# Online Learning in Limit Order Book Trade Execution

Yuanzhao Zhang, Henry Ip, Hugh Christensen
BMLL Technologies Ltd, 5 Fleet Place, London, UK, EC4M 7RD
yz551@alumni.cam.ac.uk, {henrymanden, hughchristensen}@bmlltech.com

## Introduction

Optimal execution of trades is a key problem in any investment activity. Once the decision has been made to sell a certain amount of shares the challenge often lies in how to optimally place this order in the market. More formally we define the goal as to sell a specific number of shares of a given stock during a fixed time period in a way that minimizes the accumulated cost.
The performance of online learning algorithm is tested on DAX30 data from Feb. 2018 to Jun. 2018 (6 months) and benchmarked against the Almgren-Chriss (AC) model.

## Reinforcement Learning

The system operates in rounds indexed by $\rho \in \{1, 2, ...\}$. Each round is composed of $L$ time slots, the set of time slots within each round is denoted by $\mathcal{L} := \{1, ...L\}$.

**States**:

- Private state: $I^l_\rho$, inventory level at time slot $l$ of round $\rho$.
- Market state: $M^l_\rho$, the amount of change in bid price at time slot $l$ in round $\rho$ from the bid price in the beginning of the round in units of tick size.

**Actions**:

- The amount of shares to be traded with a market order. $a^l_\rho$ is the action taken at time slot $l$ in round $\rho$.

**Transitions**:

- It is assumed that the market state evolves independently from the actions. $P(M, M')$ denotes the probability that the market state transitions from $M$ to $M'$.

**Cost**:

- Cost is defined as the implementation shortfall.

$$IS_\rho = (W_\rho p_r(\rho) - \sum_{l=1}^{L} a^l_\rho p_b(\rho, l))/(W_\rho p_r(\rho))$$

## The Almgren-Chriss Model

The AC model [1] gives a closed-form execution strategy which minimize trading costs over a fixed time horizon. It suggests that after execution $j$, the recommended amount of inventory remaining is given by:

$$x_j = \frac{\sinh(\kappa(T - t_j))}{\sinh(\kappa T)} X, j = 0, ..., N$$

## Optimal Policy

The optimal policy takes a simple form:

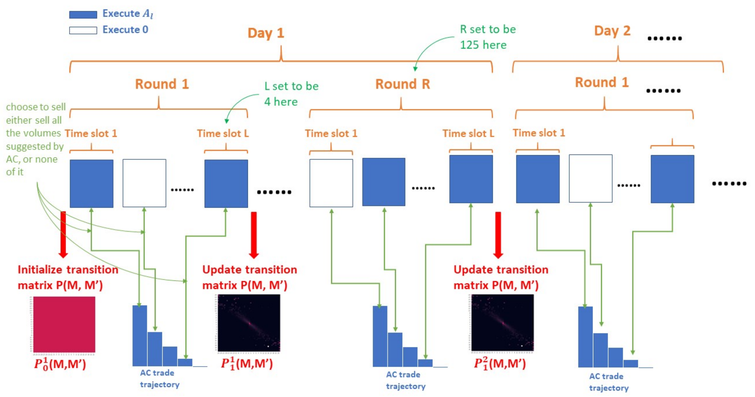
$$\pi^*_l = \begin{cases} 0 & \text{if } g_X(M_l) > E[g_X(M_L)|M_l] \\ A_l & \text{if } g_X(M_l) < E[g_X(M_L)|M_l] \end{cases}$$

which indicates that at each time slot, we choose to either sell all of the available limit at the current time slot or or save the shares up to the final time slot. It shows that the optimal action at each time slot depends on the current market state and the distribution of market state at the final time slot given the current state. In other words, if the expected market state in the final time slot is greater, we desire to wait and sell in the final time slot.
The optimal policy at each time slot is implemented using dynamic programming. In particular, we use the following equation to compute $E[g_X(M_L)|M_l]$ in a recurrent way:

$$E[g_X(M_L)|M_l] = \sum_{M_{l+1} \in \mathcal{M}} \hat{P}_\rho(M_l, M_{l+1}) E[g_X(M_L)|M_{l+1}]$$

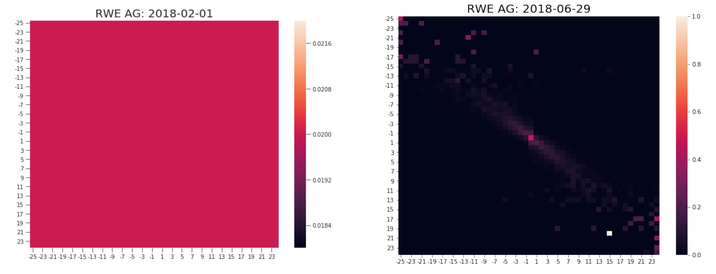
## Greedy Exploitation Algorithm: GLOBE

The online learning algorithm *Greedy exploitation in Limit Order Book Execution* (GLOBE) is a model-based approach where it is trying to learn the state transition probabilities of the market variables. Unlike the original version [2] where at the beginning of each day we need to re-initialize the transition matrix, here we use a modified version where the learned transition matrix is passed across different dates sequentially to achieve better learning result.




## Transition Matrix

A visualization of the transition matrix $\hat{P}(M, M')$ is shown below. The total number of market states is set to be 50 (from -25 to 25) according to historical data. As we can see, most of the transitions happen on the diagonal of the state space. The highest probabilities concentrates around the center $(0, 0)$ of the state space, indicating that most of the time, we do not see a big change in price within 1 minute. The left image shows the initial values of the matrix, and the right image shows the learned matrix after 6 months.




## Results & Analysis

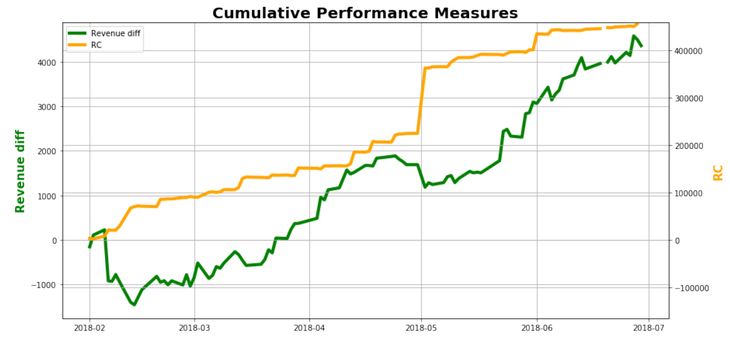
We run the experiments for DAX30 securities using Apache Spark on AWS clusters. We set the number of rounds to be 125, which corresponds to a 1-minute time interval for a time slot. The performance is evaluated by the Relative averaged Cost per round (RC):

$$RC_R(alg) = \frac{ACPR_R(AC) - ACPR_R(alg)}{|ACPR_R(AC)|} \times 100$$

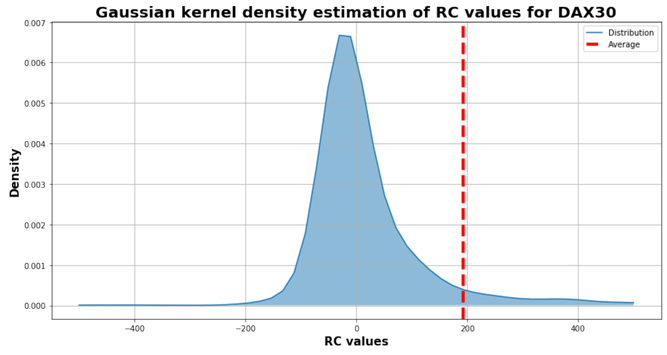
as well as the actual revenue difference we can get by selling stocks:

$$Revenue\_diff = Revenue_{AC} - Revenue_{GLOBE}$$

The target volumes to be liquidated during each round on a day is drawn uniformly at random from $[10, 100]$. A Positive RC value indicates that GLOBE outperforms AC model. In principle, the greater the value, the better the performance.
The cumulative sum of the difference (AC-GLOBE) in actual revenue received by running two models and RC are plotted in the same graph shown below. The results are aggregate for all DAX30 securities.




Another way of visualizing the results is by using the kernel density estimation. Here we are using Gaussian kernel. The red vertical line indicates the average value, which is positive.




## Conclusion & Future Work & References

The experiments show the effectiveness of the online learning algorithm by showing a positive cumulative sum of RC values of 6 months of data. However, we do see a negative cumulative sum of revenue difference, indicating that we receive less revenue by using the online learning algorithm compared with AC. Thus RC should not be the only criteria of measuring model performance. It is also recommended to learn a transition matrix across a wide range of dates rather than only one day of data.
Possible future work includes:

- Use a different performance evaluation metric, for example, Sharpe ratio.
- Learn the transition matrix beforehand using historical data instead of learn it in an online manner.
- Increase the number of possible actions at each time slot. For instance, a fraction of the volume $A_l$ suggested by the AC model instead of choosing only from 0 or $A_l$.

[1] R. Almgren and N. Chriss, "Optimal execution of portfolio transactions," *Journal of Risk*, pp. 5–39.

[2] N. Akbarzadeh, C. Tekin, and M. van der Schaar, "Online learning in limit order book trade execution," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 898–902, Nov 2017.