Hackathon Challenge 1: Team Swordfish - Report

Background

The task we have taken upon ourselves is building a framework for extracting the atomic data present in the Websocket output of CEXs.

Within this problem, we aim to address a key challenge:

The issue of aggregated data that is generally encompassed in the events reported from the crypto exchanges, leading to not being able to discern the underlying granular details of a limit order book. We aim to address this challenge by developing a framework for normalizing limit order and real-time trading dataset, received from ByBit Public Websocket server at 100 ms intervals.

Github repository: https://github.com/huynguyends/CryptoOrderBook

This document is divided into three sections, outlining the key steps and results from our proposed solution:

Data Collection

The datasets were obtained from the websocket API available from ByBit exchange. We have used the OrderbookL2_200 and trade data that is available from the public websocket endpoint of ByBit

The chosen datasets contain rich information about the market movement of BTCUSD pair at both the limit order and real-time trading level. Both of the data were fetched simultaneously at a push frequency of 100 ms so that they can be used to disentangle the underlying atomic structural details of the limit order book.

The dataset from OrderbookL2_200 contains a snapshot, which essentially is an overview of the entire order book of BTCUSD pair, 200 levels deep. In addition to price, the snapshot also contains information about their size, side and order id. Following the snapshot, the OrderbookL2_200 also provides aggregated metadata information (delta) about the updates. The delta response contains information about the delete, update and insert updates relative to the previous response at microsecond resolution. Furthermore, the delta response also contains a proprietary variable, 'cross_seq', which will be crucial in our efforts in data normalization (more on this later).

Data Normalization

Normalisation is aimed to simplify extraction of data metrics from the order book, especially where these metrics are aggregating data. In practice, the response from websocket will be used to rebuild the limit order book in real-time, providing access to high resolution atomic crypto market events for the end-users. For the purpose of our scope of work, we will be undertaking an offline demonstration of websocket data obtained at a 5 second interval from the websocket.

On a high level, normalization consists of making relational connections between multiple data points within our datasets, as illustrated in **Fig.1** below. In summary, using JSON and NumPy Python libraries, we have utilized the cross-seq and price variable to establish the connection between real-time trading and delta datasets. Upon establishing this initial relationship, we are able to utilize this information along with the delta dataset to deconvolute the aggregated event details through the design of element-wise operation on the snapshot using NumPy array operations.

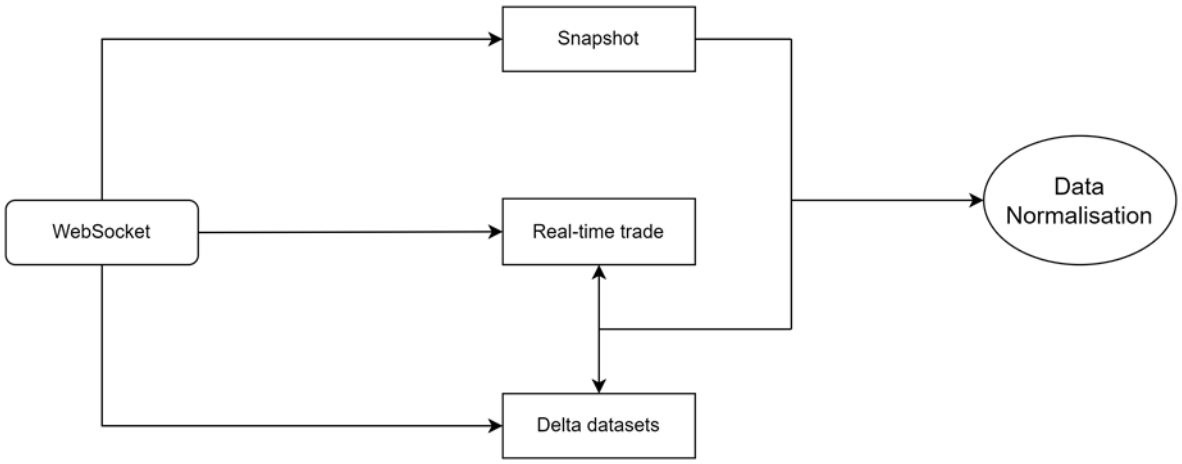

Figure 1: Data normalization flow chart

In conclusion, our normalization strategy clocked an average of 37 milliseconds over a 5-second data stream using the Python wrapper %prun.

```
<string>:32(normalize_task)
        497    0.003    0.000    0.037    0.000
```

Next Steps

In addition to that we also aim to functionalize the normalization script by using NumPy vectorization and C binding framework such as Numba or Cython. Using the normalized dataset, we will also be using established methods for calculating some key financial metrics such as OFI and TFI that could be used to predict market manipulative behaviour.

Additionally, future work could further foresee the development of the proposed framework in production by building the limit order book in memory with state-of-the-art data pipelines implemented using additional API endpoints and machine learning techniques to develop a deeper understanding of the key market behaviours. Further enhancement can also be obtained through the use of multiple centralized and decentralized exchanges, facilitating high accuracy and reliability to the data stream.

Appendix

Timeline

There were three days to complete the challenge. These three days were assigned activities according to achieving the

- Day 1

  The first day was aimed to have a clear understanding of the task and the objective that we had in mind, after grouping into teams and learning about methodologies that eventually eased our performance individually and as a group. The first day can be assigned to the headings of this documentation

- Day 2

  The second day was aimed at thinking of methods to reach the goal and working on reaching it, the objectives being collecting and saving 100ms of data for testing the data, normalising the 100ms of data and benchmarking the performance, and collecting and saving 5 seconds of data. Throughout the day, C-Binding was researched to use in the microservice, achieving execution time, learning from videos that the staff has suggested, API management, deciding how to normalise the data and storing data in memory.The second day can be assigned to the contents of this documentation.

- Day 3

  Third day was devoted to finalising the framework with specific focus on enhancing the performance of data normalization and metric evaluation.

Information and Obstacle Visualisation Methodology

We have written our doubts and thoughts throughout the hackathon on colour-coordinated sticky notes that have gone on the wall. This, much like tables and graphs, offers the visual benefit of organising information to make connections and patterns between ideas and difficulties more intuitive, as well as some between ideas and difficulties: paths of solving problems utilising information that we have had on the wall.

Double Diamond Methodology

This problem solving methodology consists of four stages in the following order: coming up with many ideas for solving a problem, narrowing them down to discover the most crucial one, devising many ideas for solving the problem, and narrowing all solutions to the one with the most potential to serve the purpose of discarding the problem.

Coalescence of the Two Methodologies

The Double Diamond methodology has been merged with the visualisation methodology by approaching the former's two idea stages using sticky notes to contain our individual ideas and deciding on patterns and connections to decide which idea we should pursue. The decision process includes a process of elimination, made by discarding notes, overall resulting in two waves of notes – analogous to the two diamonds.

Obstacles and Troubles

The foreign, demanding nature of the challenge posed incremental difficulties with understanding the task and how we would do well to set about reaching the intended outcome. While confusions were suppressed more quickly thanks to the mutual assistance from group members – occasionally utilising red sticky notes for writing those confusions – multiple times our team has encountered difficulties none of us knew how to overcome. Thankfully, unlike problems encountered in start-up and research where material is entirely new, we received ample help and direction from staff.

In the last resort case where a difficulty arose which couldn't be solved at all, we were left with no choice but to discard that path of filling the task and seek another way to answer the problem.