

Ask1PriceMilliionths	22972550000	22972550000	22502670000	22502670000
Bid1PriceMilliionths	22970150000	22970150000	22498910000	22498910000
Ask1SizeBilliionths	210000000	410000000	101856140	101856140
Bid1SizeBilliionths	25797600	25797600	280050	280050
Ask2PriceMilliionths	22972560000	22972560000	22502680000	22502680000
Bid2PriceMilliionths	22970120000	22970120000	22498690000	22498690000
Ask2SizeBilliionths	210000000	210000000	50000000	50000000
Bid2SizeBilliionths	87069610	87069610	12560150	12560150
received_utc_nanoseconds	1674521267750919800	1674521267751154000	1674611962359972000	1674611962365237000
timestamp_utc_nanoseconds	1674521267806932000	1674521267807073000	1674611962398574000	1674611962400579000
Mid	22971350000	22971350000	22500790000	22500790000

(transposed)

Here, the received time comes from the clock of the recording device, which was not synchronized to the exchange clock. Such inaccuracies in clock settings, i.e. "clock skew", can cause exchange timestamps to appear later than the time at which they are recorded as having been received.

As noted in class, exchange timestamps are not actionable, in the sense that any market participant would not see the data until considerably later. On the other hand, received timestamps, while actionable, may be subject to poor recording techniques on the client side. For this homework you may choose either, but I recommend the exchange timestamps.

3 Exercise

Write code to find τ-interval trade flow $F_i^{(\tau)}$ just prior[1] to each trade data point[2] i. Compute T-second forward returns[3] $r_i^{(T)}$. Regress them against each other in your training set, to find a coefficient β of regression.

For each data point in your test set you already have $F_i^{(\tau)}$, so your return prediction is $\hat{r}_i := \beta \cdot F_i^{(\tau)}$. Define thresholds j for $\hat{r}_i$ and assume you might attempt to trade whenever $j < |\hat{r}_i|$. Good values for j will have relatively frequent participation, but not anywhere near 100%.

4 Analysis

Assess the trading opportunities arising from using these return predictions in your test set, both with and without trading cost assumptions. Examine Sharpe ratios, drawdowns and tails. As part of this assessment, comment on the reliability/stability of β (most easily done by further splitting the data set), how you chose j, and what you might expect from using much longer training and test periods.

[1] We do not include the trade i data itself, because we are evaluating trade i in terms of the flow we would have been aware of just before it happened.

[2] NOTE: the trade data series does not necessarily have strictly increasing timestamps. Be sure not to include other trades at the same timestamp in your computation of F_i.

[3] You need not handle latency in your homework, but for your edification: a more careful implementation would account for lags. For a pessimistic approach we could choose L as, say, twice the 99th percentile of computational and communications lag. Then, it would use book data (not just trade data) to help compute return from time $t_i + L$ to $t_i + L + T$ and run regressions using that. The idea here is that it takes approximately time L to "do anything" about trade information.