Short Description for Our Simulator

Chuheng Zhang

1 Overview

We simulate the market based on replaying the historical limit-order-book-level data and simulating the order matching mechanism (cf. Figure 1). Currently, we simulate at minute-level (i.e., one time step = one minute) which can be altered later. The state is a stack of market indicators and market snapshots over the past several time steps. The raw action is the order placement and we support market orders and limit orders. We also provide several wrappers to accept canonical discrete or continuous actions. From may point of view, the reward can be configured by the contestant with the aim to generate policies that can optimize the pre-specified metrics. We consider the following factors in our simulator: 1) temporary market impact; 2) order delay. We do not consider the following factors in our simulator: 1) permanent market impact of limit orders; 2) non-resiliency limit order book (see discussion).

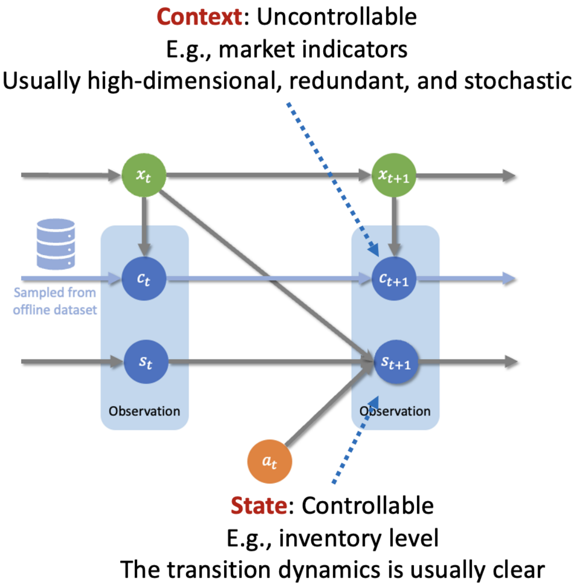
2 Dataset

Compared with the simulators based on the preset stochastic process [1] or a collection of preset interactive agents [2], the simulator driven by real market data can capture the complex market more accurately [3]. Moreover, compared with the previous work where the simulator is based on bar-level data, our simulator relies on the LOB-level data of the market which records an LOB snapshot every 3 seconds. With finer-grained data, we are able to learn more practical trading agents. For example, we can evaluate how an agent that trades using only MOs suffers from a large trading cost, which is the scheme adopted in many existing papers [see e.g., 4]. The interested time period of trade execution tasks in the industry is typically from 10 to 120 minutes, which is configurable in our simulator. Our simulator is based on the dataset that records an LOB snapshot every 3 seconds from the real market. The time period of trade execution tasks in our experiments is set to 30 minutes. To avoid a long planning horizon, the agent interacts with the simulator at a lower frequency (i.e., one minute per step). Nevertheless, our simulation is equivalent to being carried out snapshot by snapshot for higher accuracy.

3 Observations/States

The observation in our simulator consists of the private variable (i.e., the state) and the market variable (i.e., the context). The private variable consists of the remaining time and executed quantity. (This is for the trade execution task. If we want to simulate for the trading task, we the private variable may be the current portfolio of the trader.) The market variable can be the stacked features (including order-book-related features, technical indicators, raw snapshots, etc.) over several past steps. Our simulator implements a wide range of features including the features that appear in the previous papers as far as we know. For different designs on the observations space, the agent can choose from these features. To eliminate the differences in the features on different stocks, the simulator normalizes the features as follows: The price (or the feature whose dimension is price) is normalized using z-score with the open price on that trading day as the mean and the volatility on the previous trading day as the standard deviation. The volume (or the feature whose dimension is volume) is normalized by dividing by the total volume of the last trading day. In specific algorithms, we may perform another normalization on these features to fit them into a proper value range.

Preprint. Under review.


Context: Uncontrollable
E.g., market indicators
Usually high-dimensional, redundant, and stochastic

State: Controllable
E.g., inventory level
The transition dynamics is usually clear

Figure 1: Diagram of our simulator.

4 Actions

On each time step, our simulator receives a list of orders, each of which can be an MO or LO that specifies the direction, the quantity, and the price (only for LO). On top of this, we provide a series of wrappers to fit different designs on the action space (e.g., discrete/continuous/combinatorial action spaces). Our simulator will provide the best possible execution for each order. For example, if the quoted price is lower than some bid price level, the simulator will automatically place an MO to fill the outstanding bid orders whose prices are higher than the quoted price, and an LO for the remaining quantity. In our previous benchmark algorithms, the agent places an order on each step by choosing a quoted volume and a quoted price from discretized sets. The quoted volume is selected from $\{\frac{1}{2}\text{TWAP}, \text{TWAP}, \frac{3}{2}\text{TWAP}, 2\text{TWAP}\}$ where TWAP is the volume executed on each step by a TWAP strategy (i.e., selling an equal amount on each step). The quoted price is specified by a price difference w.r.t. the best ask price. If the quoted price is lower than the best bid price, the agent actually places an MO; otherwise, it is an LO. Outstanding orders at the end of each step will be withdrawn.

5 Reward

The reward function may consist of a basic revenue term (e.g., negative trading cost or average execution price for trade execution and the PnL for trading) and several regularization terms (e.g., approximating the permanent market impact or enforcing a TWAP-like strategy in trade execution). The revenue term reflects the overall objective of trade execution that minimizes the trading cost or the average execution price for a sell program. Moreover, various of regularizers are adopted to model the permanent market impact or the prior knowledge of a good execution program (e.g., enforcing a TWAP-like program). Our simulator provides various choices on the reward function design and can benchmark different designs with a uniform set of metrics (such as the trading cost or implementation shortfall [5]). In our algorithms, the reward function consists of a revenue term r_1 and a regularization term r_2, i.e., $r_t = r_1 + \beta r_2$ where β is a coefficient. The revenue term is $r_1 = n_t \bar{p}_t$ where n_t is the executed volume in the last step, and $\bar{p}_t$ is the corresponding average execution price. The regularization term is $r_2 = (v_t - v_{t,\text{TWAP}})^2$ where v_t is the remaining inventory, and $v_{t,\text{TWAP}}$ is the remaining inventory if we follow the TWAP strategy. We can let the contestant to configure the reward function and evaluate the performance using a set of metrics such as the profit, the risk-adjusted profit, the maximum drawdown, and the correlation with others strategy/classical strategies.

6 Transition Dynamics

Given a list of orders on the t-th time step, our simulator will determine the reward and the state on the next time step. For MOs, we consider the temporary market impact and the time delay. For example, when the decision of the agent is based on observation generated on time τ, the execution of an MO is based on the snapshot on time $\tau + \Delta\tau$ where $\Delta\tau$ is a preset time delay. Previously, we use $\Delta\tau = 3s$ and $\Delta\tau = 100ms$. For LOs, we determine whether the order can be executed snapshot by snapshot till the $(t + 1)$-th time step. If the highest market price (i.e., a transaction occurs on this price) in one snapshot exceeds the price quoted in the LO, we consider the order is fully executed. If the highest market price exactly equals the quoted price, the order may be partially filled and the ratio is calculated by reconstructing the transactions between snapshots. However, considering that 1) the quantity may be too large to be fully executed or 2) the LO may be at the end of the queue of the quoted price level, we impose additional trading limits on the above matching mechanism to encourage conservative simulation.

7 Discussion.

To improve fidelity, our simulator considers the temporary market impact of MOs, the time delay, and determines the execution of LOs based on reconstructing the transactions between snapshots. However, there are still components that we do not consider. First, the permanent market impact is the change of the equilibrium price during at least our planning horizon when we place an order. Here, we assume the permanent market impact is linear w.r.t. the order quantity and therefore considering this factor does not change the optimal solution of our strategy [see 6]. Then, MOs in our simulation not only change the current LOB but also the LOB of the next time step, possibly resulting in degenerated fidelity. Therefore, we also rely on the assumption that the limit orders are resilient within a short period of time (which should be smaller or comparable to the time interval between two simulation steps). Fortunately, this is verified by empirical studies such as Degryse et al. [7]; Cummings and Frino [8]; Gomber et al. [9].

References

[1] Kevin Dabérius, Elvin Granat, and Patrik Karlsson. Deep execution-value and policy based reinforcement learning for trading and beating market benchmarks. *Available at SSRN 3374766*, 2019.

[2] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. ABIDES: Towards high-fidelity market simulation for AI research. *arXiv preprint arXiv:1904.12066*, 2019.

[3] Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Balch. Get real: Realism metrics for robust limit order book market simulations. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

[4] Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou, Weinan Zhang, Jiang Bian, Yong Yu, and Tie-Yan Liu. Universal trading for order execution with oracle policy distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 107–115, 2021.

[5] A. F. Perold. The implementation shortfall: Paper vs. reality. *Journal of Portfolio Management*, 14(3):4–9, 1988.

[6] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3: 5–40, 2001.

[7] Hans Degryse, Frank De Jong, Maarten Van Ravenswaaij, and Gunther Wuyts. Aggressive orders and the resiliency of a limit order market. *Review of Finance*, 9(2):201–242, 2005.

[8] James Richard Cummings and Alex Frino. Further analysis of the speed of response to large trades in interest rate futures. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 30(8):705–724, 2010.

[9] Peter Gomber, Uwe Schweickert, and Erik Theissen. Liquidity dynamics in an electronic open limit order book: An event study approach. *European Financial Management*, 21(1):52–78, 2015.