Deep Reinforcement Learning for Market Making

Extended Abstract

Pankaj Kumar
Copenhagen Business School, Denmark
pk.mpp@cbs.dk

ABSTRACT

Market Making is high frequency trading strategy in which an agent provides liquidity simultaneously quoting a bid price and an ask price on an asset. Market Makers reaps profits in the form of the spread between the quoted price placed on the buy and sell prices. Due to complexity in inventory risk, counterparties to trades and information asymmetry, understanding of market making algorithms is relatively unexplored by academicians across disciplines. In this paper, we develop realistic simulations of limit order markets and use it to design a market making agent using Deep Recurrent Q-Networks. Our approach outperforms a prominent benchmark strategy from literature, which uses temporal-difference reinforcement learning to design market maker agents. The agents successfully reproduce stylized facts in historical trade data from each simulation.

KEYWORDS

Deep Reinforcement Learning; Market Making; Limit Order Books; High Frequecy Tading Stratergies; Agent Based Models.

ACM Reference Format:
Pankaj Kumar. 2020. Deep Reinforcement Learning for Market Making. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

The electronification of securities trading has transformed traditional human-driven markets into predominantly automated, where high frequency trading (HFT) typically exceeds 80% of total volume traded in U.S listed equities [12, 13]. HFT is a form of automated trading in which security positions are turned over very quickly by leveraging advanced technology and the associated extremely low latency rates [14]. Market Making is HFT based strategies contributing to market liquidity by matching buyer and seller orders. The profit is earned as the spread between the quoted price placed on the buy and sell prices. With every-growing minuscule limit order book (LOB) data, complexity in inventory risk, counterparties to trades and information asymmetry, understating of market making algorithms is relatively shallow [2, 3, 20]. This paper uses a variant of Deep Recurrent Q-Networks (DRQN) to design market making agents interacting with realistic limit order book simulation framework.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

1.1 Related Work

A number of market making strategies have been proposed across disciple, including finance [3, 6], econophyics [12] and machine learning[2, 4, 20]. Earlier work in finance considers maker making as a problem of stochastic optimal control, where order book dynamics are designed using control algorithms after developing the arrival and execution model [3, 5] to understand the price impact, adverse selection, risk measures, and inventory constraints.

Another prominent approach, agent based model (ABM), ranging from zero intelligence to intelligent variants are used to study market making, but are typically evaluated in simulated markets without using real market data. It gives the modeler flexibility to churn out potentially emergent phenomenon as a result of interaction between agents. With evolving technology-based disruption in HFT, the existing learning models and empirical models are deficient and may no longer be appropriate. Reinforcement learning (RL) has been applied for market making [20], algorithmic trading [22], optimal execution [16], and foreign exchange trading [8]. However, defining hand-crafting features in reinforcement learning for agents to learn while interacting within a dynamic environment is a major throttle block. Also, RL could be slow to learn in large state spaces and the methods did not generalize (across the state space).

Deep learning eliminates the need for manual feature design, thus finding compact representations in high-dimensional data. It also helps to generalize across states improving the sample efficiency for large state-space RL problems. Augmenting deep learning with reinforcement learning, deep reinforcement learning (DRL), enables RL to scale to problems with high-dimensional state and action spaces. The outstanding success stories of DeepMind's, kick-starting with superhuman level performance in Atari 2600 video games [15], AlphaGo [19], and AlphaStar [21] proves the effectiveness of DRL. However, only a few works is featured optimal execution [17], market making [9], and high frequency trading [22] as compared to the games.

The success of such single DRL's can be accredited to the use of experience replay memories, which legitimate Deep Q-Networks (DQNs) to be trained efficiently through sampling stored state transitions. However, despite the ever-increasing performance on popular benchmarks such as Atari 2600 games, DQN struggle to generalize when evaluated in different environments. It does not perform well in partially observable domains [11], overestimate action values under certain conditions [10], and not efficient when experience replay needs to be prioritized [18]. Deep Recurrent Q-Networks (DRQN) [11] proposed using recurrent neural networks, in particular, LSTMs (Long Short-Term Memory) solves the above problem by replacing the first post-convolutional fully connected layer with an LSTM layer in DQN setting. With this incorporation, DRQN has memory capacity so that it can even work with only one input rather than a stacked input of consecutive frames. Double DQN [10] obliterate the overestimation problem in DQN, resulting in more stable and reliable learning. By prioritizing experience, authors [18] achieved a new state of art human-level performance across benchmark Atari games.

1.2 Contributions

The main contribution of this paper is to develop realistic simulations of limit order markets and use it to design a market making agent using DRQN. The simulation framework takes account of the agent's latency and have build-up maker/taker fees as defined in NYSE. We modify the classical DQRN architecture and incorporate double Q-learning and prioritized experience to take account of volatile, illiquid and stagnant markets. Our approach outperforms a prominent benchmark strategy from literature, which uses temporal-difference reinforcement learning to design market maker agents.

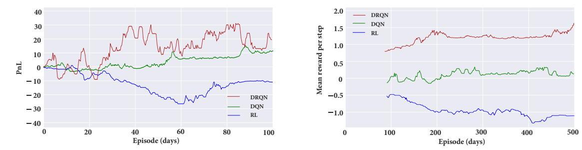
2 EXPERIMENTS AND RESULTS

We run the model for 1000 iterations to find relevant hyper-parameter using random search. After that, we train the models for some ten million time steps for intervals of 10000, which is equivalent to 500 trading days to collect data, monitor and visualize the learning of the agent. Then, testing the environment on the benchmark to see the agent's learning pattern. We use a spread-based benchmark strategy proposed by [20], which uses temporal-difference reinforcement learning to design market making agents. All the analysis was done using single market making agents with multiple market-takers.

To evaluate the performance of agents, we use profit and loss with exponential transaction cost and maker-taker fee (PnL) computed for each hour. The trading strategy's efficiency to capture the spread is evaluated by normalized daily PnL (NPnL) [20]. We also use the mean absolute position (MAP) to capture the important characteristic of market makers where agents avoids large inventories [20]. We report the NPnL and MAP with the standard deviation and mean respectively.

2.1 Results and Analysis

The performance of the agents is compared in Figure 1. In-spite of handcrafted strategy, where actions with various quantities are taken at different states, the RL agent performs badly and not stable as compared to DRQN and DQN agents. It is to be noticed that the trading strategy which RL agents follow doesn't take account of order size, cancellation, adverse selection, transaction cost and volatility, which the current simulator introduces while interaction. Adding to the same, the order matching is subject to market-takers, who trades on market trends as described in agent's trading strategies. DQN performance is stable , but fails to outperform the DRQN. The reasoning may be linked to not efficient state representation, overestimated action values, partial observability and pritorized experience, which DRQN incorporates. To understand the performance better, we need to action selection with respect to limit order book dynamics, which we plan to do next.



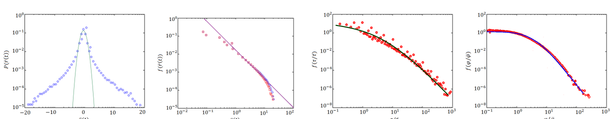
(a) Reward. **(b) Mean Reward.**

Figure 1: Trading agent performace.

2.2 Validation

In agent-based models of financial markets, it is standard practice to measure the validity of the model by investigating whether the order-book data have particular characteristics, known as the "stylized facts" [1]. We present some of the stylized facts reproduced from historical trade data.

To reproduce stylized facts concerning price, we first calculate return, which is given by $r(t) = \log(p_t) - \log(p_t)$. The heavy tails (HT) in the distribution of returns is depicted in Figure 2a. The normalized return distribution has a fatter tail than green Gaussian distribution. Furthermore, the cumulative distributions function [1, 7], shown as the blue (positive tail) and red (negative tails) in Figure 2b, exhibits power law (PL). The violet line is the asymptotic power-law function with tail exponent 4.



(a) HT. **(b) PL.** **(c) LOS.** **(d) LOC.**

Figure 2: Stylized facts.

We now switch from price to order size. The Figure 2c illustrates the probability density distribution (PDF) $f(\tau/\bar{\tau})$ of limit order size (LOS) τ, where $\bar{\tau}$ is mean order size of individual stock. The green line is Gamma distribution fit to the normalized order size. It is evident from the figure that the Gama distribution fits remarkably good to empirical PDF. This is in line with the existing literature [1], confirming the existence of heavy tail in limit order size. The limit order cancellation (LOS) also follows Gama distribution which can be seen in Figure 2d. The fitting procedure is the same as the limit order size.

3 CONCLUSIONS

In this paper, we have designed a market making agent using deep recurrent Q-network that outperforms a prominent benchmark strategy, which uses temporal-difference reinforcement learning. The market making agents interact with highly realistic simulation of the limit order book, which till now is non-existence in the academic research. The suitable modification in the exciting DRQN network architecture [11] and training procedure allowed our agents to yield predominant performance. The future extension of this work would be to incorporate order book data with deep reinforcement learning, and extend it to a multi-agent setting, where all agents learn and trade simultaneously.

REFERENCES

[1] Frederic Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. 2016. *Limit Order Books.* Cambridge University Press.

[2] Jacob Abernethy and Satyen Kale. 2013. Adaptive Market Making via Online Learning. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2058–2066.

[3] Marco Avellaneda and Sasha Stoikov. 2008. High-frequency trading in a limit order book. *Quantitative Finance* 8, 3 (2008), 217–224.

[4] Aseem Brahma, Mithun Chakraborty, Sanmay Das, Allen Lavoie, and Malik Magdon-Ismail. 2012. A Bayesian Market Maker. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12).* 215–232.

[5] Alvaro Cartea, Sebastian Jaimungal, and Jason Ricci. 2011. Buy Low Sell High: A High Frequency Trading Perspective. *SIAM Journal on Financial Mathematics* 5 (11 2011).

[6] Tanmoy Chakraborty and Michael Kearns. 2011. Market Making and Mean Reversion. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC '11).* 307–314.

[7] R. Cont. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* 1, 2 (2001), 223–236.

[8] Michael Dempster and V. Leemans. 2006. An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications* 30 (04 2006), 543–552.

[9] Marcus Elwin. 2019. *Simulating market maker behavior using Deep Reinforcement Learning to understand market microstructure.* Master's thesis. KTH Royal Institute of Technology, Stockholm.

[10] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16).* AAAI Press, 2094–2100.

[11] Matthew Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15).*

[12] Frank McGroarty, Ash Booth, Enrico Gerding, and V. L. Raju Chinthalapati. 2018. High frequency trading strategies, market fragility and price spikes: an agent based model perspective. *Annals of Operations Research* (Aug 2018).

[13] Albert Menkveld. 2013. High frequency trading and the new market makers. *Journal of Financial Markets* 16, 4 (2013), 712–740.

[14] Albert Menkveld. 2016. The Economics of High-Frequency Trading: Taking Stock. *Annual Review of Financial Economics* 8, 1 (2016), 1–24.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (Feb. 2015), 529–533.

[16] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. 2006. Reinforcement Learning for Optimized Trade Execution. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06).* 673–680.

[17] Brian Ning, Franco Ho Ting Ling, and Sebastian Jaimungal. 2019. Double Deep Q-Learning for Optimal Execution. *ArXiv* abs/1812.06600 (2019).

[18] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. In *International Conference on Learning Representations.* Puerto Rico.

[19] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* 529, 7587 (Jan. 2016), 484–489.

[20] Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. Market Making via Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18).* Richland, SC, 434–442.

[21] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M. Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/. (2019).

[22] Haoran Wei, Yuanbo Wang, Lidia Mangu, and Keith Decker. 2019. Model-based Reinforcement Learning for Predictions and Control for Limit Order Books. *ArXiv* arXiv:1910.03743 (2019).