A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution

D. Hendricks and D. Wilcox

劉冠銘, 羅名志 / NYCU IMF

Outline

- Introduction
- Review : The Almgren-Chriss model
- Implementation
  - States, Actions, Rewards
  - Algorithm and Methodology
- Data & Results
- Implementation – polymer data

Introduction

Introduction

- Instead of *pure RL solution to the problem, here propose a hybrid approach :
  - Using Almgren-Chriss (AC) model as a base
  - the algorithm determines the proportion of the AC-suggested trajectory to trade based on prevailing volume / spread attributes, etc.
- The problem is a finite-horizon Markov Decision Problem (MDP)
- The model is compared with the base AC model

* Y. Nevmyvaka, Y. Feng., M. Kearns. **Reinforcement learning for optimal trade execution**, Proceedings of the 23rd international conference on machine learning, pp. 673-680, 2006.

Review

Almgren–Chriss Model

ref : AC_presentation

*R. Almgren, N. Chriss. Optimal execution of portfolio transactions

Almgren-Chriss Model

- Consider the execution of portfolio transactions with the aim of minimizing a combination of risk and market impact :

$$\min_x \big(E(x) + \lambda V(x)\big)$$

- derive closed form solutions with discrete time horizon, given volume
- includes risk aversion, permanent/temporary market impact

Parameters

$$x_k = X - \sum_{j=1}^{k} n_j = \sum_{j=k+1}^{N} n_j \quad k = 0, \ldots, N$$

- X : total shares to trade
- x_k : remaining inventory at time k
- τ : time interval (τ =T/N)
- $t_k = k\tau$ for k = 0, 1, …, N : total passed time
- $n_k = x_{k-1} - x_k$: trade size at time t
- $v_k = \frac{n_k}{\tau}$: velocity (shares per unit time)
- in this paper : sell side

Price Dynamics(theoretical price)

$$S_k = S_{k-1} + \sigma\tau^{1/2}\xi_k - \tau g\left(\frac{n_k}{\tau}\right) = S_0 + \sigma\tau^{1/2}\sum_{j=1}^{k}\xi_j - \tau\sum_{j=1}^{k} g(v_j)$$

ξ_j: standard normal deviation ($\mu = 0, \sigma = 1$)

σ : volatility of the asset

$g(v)$: permanent impact function

(function of average rate v, assume no drift term)

With Temporary impact (actual price)

$$\tilde{S}_k = S_{k-1} - h\left(\frac{n_k}{\tau}\right)$$

S_{k-1}: including its own permanent impact and deviation

$h(v)$: temporary impact function

(function of average rate v)

Trading trajectories

$$\tilde{S}_k = S_{k-1} - h\left(\frac{n_k}{\tau}\right)$$

$$S_k = S_0 + \sigma\tau^{1/2}\sum_{j=1}^{k}\xi_j - \tau\sum_{j=1}^{k} g(v_j)$$

- Capture(gain): $\sum_{k=1}^{N} n_k \tilde{S}_k = X S_0 + \sum_{k=1}^{N}\left(\sigma\tau^{1/2}\xi_k - \tau g\left(\frac{n_k}{\tau}\right)\right) x_k - \sum_{k=1}^{N} n_k h\left(\frac{n_k}{\tau}\right)$
- Trading cost: $X S_0 - \sum_{k=1}^{N} n_k \tilde{S}_k = -\sum_{k=1}^{N}\left(\sigma\tau^{1/2}\xi_k - \tau g\left(\frac{n_k}{\tau}\right)\right) x_k + \sum_{k=1}^{N} n_k h\left(\frac{n_k}{\tau}\right)$

Our goal is to minimize the trading negative utility

$$\min_x \left(E(x) + \lambda V(x)\right)$$

Minimize $X S_0 - \sum_{k=1}^{N} n_k \tilde{S}_k = - \sum_{k=1}^{N} \left(\sigma \tau^{1/2} \xi_k - \tau g\left(\frac{n_k}{\tau}\right)\right) x_k + \sum_{k=1}^{N} n_k h\left(\frac{n_k}{\tau}\right)$

- E(x) : expected cost of trading cost

$$E(x) = \sum_{k=1}^{N} \tau x_k \, g\left(\frac{n_k}{\tau}\right) + \sum_{k=1}^{N} n_k \, h\left(\frac{n_k}{\tau}\right)$$

- V(x) : variation of trading cost

$$V(x) = \sigma^2 \sum_{k=1}^{N} \tau x_k^2.$$

- We will show that for each value of λ such that E(x)+λV(x) is minimal

Assumption of Linear impact

$$S_k = S_0 + \sigma\tau^{1/2}\sum \xi_j - \tau\sum_{j=1}^{k} g(v_j)$$

$$\tilde{S}_k = S_{k-1} - h\left(\frac{n_k}{\tau}\right)$$

- For permanent

$g(v) = \gamma v$, where γ has units ($/share)/(share/time)

- For temporary

$h(v) = \varepsilon\,\mathrm{sgn}(n_k) + \eta\, v$

, where units of ε are $/share, and η are ($/share)/(share/time)
ε : fixed cost of selling (1/2 bid ask spread)

linear market impact model

$$E(x) = \sum_{k=1}^{N} \tau x_k \, g\left(\frac{n_k}{\tau}\right) + \sum_{k=1}^{N} n_k \, h\left(\frac{n_k}{\tau}\right)$$

$$V(x) = \sigma^2 \sum_{k=1}^{N} \tau x_k^2$$

- rewrite

(1) $$E(x) = \tfrac{1}{2}\gamma X^2 + \epsilon \sum_{k=1}^{N} |n_k| + \frac{\tilde{\eta}}{\tau} \sum_{k=1}^{N} n_k^2$$, where $\tilde{\eta} = \eta - \tfrac{1}{2}\gamma\tau$

(2) $$V(x) = \sigma^2 \sum_{k=1}^{N} \tau x_k^2$$

The utility function

$$\min_x \left(E(x) + \lambda V(x)\right)$$, where λ is risk aversion

the term ‘utility’ in this paper is the function above, to prevent ambiguity, here we use ‘negative utility’ to describe the combination of trading cost and risk.

optimal strategy

- $\min_x(E(x)+\lambda V(x))$ => differentiate negative utility

$$\frac{\partial U}{\partial x_j}$$

- we get optimal x_j, n_j:

$$x_j = \frac{\sinh\big(\kappa(T - t_j)\big)}{\sinh\big(\kappa T\big)} X$$

where $\tilde{k}^2 = \frac{\lambda\sigma^2}{\tilde{\eta}}$, and $\tilde{k}^2 = \frac{2}{\tau^2}(\cosh(k\tau) - 1)$

$$n_j = \frac{2\sinh\big(\frac{1}{2}\kappa\tau\big)}{\sinh\big(\kappa T\big)} \cosh\Big(\kappa\left(T - t_{j-\frac{1}{2}}\right)\Big) X$$

Implementation

States, Actions, Rewards

Algorithm and Methodology

States

- T = Trading Horizon
- V = Total Volume-to-Trade,
- H = Hour of day when trading will begin
- I = Number of remaining inventory states
- B = Number of spread states
- W = Number of volume states
- sp_n = %ile Spread of the nth tuple
- vp_n = %ile Bid/Ask Volume of the nth tuple

- Elapsed Time: t_n = 1, 2, 3, ..., T
- Remaining Inventory: i_n= 1, 2, 3, ..., I
- Spread State: $s_n = \begin{cases} 1, \quad if 0 < sp_n \leq \frac{1}{B} \\ 2, \ if \frac{1}{B} < sp_n \leq \frac{2}{B} \\ \quad \cdots \\ B, \ if \frac{B-1}{R} < sp_n \leq 1 \end{cases}$
- Volume state : $v_n = \begin{cases} 1, if \ 0 < vp_n \leq \frac{1}{W} \\ 2, if \frac{1}{W} < vp_n \leq \frac{2}{W} \\ \quad \cdots \\ W, if \frac{W-1}{W} < vp_n \leq W \end{cases}$

States

- for the nth episode, the state attributes can be summarized as the following tuple :
  - $z_n =< t_n, i_n, s_n, v_n >$
- for s_n, v_n, we first construct a historical distribution of spreads and volumes.

Actions

- Based on the AC model, we first calculate AC volume trajectory
- our learning agent is to modify the AC volume trajectory based on prevailing states
- the possible actions for our agent include :
  - $\beta_j = Proportion\ of\ AC_t\ to\ trade$
  - $\beta_{LB} = Lower\ bound\ of\ volume\ proportion\ to\ trade$
  - $\beta_{UB} = Upper\ bound\ of\ volume\ proportion\ to\ trade$
  - **Action :** $a_{jt} = \beta_j AC_t, where\ \beta_{LB} \leq \beta_j \leq \beta_{UB}\ and\ \beta_j = \beta_{j-1} + \beta_{incr}$

Rewards

- if we consider ‘Buy’
- every iteration, the rewards is calculate by initial price - average execution price (VWAP)
- need the information of LOB

Q-learning

$$V^*(x) = V^{\pi^*}(x) = \max_a\{ R_x(a) + \gamma \sum_y P_{xy}(a) V^{\pi^*}(y)\}$$

- Algorithm :
  - observes its current state x_n
  - selects and performs an action a_n
  - observes the subsequent state y_n as a result of performing action a_n
  - receives an immediate reward r_n and
  - uses a learning factor α_n , which decreases gradually over time.

  - Q is updated as follows :
    - $Q^\pi(x_n, a_n) = (1 - \alpha_n) Q^\pi(x_n, a_n) + \alpha_n(r_n + \gamma \max_b Q(x_{n+1}, b))$

Algorithm

```
Optimal_strategy (V, H, T, I, L)
    From episode 1 to n{
        For t = T to 1 {
                For i = 0 to I {
                        For a = 1 to A {
                                Set x = {t, i, s, v}
                                calculate IS from trade R(x,a)
                                Simulate transition x → y
                                Look up argmax Q(y, p)
                                Update Q(x, a) = Q(x, a) + alpha*U        }}}}
```

Data & Results

- market depth tick data (2012/06 – 2012/12), stocks from South Africa
- data include 5 levels of order book depth
- Stocks : SBK, AGL, SAB

- the RL model is able to improve implementation shortfall by 4.8%

Parameter

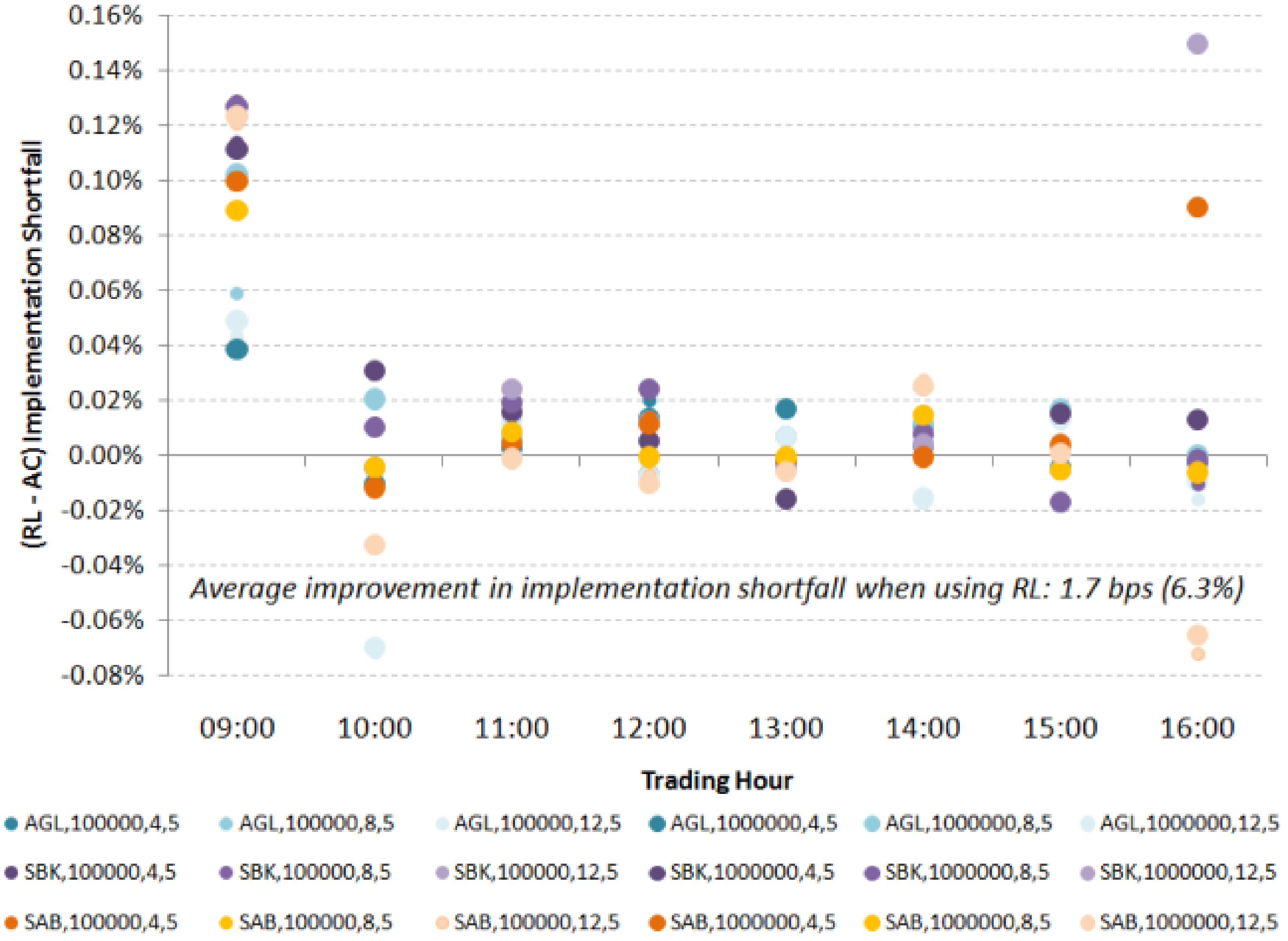
- β_{LB} = 0, β_{UB} = 2, β_{incr} = 0.25
- γ = 1, λ = 0.01(risk aversion coef)
- τ(time interval) = 5-min, α_0(initial decay factor) = 1
- T(trading horizon) : 4(20min), 8(40min), 12(60min)
- H(starting hour) : 9, 10, 11, 12, 13, 14, 15, 16
- V(total shares to trade) = 100000, 1000000
- I,B,W (nums of inventory / spread / volume states)= 5, 10
- BUY / SELL : BUY

Result

Parameters			Trading Time(hour)								Average
V	T	I,B,W	9	10	11	12	13	14	15	16	
100000	4	5	23.9	-1.4	4.7	13.4	1.8	3.3	1.8	35.1	10.3
100000	8	5	25.3	4.3	8.3	2.3	1.4	9.9	-0.6	-1.9	6.1
100000	12	5	32.7	-25.2	7.2	-2.7	-1.5	4.6	4.5	-3.3	2.1
1000000	4	5	23.3	-1.3	4.8	9.3	1.9	3.5	1.8	35.0	9.8
1000000	8	5	28.8	5.6	8.2	1.9	1.4	9.9	-0.3	-2.6	6.6
1000000	12	5	33.1	-25.0	7.2	-4.0	-0.8	4.8	4.8	1.2	2.7
100000	4	10	22.9	1.3	3.0	9.7	2.7	5.8	3.5	-26.1	2.8
100000	8	10	26.0	4.3	6.7	-0.2	3.5	8.6	1.6	-3.1	5.9
100000	12	10	27.8	-21.9	7.5	-4.1	0.6	1.8	6.2	-9.5	1.1
1000000	4	10	22.6	1.4	3.1	9.3	2.5	6.0	3.6	-26.1	2.8
1000000	8	10	26.3	5.0	7.2	-0.5	3.3	7.0	2.3	-1.8	6.1
1000000	12	10	27.9	-24.3	8.3	-6.9	0.5	1.8	7.5	-3.3	1.4

Parameters			Standard Deviation(%)		% improvement
V	T	I,B,W	AC	RL	in IS
100000	4	5	0.13	0.17	10.3
100000	8	5	0.14	0.23	6.1
100000	12	5	0.14	0.26	2.1
1000000	4	5	0.13	0.17	9.8
1000000	8	5	0.14	0.23	6.6
1000000	12	5	0.14	0.26	2.7
100000	4	10	0.13	0.17	2.8
100000	8	10	0.14	0.22	5.9
100000	12	10	0.14	0.26	1.1
1000000	4	10	0.13	0.17	2.8
1000000	8	10	0.14	0.22	6.1
1000000	12	10	0.14	0.26	1.4
Average			0.14	0.22	4.8

Result



0.16%
0.14%
0.12%
0.10%
0.08%
0.06%
0.04%
0.02%
0.00%
-0.02%
-0.04%
-0.06%
-0.08%
(RL - AC) Implementation Shortfall
Average improvement in implementation shortfall when using RL: 1.7 bps (6.3%)
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
Trading Hour
AGL,100000,4,5
AGL,100000,8,5
AGL,100000,12,5
AGL,1000000,4,5
AGL,1000000,8,5
AGL,1000000,12,5
SBK,100000,4,5
SBK,100000,8,5
SBK,100000,12,5
SBK,1000000,4,5
SBK,1000000,8,5
SBK,1000000,12,5
SAB,100000,4,5
SAB,100000,8,5
SAB,100000,12,5
SAB,1000000,4,5
SAB,1000000,8,5
SAB,1000000,12,5

Implementation

Setting and Parameter

- data :
  - 2914.T (**Japan Tobacco Inc.**)
  - training data (each day) : 2021 / 7 / 1 – 2022 / 6 / 23 (about 200 trading days)
  - testing data (each day): 2022 / 6 / 24 – 2022 / 6 / 30 (about 5 trading days)
- setting :
  - to sell X shares in time period T

Setting and Parameter

- β_{LB} = 0, β_{UB} = 2, β_{incr} = 0.25
- γ = 1, λ = $2 * 10^{-7}$(risk aversion coef)
- τ(time interval) = 1 hour, α_0(initial decay factor) = 1
- T(trading horizon) : 6 (1 day, 6 hours)
- H(starting hour) : 9
- V(total shares to trade) = 100000
- I,B,W (nums of inventory / spread / volume states)= 10
- BUY / SELL : SELL

Adjustment

- implementation shortfall : initial mid price – vwap
- cause we don't have market depth data :
  - **assume the distribution of LOB is uniformly distributed**
- trade direction is 'SELL ' :
  - implementation shortfall : vwap – initial mid price
- reward :
  - at t = T : action = inv
  - if shares to trade > inventory level : reward = inventory level – shares to trade

AC trajectory

	n[1]	n[2]	n[3]	n[4]	n[5]	n[6]	total volu	expected	variance	U(2e-07)	U(1e-20)	U(-2e-08)
2.00E-07	30558.09	22013.97	16178.45	12333.53	10006.11	8909.846	100000	252477.7	9.64E+11	445199.7	252477.7	233205.5
1.00E-20	16666.67	16666.67	16666.67	16666.67	16666.67	16666.67	100000	223234.9	1.32E+12	486773	223234.9	196881.1
-2.00E-08	14713.53	15762.89	16618.31	17269.25	17707.72	17928.31	100000	223885.1	1.38E+12	500140.9	223885.1	196259.5

the case of risk aversion :

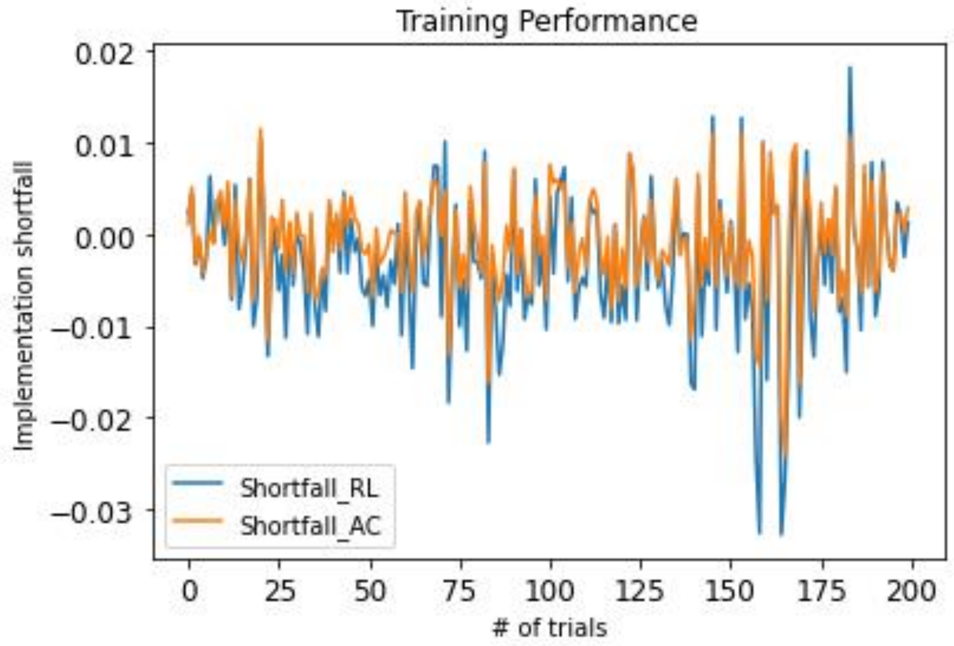
1 : 30558 (shares)
2 : 22014 (shares)
3 : 16178 (shares)
4 : 12334 (shares)
5 : 10006 (shares)
6 : 8910 (shares)

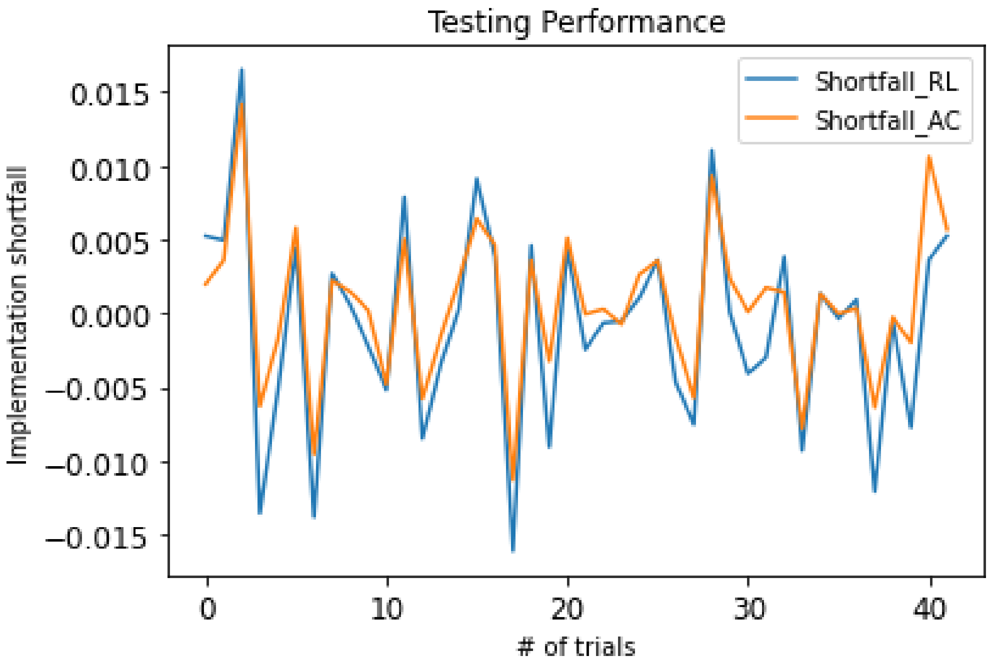
RL strategy

- after each epoch :
  - store the implementation shortfall of AC strategy and RL strategy
  - at the end compare the performance

	action	trade_ratio	AC_trajectory	trading shares
1	1	0.2	30558	6111
2	2	0.4	22014	8805
3	6	1.2	16178	19413
4	7	1.4	12334	17267
5	9	1.8	10006	18010
6	-1	nan	8910	30394

Result



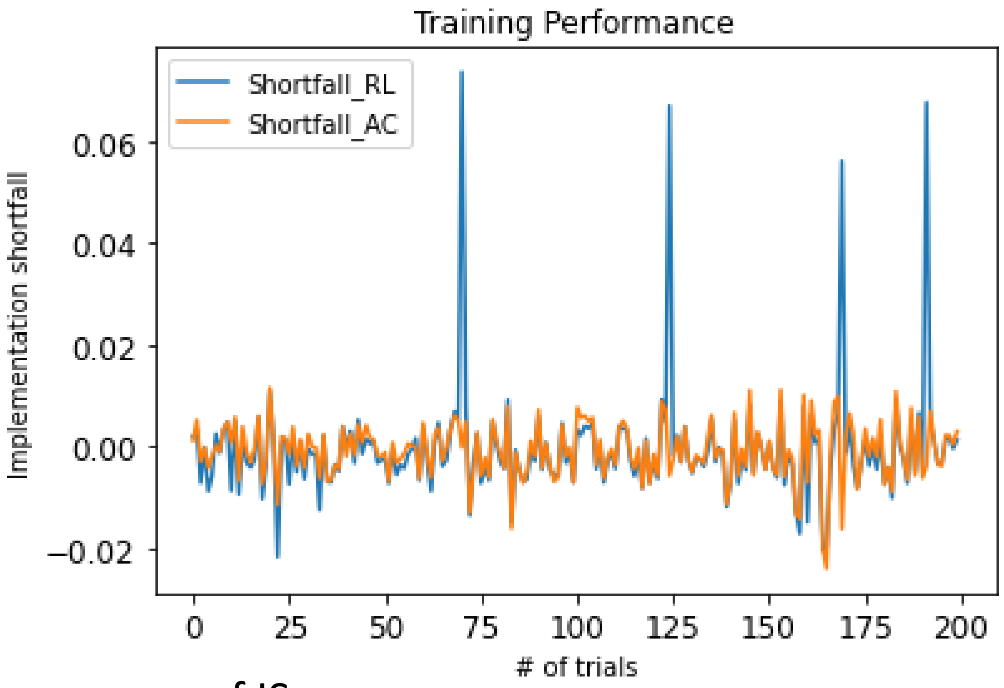


average of IS :
- RL strategy : - 0.00325
- AC strategy : - 0.00089

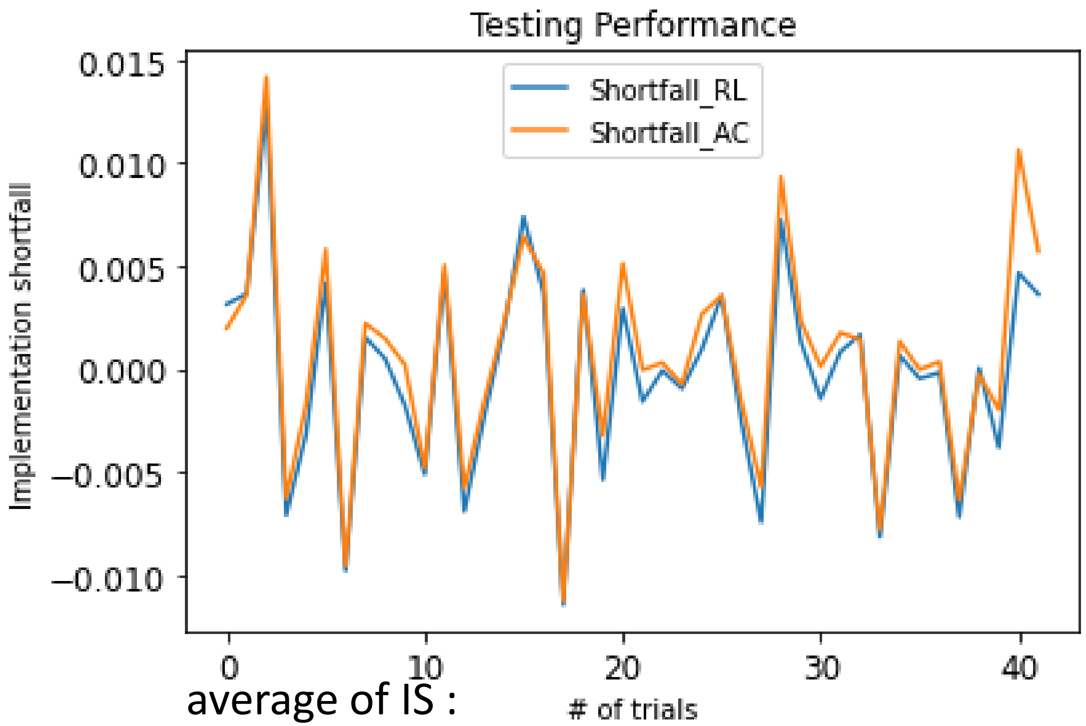
average of IS :
- RL strategy : -0.00083
- AC strategy : 0.00064

num of I, V, S

- assume I, V, S = 5 (from 10 to 5)



average of IS :
- RL strategy : -0.00051
- AC strategy : - 0.00089



average of IS :
- RL strategy : -0.00029
- AC strategy : 0.00064

possible problems

- no sufficient training data (Q table is sparse)
- Q update function
  - $Q = (1 - \alpha)Q(x, a) + (\alpha)[R(x, a) + \max Q(y, p))]$
- unit of remaining inventory and action aren’t consistent

Future Work

- time scale : hour -> minute
- consider temporary price impact
- using DQN
  - **“An application of deep reinforcement learning to algorithmic trading.”** 2021.
  - **“Double deep q-learning for optimal execution.”** 2021.
- continuous state, action

Ref.

- https://arxiv.org/pdf/1403.2229.pdf
- https://zhuanlan.zhihu.com/p/416082069
- https://youtu.be/z95ZYgPgXOY
- https://github.com/fdasilva59/Udacity-Deep-Reinforcement-Learning-Nanodegree/blob/master/drl-finance/4-DRL.ipynb
- https://zhuanlan.zhihu.com/p/431137706
- https://github.com/fdasilva59/Udacity-Deep-Reinforcement-Learning-Nanodegree/blob/master/drl-finance/syntheticChrissAlmgren.py
- https://www.zhihu.com/question/26408259
- https://youtu.be/Ye018rCVvOo