Order Imbalance Market Making Strategy

Juan Obligado

1. Overview of your market-making strategy and its implementation.

We implemented a Market making strategy that every 1 second updates 1 buy and 1 sell limit order which are separated by a given target spread..

In order to enhance our signal we wanted to skew both orders to a higher price in case that the market balance is bought (since there are more buyers than sellers we can sell our goods a little bit higher), alternatively we skew to the lower end in case that the market is sold..

In order to measure how much we should skew our orders, we use the Order Book Imbalance metric (OBI) which is defined askey-1:.

Definition 1. $OBI = \frac{V_{bid} - V_{ask}}{V_{bid} + v_{ask}}$

This signal belongs to the interval $[-1, 1]$, with 1 for the case where all orders are buys, -1 all the orders are sells and 0 for a balanced book. So from a general point of view our strategy will calculate the OBI and generate a skewed reference price p_{ref} for our buy and sell orders accordingly with the following formula:

Definition 2. $p_{ref} = p_{mid} + OBI_t * f(spread, position, t)$

In order to perform a super simple implementation we choose f to be just a constant which is a proportion of the mean spread amount. However if having more time we should consider different candidates and scenarios for $f()$*..*

2. Analysis of your strategy's performance, including the impact of latency.

Strategy performance can be seen in Figure 1. We noticed that we have a low amount of trades for a single day. However the Order book imbalance helped deciding the side of the trades. Anyway the amount of trades seems really low and would need to be checked in further steps..

Parameter	w/o Latency	w/ Latency
Sharpe	18.95496	18.95496
Number of trades	9.031	9.031
Return	52488	52488

Table 1: Strategy performance comparison

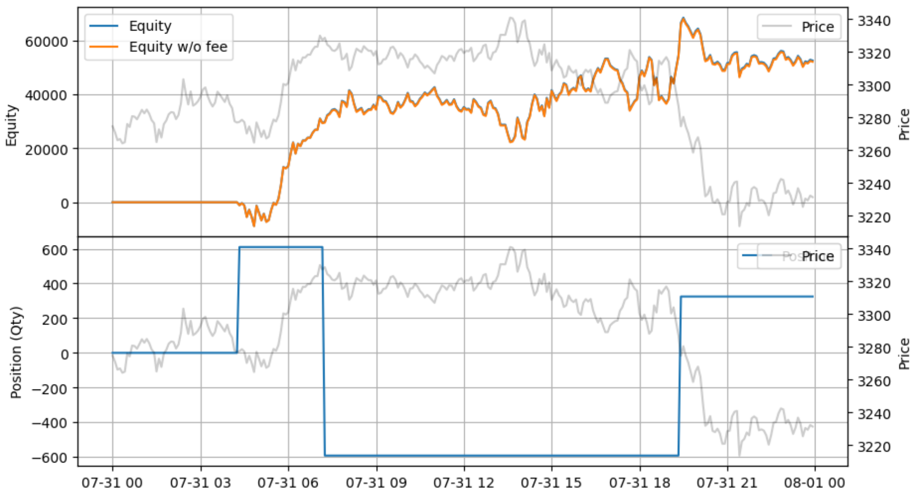

Figure 1: Strategy Performance without simulating latency

In addition with running the performance, we backtested the strategy using latency data that we generated from the data feed. Taking the latency from the feed we applied a proportional latency for order placement and generated a latency file for simulation. Unfourtunately this file didnt resulted in much difference from a latency perspective and the strategy end up picking the same trades. This could be due to having a low trade frequency or not enough order latency in our simulation parameters..

3. Discussion of your optimization approach and its effects.

To be completely honest, without having a production ready environment and being able to measure where the bottlenecks are. Optimization is really more a theoretical exercise than a real world application.. In a real life scenario would first measure which is the most critical component having

1. Measure latency for price Gateway
2. Measure Book preparation performance
3. Measure Signal calculation
4. Measure order gateway latency

With this amount of time the only parameter for optimization was to implement the strategy in Rust, however this would probably wont be the first choice in real life but order gateway, since latency tends to be greater here (reason is that exchange api need to perform risk checks before authorizing any order in addition that sending messages to remote apis can be measured easily in milliseconds rather than microseconds.

Being said that after implementing the strategy in Rust we were able to reduce backtest running time from 32 minutes to 6 minutes, however noticed that data warmup in Rust takes considerably more time than in Python

4. How you'd improve the strategy given more time.

"I think 99 times and find nothing. I stop thinking, swim in silence, and the truth comes to me", Albert Einstein

There is still plenty of stuff to do, at some point this exercise is just a first step. If having more time would:.

- Perform a more in depth analysis of the market data (we only analyzed a few days and a single market), analysis would need to be extended to multiple pairs and a bigger dataset.
- Include unit tests to reach near 100% coverage
- Implement proper price and order gateways
- Define and implement monitoring tools
- Make sure about having rebalance flow ready to have enough money to trade the strategy.
- Also since not being an expert in market making literature would like to go through some papers and grasp analysis best practices (think that references from below would be a good start)
- Explore adding delta hedging, so we can cover again volatility on the underlying assets. Some simple way which we could do this would be buying out of the money put to cover against price drawdowns in the crypto asset mainly.

References

[1] Alvaro Cartea, Ryan Donnelly and Sebastian Jaimungal: Enhancing Trading Strategies with Order Book Signals, 2004.

[2] Sasha Stoikov. The Micro-Price: A High Frequency Estimator of future prices, 2017.