# COMP 579 Final Project

Tianhao Xie (260954275); Jerry Zhu (260205780)

May 2023

## 1 Abstract

This course project consisted of comparing Reinforcement Learning (RL) methods learned in class with more traditional algorithmic approaches in the context of Market Making (MM): the process of simultaneously and continuously providing buy and sell prices in a financial asset. To serve its purpose as a simplified introduction into the field, a Simple Probabilistic Model (SPM) of a Limit Order Book (LOB) was used to compare analytical and RL strategies, later discussed below. Role-wise, Tianhao was responsible for implementing the environment and algorithms, and Jerry was responsible for MM experiment design, testing, and project report.

## 2 Introduction

The primary motivation behind this project was the mathematical similarities between MM and RL. The core objectives of a market maker are to (1) maximize profits and (2) limit the risk of holding too large of an inventory, which may lead to large losses. Thus, maximizing a risk-adjusted return measure is necessary in optimal MM. This lends itself well to RL, which involves an agent learning purely by its interaction in a stochastic environment. Indeed, despite its novelty, many academics agree that RL approaches to optimal MM are superior to standard strategies. As such, through this experiment we will attempt to answer the question: are reinforcement learning methods able to outperform analytically derived strategies?

## 3 Background

It is important to first define how exactly MMs operate as well as the derivation of the traditionally optimal strategies before we proceed later with our experiment. As mentioned above, MM is the process of simultaneously and continuously providing buy (bid) and sell (ask) prices in a financial asset in a market. The difference between their bid and ask, called a spread, determines their profit for each asset that is bought and subsequently sold.

In the process of market making, these bid and ask prices not only have to be set and continuously updated, but also must be optimized to maximize profit and minimize inventory risk. For instance, setting "good" prices would mean more trades but a lower spread and a higher inventory risk, and vice-versa. All of this relies on the Limit Order Book (LOB), which aggregates and displays all offers from market participants to buy or sell the financial asset at a maximum or minimum price, called a limit order LO. When participants accept to buy, they place a Market Order (MO), which executes against the LO at the best available price, and when they refuse, they place a Cancellation Order (CO), which cancels their outstanding CO. The participants rely on the displayed MM's bid and ask prices to execute such decisions.

## 4 Methodology

In this experiment, we used Tabular Q-Learning.

Following directly from our definition of a MM's role and objectives, let us respectively define our agent's state space, action space, and reward function. First, the state space is represented by two variables: time and inventory. More specifically, we have $(t, Q_t) \in S = \{0, 1, \cdots, T\} \times \{\underline{q}. \cdots, \bar{q}\}$ where we choose $T = 5$ and $\underline{q} = \bar{q} = 3$. For the action space, there are two variables: bid depth and ask depth. Both are represented by the number of ticks away from the mid-price that the bid and ask orders should be posted. Finally, the reward function is $R_t = \delta V_t - \phi {Q_t}^2$, where $V_t$ is the value process, defined as $V_t = X_t + H_t$, $X_t$ is MM cash value at t, $H_t$ is MM inventory value at $t \to Q_T(S_T - \alpha Q_t)$, $S_T$ is mid price process at $t$, $Q_T$ is inventory process at $t$, and $\alpha$ is fees of taking liquidity. Roughly, the first term represents maximization of profits, and the second term represents minimization of inventory risk. (More to be explained below.)

To simulate the LOB, we reproduced a simplified model presented by Cartea et al. in Algorithmic and high-frequency trading using the Gym environment API, which we refer to as the Simple Probabilistic Model (SPM). This simplification is done through two core assumptions. First, it is to note that this model only accounts for intraday evolution of the LOB, meaning the sequence terminates at the end of the day. Indeed, we assume here that at the end of the day, MM liquidates the entirety of his inventory, as defined by $H_t = Q_T(S_T - \alpha Q_t)$ in the reward function. Second, the environment accounts for one single MM, as well as an aggregated LOB through the batching of sell and buy LOs by time steps. At each time step, it only assumes that there is one MM who puts up a single sell LO and a single buy LO. As for the SPM, rather than directly modeling the LOB, it instead simulates the mid-price dynamics (Brownian motion), the arrival of market orders (Poisson processes), and the conditional probability of an arriving market order to fill a limit order. In real life, the environment would clearly be more granular.

The SPM follows the following **step(action a)** algorithm:

1. Update mid price
2. If current time $t$ is lower than parameter $T$ then
3. Set new bid and ask price according to action a
4. Simulate number of MO arrivals
5. Simulate number of MO executions
6. Execute the orders
7. Else, if current time $t$ is equal to parameter $T$ then
8. Liquidate MM position
9. Calculate reward $R_t = t + dt$

We designed our experiment protocol based on Carlson and Regnell's seminal RL for MM thesis paper, which instead specifically explores Double Deep Q-Networks in a Markov Chain environment. Here, our agent is trained for 10,000,000 episodes, ten separate times. Exploring starts for the starting volume is used to find the best strategy for t = 0 and increase the chances of visiting states with high absolute volumes early in the episodes. An $\epsilon$-greedy policy is used to aid convergence to the optimal strategy. The exploration rates of both these methods are decreased linearly during training, whereas the learning rate is decreased exponentially. The discount rate was set to $\gamma = 1$ since the episodes are of finite length.

An exponentially decaying learning rate was used since it was observed that a higher learning rate only was needed for a short period of time for the agent to get close to the optimal action-values. The initial Q-matrix was set to the null matrix.

1. Exploring, 1, 50, 0.05, linear

2. E-greedy, 1, 50, 0.05, linear

3. Learning rate, 0.5, 100, $5 \times 10^{-6}$, exponential

Finally, the analytical solution for optimal bid and ask depth against which our RL model will later be compared is taken directly from Chapter 10.2 of Cartea et al. The derivation is omitted due to page limit.

# 5 Results

For each of the four runs, we calculated the average return, the standard deviation, as well as the optimal action and its q-value. Below are the training plot and results:

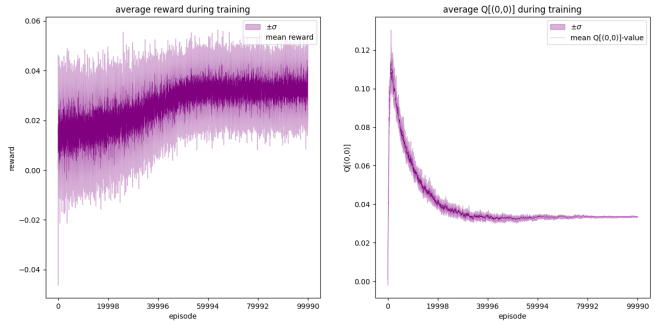


Figure 1: IMDB

| run | mean reward | std reward | opt action | q-value |
|---|---|---|---|---|
| run 1 | 0.0336902 | 0.0307304 | (1, 1) | 0.0336927 |
| run 3 | 0.0336504 | 0.0312427 | (1, 1) | 0.0330163 |
| run 4 | 0.0330592 | 0.0313556 | (1, 1) | 0.0333254 |
| run 5 | 0.033263 | 0.0307697 | (1, 1) | 0.0337307 |
| run 6 | 0.0329219 | 0.0306521 | (1, 1) | 0.0335001 |
| run 7 | 0.0331807 | 0.0312536 | (1, 1) | 0.0339156 |
| run 8 | 0.0340776 | 0.0318026 | (1, 1) | 0.0338797 |
| run 9 | 0.0337635 | 0.0306513 | (1, 1) | 0.0334388 |
| run 10 | 0.0337378 | 0.0309658 | (1, 1) | 0.0334437 |

Table 1: Different Runs Table

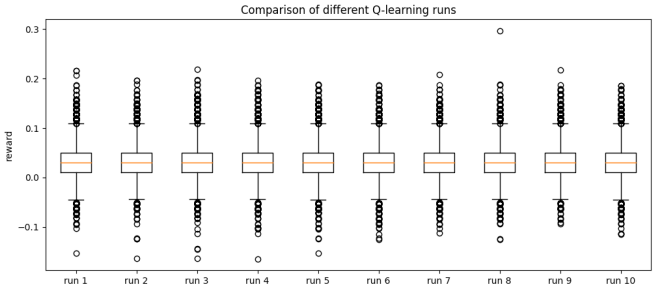


Figure 2: Different Runs Plot

The average reward clearly converges similarly at around 0.033.

We then compared this with various other strategies, in particular the analytically optimal discrete strategy (quotation depths rounded to closest tick), the analytically optimal continuous strategy (no rounding), a constant benchmark strategy (same quoted constant depth), as well as a random benchmark strategy.

| Strategy | Mean Reward | Std Reward |
|---|---|---|
| analytical discrete | 0.0338348 | 0.0311443 |
| analytical continuous | 0.0350523 | 0.0308443 |
| constant (d=2) | 0.0259898 | 0.0278544 |
| random | 0.0181398 | 0.0350977 |
| Q-learning (best run) | 0.0338141 | 0.0319561 |
| Q-learning (average) | 0.0342294 | 0.0316268 |

Table 2: Benchmarking Table

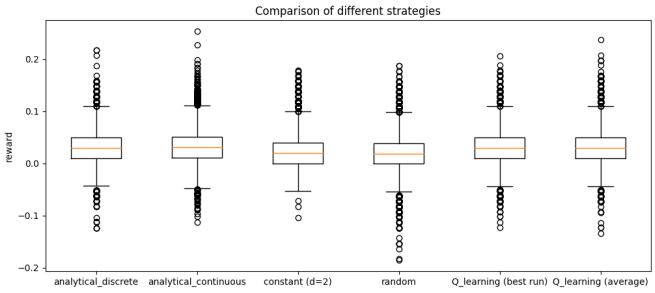


Figure 3: Benchmarking Plot

Unsurprisingly, the constant and the random strategies perform the worst.

The best single Q-learning strategy is beaten by the mean strategy, indicating some noise in the strategies being averaged out.

Generally speaking, we notice that Q-learning strategies and analytically optimal strategies perform similarly. More specifically, the averaged Q-learning strategy beats the analytical discrete strategy by a small margin, but is slightly outperformed by the analytical continuous strategy. Moreover, the standard deviation in Q-learning is slightly higher than the standard deviation of analytical strategies, indicating a slightly higher volatility in our agent.

We theorize two core explanations as to why analytical discrete strategy is sub-optimal in comparison to our best Q-Learning agent, the averaged one, and why overall the analytical continuous strategy does not outperform the latter by a lot. First, the LOB environment is discretized in terms of time. Secondly, the analytically optimal depths are discretized by rounding them to the closest tick. For instance, rounding a depth of 0.004 to 0 (at the mid price) instead of to 0.01 (one tick away from the mid price) means that the market maker will not earn any spread at all on that trade, albeit it may be used to reduce the penalty received from holding a large absolute volume.

There are also other observations worth mentioning. The optimal action is almost always 1 in depth for bid/ask, as can be easily observed in the heatmaps below (better visualized than line plots).

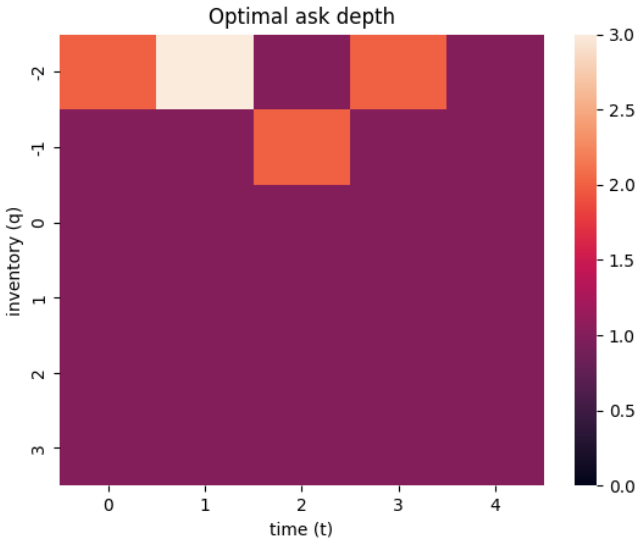


Figure 4: Optimal ask depth for a MM based on time (t) and inventory (q) (Q-learning)

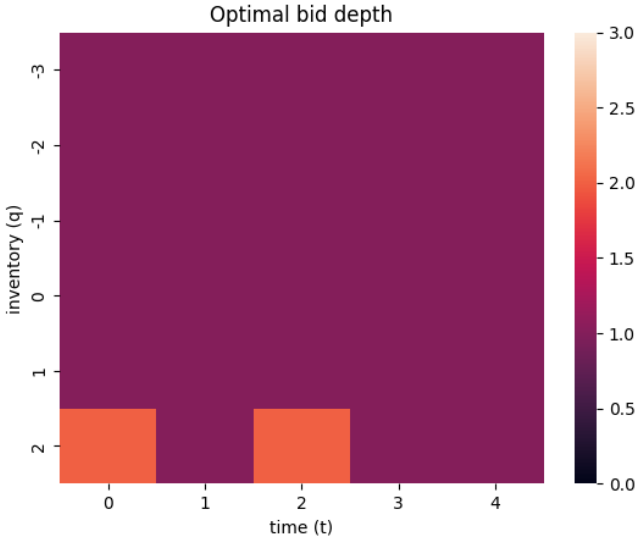


Figure 5: Optimal bid depth for a MM based on time (t) and inventory (q) (Q-learning)

Moreover, the difference between strategies are minimal, which may naturally be attributed to the short time period we have set for this experiment.

Finally, in order to visualize the stability of Q-learning strategies, we plotted a heatmap for the number of unique optimal actions per state space.

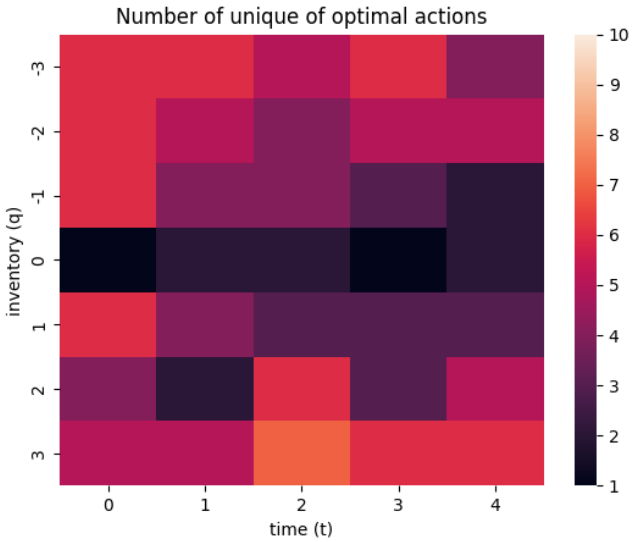


Figure 6: IMDB

Here, we can see that the heatmap shows great stability for the state $(t, Qt) = (0, 0)$ as well as states closer to the end of the episode with lower absolute volume. This is because fewer rewards are to be received as the episode approaches its end, speeding up convergence. The initial state being more stable is as the exploration rate is highest in the beginning. We also observe that the upper and lower portions of the heatmap have higher values: this is because the chance of a trade being executed on that side of the order book is very rare and therefore those states have not properly converged to a unique optimal action.

# 6 Conclusion and Future Work

Market Making, the process of simultaneously and continuously providing buy and sell prices in a financial asset, is complicated to optimize. While traditional attempts to optimize a MM's operations are based on stochastic differential equations, RL is a growing field that is also conceptually well suited for it, yet is still relatively new in this particular application. As such, the main goal of this project was to compare analytical and reinforcement learning strategies for MM. To do so, we used a simple probabilistic model of a limit order book to compare analytical and RL-derived strategies, and provided a baseline result that RL strategies are at least as good as analytically derived strategies. To build upon this preliminary result, we are considering reformulating MM as an infinite horizon task, and employing multi-agent algorithms to allow multiple MMs to compete against each other (market fragmentation) as a way to better simulate something that is closer to reality, and that may later be reliably used in industry.

# References

CR *S. Carlsson and A. Regnell (2022). "Reinforcement Learning for Market Making". KTH Royal Institute of Technology.*

Cartea, Á., Jaimungal, S. and Penalva, J. *Algorithmic and high-frequency trading* (Cambridge University Press, 2015)