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 that introduces diverse evaluation methods into the life cycle. The TradeMaster Sandbox evaluates QT algorithms in various paper-trading scenarios. At the end of the day, 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

Market dynamics is a long study and not a well-established market feature. The market dynamics are formulated differently according to specific tasks, it can be a factor crafted manually or learned by a Hidden Markov Model (HMM). For our TradeMaster evaluation sandbox, we want to evaluate policy performance on specific market dynamics labeled automatically. We start with the most explainable formulation of market dynamics. What's more, we also understand that criteria vary with tasks and people, so we set several hyper-parameters to control the market dynamic labeling process.

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 when labeling_method=slope
dynamic_number	The number of dynamics to be modeled
max_length_expectation	Slice 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
min_length_limit	Every slice will have at least this length
merging_metric	The method that is used for slice merging
merging_threshold	The metric threshold that is used to decide whether a slice will be merged
merging_dynamic_constraint	Neighbor segment of dynamics spans greater than this number will not be merged(setting this to -1 will disable the constraint)

Algorithm 1 Market Dynamics Modeling

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

The data of different markets and tasks are heterogeneous and require hyper-parameters tuning, we provide detailed instructions in this document[1].

3 Usage

3.1 TradeMaster Library

All the evaluation tools are be integrated into the evaluation module of TradeMaster Library[2]. For now we only release the Market Dynamics Model, ongoing projects are listed in Section 4.

3.2 Website

In addition to the TradeMaster library, we also provide demos and sandbox api on our TradeMaster website[3]

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.

[1] https://github.com/TradeMaster-NTU/TradeMaster/blob/1.0.0/docs/source/tool/EvaluationSandbox_MDM.md
[2] https://github.com/TradeMaster-NTU/TradeMaster/tree/1.0.0/trademaster/evaluation
[3] Website: http://trademaster.ai/