TradeMaster Sandbox Whitepaper

Haochong Xia
Nanyang Technological University
HAOCHONG001@e.ntu.edu.sg
May 2023

1 Abstract

Implementing DRL agents to live trading markets suffers from the risk caused by the simulation-to-reality gap. To control the unseen risks, we made an evaluation sandbox introducing diverse evaluation methods into the life cycle. The TradeMaster Sandbox evaluates QT algorithms in various paper-trading scenarios. We want to conquer the sim-to-real gap for QT algorithms with the use of this sandbox.

2 Tools

2.1 Market Dynamics Modeling

2.1.1 Background

Market dynamic is a long-studied and not well-established market feature. The market dynamics are formulated differently according to specific tasks and markets, while the markets are commonly classified into 'Bear', 'Volatile', and 'Bull' categories, the labeling method is highly subjective. Leading indicators(e.g., S&P 500, Dow Jones Industrial Average, NASDAQ Composite) are not always accurate and often fail to capture the specific need of the task. On the other hand, there are data-driven methods to classify the state of the market, for example, the Hidden Markov Model. These machine-learning techniques often lack explainability and interpretability.

2.1.2 Method

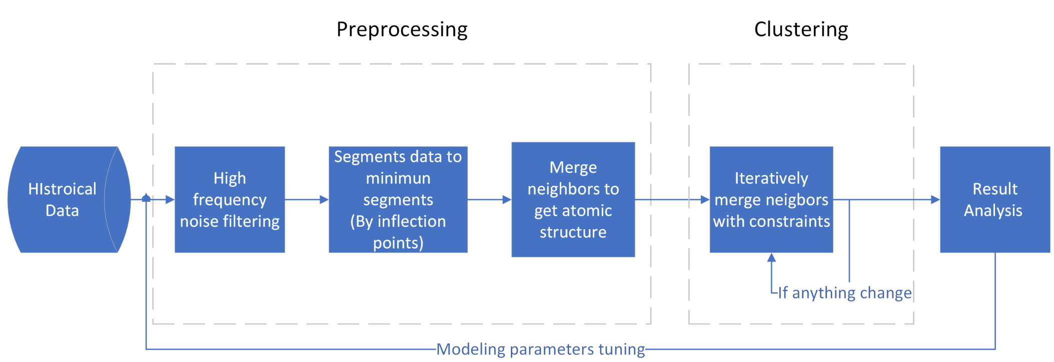

Figure 1: Framework of market dynamics modeling

We built a market dynamics modeling tool that models the dynamics of given data. The framework of this tool is shown in Fig 1. This modeling tool provides an interactable way to slice long historical data into chunks of different market dynamics.

Algorithm 1 Market Dynamics Modeling

1. Filter the data to eliminate high-frequency noises.
2. Slice the data to shortest chunks where neighbor chunks have opposite slopes (calculated by linear model)
3. Merge the chunks until they have at least min_length_limit to the atomic structures
4. Iterative merge neighbor chunks which are less than max_length_expectation if the distance (E.g DTW distance) is smaller than merging_threshold and within merging_dynamic_constraint until nothing changes

The modeling process is two-stage: we first find the atomic patterns, then merge the atomic structures with neighbors into clusters. We start with the most explainable formulation of market dynamics, using the slope to find patterns. Patterns are then clustered into longer sequences by the dynamic time warping (DTW) method with constraints, the detailed algorithm is introduced in Algorithm 1.

Table 1: Hyper-parameters of market dynamics labeling

Name	Description
filter_strength	The strength of the low-pass Butterworth filter, the bigger the lower cutoff frequency, "1" have the cutoff frequency of min_length_limit period
slope_interval	The low, high slope if labeling_method=slope, setting to [1,-1] will enable auto zooming the slope interval to the range of slopes of current chunks, must set to [a, a] if dynamics number is 2
dynamic_number	The number of dynamics to be modeled
max_length_expectation	Chunk longer than this number will not merge actively
key_indicator	The column name of the feature in the data that will be used for dynamic modeling
timestamp	The column name of the feature in the data that is the timestamp
tic	The column name of the feature in the data that marks the tic
labeling_method	The method that is used for dynamic labeling 'slope': use the slope of each linear regression model of the chunks to label them, can set the slope threshold by 'slope_interval' or use an auto-zooming mode. 'quantile': use quantile of slopes of each chunk to label them, can keep chunks number in each dynamic balanced. 'DTW': use DTW clustering on the one-step return of the key_indicator to label the chunks.
min_length_limit	Every chunk will have at least this length, any pattern that have a smaller length than this is an inherent market micro-structure that can't be sliced
merging_metric	The method that is used for neighbor chunks merging 'DTW_distance': A normalized Dynamic Time Wrapping distance between two chunks
merging_threshold	The metric threshold that is used to decide whether a chunk will be merged with neighbor
merging_dynamic_constraint	Neighbor chunks of dynamics span greater than this number will not be merged(setting this to -1 will disable the constraint)

Due to the criteria varying with tasks and people, we set several hyper-parameters to control the market dynamic modeling, the detailed parameters are introduced in Table 1. The data of different markets and tasks are heterogeneous and require hyper-parameters tuning, we provide detailed instructions in this document[1].

2.1.3 Use Case

We have done an experiment on the second level Limit Order Book data of BTCUSDT processed from an example dataset of Tardis.dev[2].

Table 2: Experiment setting

Name	Description
filter_strength	1
dynamic_number	5
max_length_expectation	3600
min_length_limit	60
merging_metric	DTW_distance
merging_threshold	0.0003
labeling_method	Quantile
merging_dynamic_constraint	1

Experiment setup in Table 1 means that the data will be classified into 5 dynamics where each dynamic has the same number of chunks. The way the dynamics are given to the chunks is based on the slope of its linear regression. For example, the dynamics '0' contains the chunks that have a slope of the last 20% in all the chunks. For each chunk, it will have a length of at least 60 ticks, any micro-structure that has a length less than 60 is considered reasonable to be connected to a larger sequence. For example, we can have a 20 ticks 'bear' market in a 600 ticks 'bull' market. We will use a normalized DTW_distance to decide whether chunks can be merged by a threshold of 0.0003 and we will label the current chunks by their slope before they merge in every round, the neighbor that has a dynamic span larger than 1 will not be merged.

The five dynamics in Fig 2, from '0' to '4' are 'Bear', 'Less bear', 'Volatile', 'Less bull', and 'Bull'.

In addition to visualizations, we also proposed quantitative metrics to evaluate if the modeling is reasonable, as shown in Fig 3. By calculating these metrics, we want to see if there are chunks of different 'dynamics' merged and categorized into the wrong dynamic. As the slope is a key indicator of dynamics in this model so we use the max drawdown(mdd) and the max uptrend (the corresponding mdd for uptrend) to evaluate if this phenomenon exists. We can see that in 'Bear' markets, the max uptrend length is less than 60 while in the 'Bull' markets the max drawdown length is less than 60. At the same time, for the more extreme the bear/bull dynamic, their average max drawdown/ max uptrend slopes are larger. At the same time, long and small slope 'Volatile' chunks fall into dynamic '2'.

3 Usage

3.1 TradeMaster Library

All the evaluation tools are being integrated into the evaluation module of TradeMaster Library[3]. For now we only release the Market Dynamics Model, ongoing projects are listed in Section 4.

[1] https://github.com/TradeMaster-NTU/TradeMaster/blob/1.0.0/docs/source/tool/EvaluationSandbox_MDM.md
[2] https://datasets.tardis.dev/v1/binance-futures/book_snapshot_5/2020/09/01/BTCUSDT.csv.gz
[3] https://github.com/TradeMaster-NTU/TradeMaster/tree/1.0.0/trademaster/evaluation

Figure 2: Visualization of Market Dynamics Modeling

3.2 Website

In addition to the TradeMaster library, we also provide demos and sandbox api on our TradeMaster website[4]

[4]Website: http://trademaster.ai/

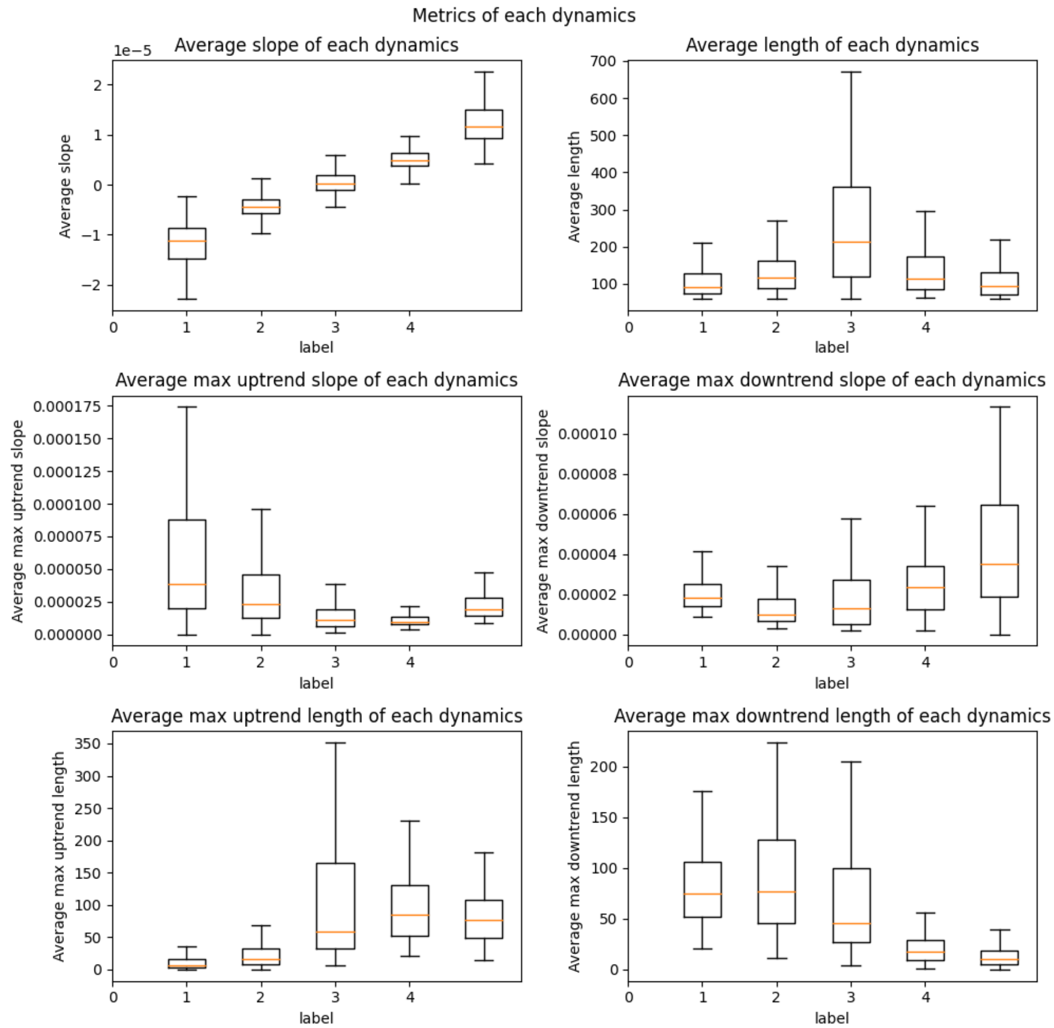

Figure 3: Quantitative metrics of market dynamics modeling

4 On-going Projects and Future Plans

4.1 Market Feature Simulator

The market feature simulator uses a generative adversarial network to generate mid and low-frequency market features, for example, Open, High, Low, and Close. The generated data is controlled by given conditions, for example, market dynamics and stock tics. With this market feature simulator, we can evaluate our QT algorithms beyond historical data on specific market dynamics. This would be helpful to predict our algorithms' performance under extreme markets and avoid uncontrollable risks. Furthermore, the generated data can be used in the training phase for more robust live-market trading performance, this would enhance the generalization ability of our algorithms. Currently, we are able to generate mid and low-frequency market features with controlled market dynamics and stock ticks.

4.2 Limit Order Book Simulator

The Limit-Order-Book Simulator uses an agent-based model to simulate every activity traders take in the financial markets, including the limit order book and the trades. This simulator is an interactive paper-trading environment for RL-based QT agents to train and test in. Our simulation is a hybridization of traditional financial methods and data-driven machine learning methods. Beyond empirical agents, we incorporate learning agents in the simulator to reconstruct market behaviors based on historical data. Currently, the market simulator shows close-to-real market stylized facts. We are working on the reproduction of the market micro-structures such as the order book liquidity.