

Master Thesis

Reinforcement Learning for Optimal Order Placement in Stock Trading

Axel Perschmann

January 27, 2017

Version 1

Table of contents

1. Introduction
2. Reinforcement Learning Approach
3. Technical Background
4. Roadmap

Introduction

Order Book

An orderbook is a list of orders

- Indicates interest of buyers (bids) and sellers (asks)

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

- Alice is selling 25 shares of AIWC^a per 29\$
- Bob and Cedar are selling 20 and 30 shares respectively per 30\$
- David is selling 200 shares per 31\$.

^aAcme Internet Widget Company

Order types: Simple Market Order

Scenario 1: You want to buy 100 shares of AIWC.

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Solution: Buy them right away for the current market price:

- from **Alice**: $\Rightarrow 25 * 29\$ = 725\$$
- from **Bob and Cedar**: $\Rightarrow (20 + 30) * 30\$ = 1500\$$
- from **David**: $\Rightarrow 25 * 31\$ = 775\$$

Total: 3000\$ (avg price: 30\$)

Fast but costly (Average price differs from best price)

Order types: Limit Order

Scenario 2: Identical, buy only willing pay 30\$ per share.

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Solution: Place a limit order (also called *bid*): BUY 100 @30\$
75 shares fall in your bid limit and are matched immediately.

Order types: Limit Order

Scenario 2: Identical, buy only willing pay 30\$ per share.

	Amount	Type	Volume	VolumeAcc	norm_Price
28.50	100.0	bid	2850.0	9340.0	0.934510
28.70	200.0	bid	5740.0	6490.0	0.941068
30.00	25.0	bid	750.0	750.0	0.983695
30.50	NaN	center	NaN	NaN	NaN
31.00	200.0	ask	6200.0	6200.0	1.016485
31.50	100.0	ask	3150.0	9350.0	1.032879
32.00	75.0	ask	2400.0	11750.0	1.049274

Solution: Place a limit order (also called *bid*): BUY 100 @30\$

75 shares fall in your bid limit and are matched immediately.

25 remaining orders are added to orderbook and wait for matching offers.

- from Alice: $\Rightarrow 25 * 29\$ = 725\$$
- from Bob and Cedar: $\Rightarrow (20 + 30) * 30\$ = 1500\$$

Total: 2225\$ (avg price: 29.67\$)

Reduced Risk, but not guaranteed to execute fully

Limit Order Book: Visualization

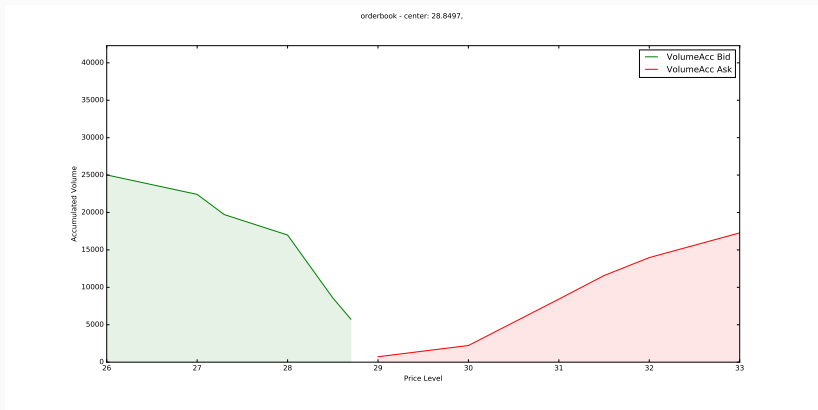


Figure 1: Limit Order Book (initial)

Limit Order Book: Visualization

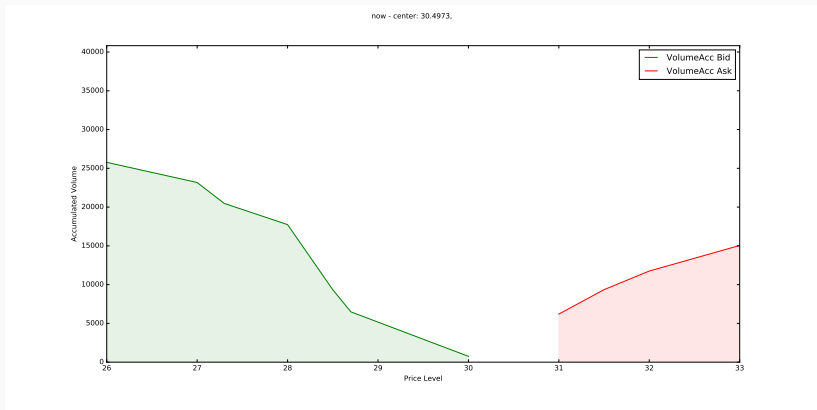


Figure 1: Limit Order Book (after trade)

Slippage

Placing small orders, chances are high to get a good price. Larger orders *eat* into the book and are fulfilled at successively worse price levels.

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

20 shares:	150 shares:	500 shares
Buy 20 @ 29	Buy 25 @ 29\$ Buy 50 @ 30\$ Buy 75 @ 31\$	Buy 25 @ 29\$ Buy 50 @ 30\$ Buy 200 @ 31\$ Buy 100 @ 31.50\$ Buy 75 @ 32\$ Buy 50 @ 33\$
580\$	4,549.50\$	15,625\$
Avg: 29\$	Avg: 30.33\$	Avg: 31.25\$
Slippage: 0.00\$	Slippage 1.33 \$	Slippage 2.25 \$
Cost: 0.00\$	Cost 200 \$	Cost 1,125 \$

Order strategy: Submit and Leave Strategy

Scenario: You want to buy 100 shares of AIWC within the next 2 hours.

Solution: Place a limit order at $t=0$: BUY 100 @29.5\$

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

25 shares fall in your bid limit and are matched immediately (at $t=0$).

Order strategy: Submit and Leave Strategy

Scenario: You want to buy 100 shares of AIWC within the next 2 hours.

Solution: Place a limit order at $t=0$: BUY 100 @29.5\$

	Amount	Type	Volume	VolumeAcc	norm_Price
28.50	100.0	bid	2850.0	10802.5	0.958006
28.70	200.0	bid	5740.0	7952.5	0.964729
29.50	75.0	bid	2212.5	2212.5	0.991620
29.75	NaN	center	NaN	NaN	NaN
30.00	50.0	ask	1500.0	1500.0	1.008427
31.00	200.0	ask	6200.0	7700.0	1.042041
31.50	100.0	ask	3150.0	10850.0	1.058848

75 shares remain as bid in the orderbook.

Order strategy: Submit and Leave Strategy

Scenario: You want to buy 100 shares of AIWC within the next 2 hours.

Solution: Place a limit order at $t=0$: BUY 100 @29.5\$

	Amount	Type	Volume	VolumeAcc	norm_Price
28.50	100.0	bid	2850.0	9917.5	0.958006
28.70	200.0	bid	5740.0	7067.5	0.964729
29.50	45.0	bid	1327.5	1327.5	0.991620
29.75	NaN	center	NaN	NaN	NaN
30.00	50.0	ask	1500.0	1500.0	1.008427
31.00	200.0	ask	6200.0	7700.0	1.042041
31.50	100.0	ask	3150.0	10850.0	1.058848

At $t=57$ Warren Buffett accepts your bid and sells you 30 shares for 29.5\$ each.

45 shares remain as bid in the orderbook.

Order strategy: Submit and Leave Strategy

Scenario: You want to buy 100 shares of AIWC within the next 2 hours.

Solution: Place a limit order at $t=0$: BUY 100 @29.5\$

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.954159
28.50	100.0	bid	2850.0	8590.0	0.971198
28.70	200.0	bid	5740.0	5740.0	0.978013
29.35	NaN	center	NaN	NaN	NaN
30.00	5.0	ask	150.0	150.0	1.022314
31.00	200.0	ask	6200.0	6350.0	1.056391
31.50	100.0	ask	3150.0	9500.0	1.073429

At $t=119$ time is over. Sell all remaining shares for current market price

- from Alice: $\Rightarrow 25 * 29\$ = 725\$$ (at $t=0$)
- from Warren Buffett: $\Rightarrow (30) * 29.5\$ = 885\$$ (at $t=57$)
- from Bob and Cedar: $\Rightarrow (20 + 25) * 30\$ = 1350\$$ (at $t=119$)

Total: 2960\$ (avg price: 29.6\$)

Saved 40\$ compared to Simple Market Order

Order strategy: Submit and Leave Strategy v2

- Place limit order, periodically (e.g. every 10 minutes) update limit, depending on trade process.
- Increasing aggression (= accepting more slippage) towards end of period)
- Use Reinforcement Learning to find optimal aggressiveness (=limits).
- Exploit Market/Orderbook Features
 - Indication of price fall or rise?!
 - More people trying to sell or to buy?!

Reinforcement Learning Approach

Reinforcement Learning Approach: Actions

Actions: $a \in \mathbb{R}[-1, 1]$

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Example:

current best price: 29.00\$

worst price level, when buying 100 shares: 31.00\$

$a = 1.00 \Rightarrow \text{limit} = 31$ (i.e. simple market order)

$a = 0.25 \Rightarrow \text{limit} = 29.5$

$a = 0.00 \Rightarrow \text{limit} = 29$ (i.e. buy only for current best price)

$a = -1.0 \Rightarrow \text{limit} = 27$

Reinforcement Learning Approach: Costs

Costs: Difference between benchmark and obtained price.

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Example:

current best price: 29.00\$ (benchmark: $29\$ * 100 = 2900\$$)

- $25 * 29\$ = 725\$$ (at $t=0$), costs=0
- $30 * 29.5\$ = 885\$$ (at $t=57$), costs = $30 * 0.5 \Rightarrow 15\$$
- $(20 + 25) * 30\$ = 1350\$$ (at $t=119$), costs = $45 * 1.0 \Rightarrow 45\$$

Total: 2960\$, Acc Costs: 60\$

Reinforcement Learning Approach: States

States: [private variables + market variables]

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Private Variables:

- Time left (t=119, 118, ..., 0)
- Volume left (vol=100, 75, 37.8, ..., 0)

Market Variables:

- Bid-Ask Spread
- Bid-Ask Volume Misbalance
- current market price
- ... research topic ...

Reinforcement Learning Approach: Algorithm

- Q-Learning
- Q Function Approximation: (1 layer NN with 64 neurons)
- Replay Memory

Technical Background

Data Origination

Data fetched from Poloniex Bitcoin Market Place via HTTP GET

request:

Call https://poloniex.com/public?command=returnOrderBook¤cyPair=USDT_BTC&depth=5000

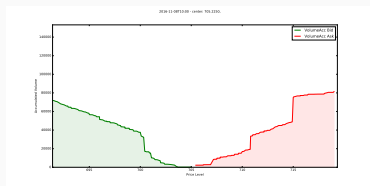
```
1 { "asks": [[ "705.450000" , 2.772181 ], [ "706.170000" , 0.052838 ],  
    ... ], "bids": [[ "705.000000" , 0.158232 ], [ "703.700000" ,  
    0.001250 ], ... ], "isFrozen": 0, "seq": 63413296 }
```

Data Origination

Data fetched from Poloniex Bitcoin Market Place via HTTP GET request:

Call https://poloniex.com/public?command=returnOrderBook¤cyPair=USDT_BTC&depth=5000

```
{ "asks": [[ "705.450000", 2.772181 ], [ "706.170000", 0.052838 ],  
  ... ], "bids": [[ "705.000000", 0.158232 ], [ "703.700000",  
  0.001250 ], ... ], "isFrozen": 0, "seq": 63413296 }
```



- 9 Currencypairs:
'USDT_BTC', 'BTC_ETH', 'BTC_XMR', '
BTC_XRP', 'BTC_FCT', 'BTC_NAV', '
BTC_DASH', 'BTC_MAID', 'BTC_ZEC'
- Frequency: 1 per Minute since
2016-11-08T10:00
- Datasize: ~290KB per full query
- 2.8GB for 109,582^a snapshots of
USDT_BTC

^aas per 2017-01-23T13:36

`manage_orderbooks.py` \Rightarrow `extract_orderbooks_for_one_currencypair`

- one file per currencypair, one line per minute
- round and aggregate similar pricelevels:

$$2.87 * 873.00 = \begin{cases} 0.34 * 873.00000122 \\ 2.53 * 873.00000121 \end{cases}$$

- Note: Bitcoins can be split into arbitrarily small chunks. In contrast it is usually not possible to buy 1.76 shares of stocks like AIWC.
- Conversion to custom class `OrderbookContainer`

Orderbook Container

Implemented in Python

```
ob=OrderbookContainer(timestamp="2016-11-08T10:00",
2                      bids=pd.DataFrame([200,100,300],
4                      columns=['Amount'], index=[28.7,28.5,28]),
                      asks=pd.DataFrame([25,50,200],
6                      columns=['Amount'], index=[29,30,31]))

# Available methods
8 ob.plot(outfile='sample.pdf') # plt.show or plt.savefig
ob.asks # pd.DataFrame
10 ob.bids # pd.DataFrame
ob.get_ask() # float
12 ob.get_current_price(volume=100) # float
ob.enrich() # Adds Volume, VolumeAcc and norm_Price
14 ob.head(depth=3) # pd.DataFrame
```

	Amount	Type	Volume	VolumeAcc	norm_Price
28.00	300.0	bid	8400.0	16990.0	0.970546
28.50	100.0	bid	2850.0	8590.0	0.987877
28.70	200.0	bid	5740.0	5740.0	0.994810
28.85	NaN	center	NaN	NaN	NaN
29.00	25.0	ask	725.0	725.0	1.005208
30.00	50.0	ask	1500.0	2225.0	1.039871
31.00	200.0	ask	6200.0	8425.0	1.074533

Orderbook Windows (1)

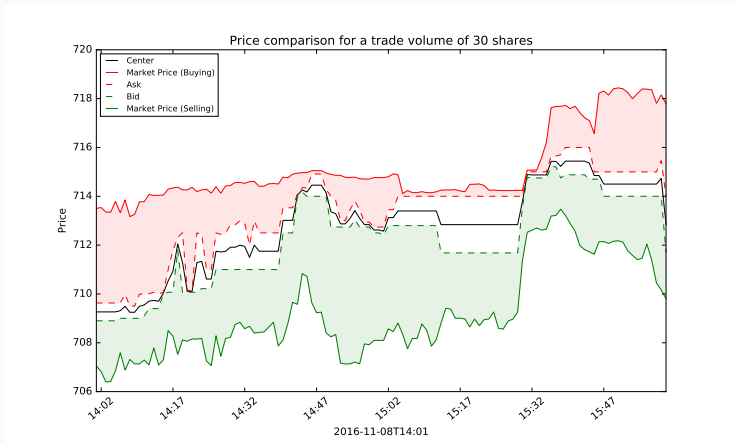


Figure 2: Episode Window over 120 minutes (increasing prices)

Ideal: Moderat or negativ aggression in early window. Aggressiv towards end of window

Orderbook Windows (2)

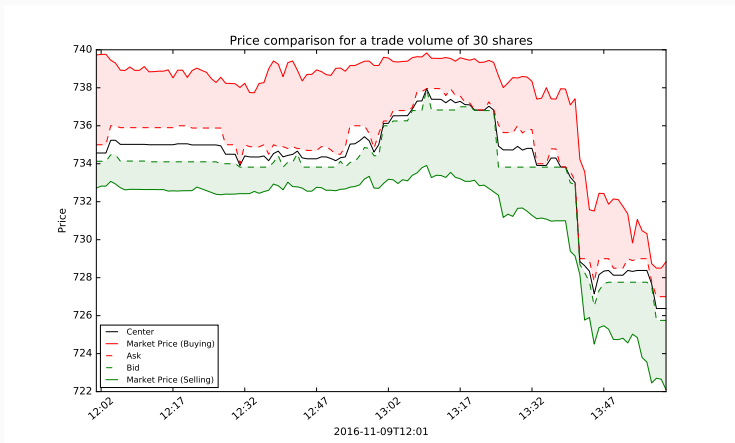


Figure 3: Episode Window over 120 minutes (decreasing prices)

Ideal: Moderat aggression in early window. Try everything to finish trade before the price drops.

Orderbook Trade Simulator

```
1 vol = 20
2 T   = 12
3 P   = 10
4 ots = OrderbookTradingSimulator(volume=vol, tradingperiods=T,
   decisionfrequency=P)
5 for e in range(T):
6     action=0.4
7     res = ots.trade(orderbooks=episode_windows[2][P*e:P*(e+1)],
8         agression_factor=action)
9
10 display(ots.history)
```

	a	ASK	BID	CENTER	LIMIT	SPREAD	VOLUME	avg	cashflow	cost	cost_avg	forced	high	low	traded	LIMIT_MAX
14:01	0.4	709.63	708.90	709.26	711.634	0.73	20.00	710.22	-5374.46	4.52	0.59	False	711.61	709.50	7.567243	714.64
14:11	0.4	710.66	709.10	709.87	712.120	1.56	12.43	711.79	-1810.06	5.49	2.16	False	712.12	710.01	2.542967	714.31
14:21	0.4	711.49	710.07	710.77	712.618	1.42	9.88	712.40	-6641.97	25.88	2.77	False	712.50	711.49	9.323293	714.31
14:31	0.4	713.00	711.00	711.99	713.000	2.00	0.56	713.00	-403.91	1.90	3.37	False	713.00	713.00	0.566497	713.00

- 12 Decision points (update of limit possible)
- timespan of 10 minutes

Roadmap

Roadmap: General Ideas

- Buying without limit (= Market Order) is costly, but fast.
- Setting a limit does not guarantee trade fulfillment
- Idea: Define trading period (e.g. 120 minutes)
 - Place limit order, periodically (e.g. every 10 minutes) adapt limit
 - Early Trading period: Careful order limits (avoid slippage)
 - End of Trading period: Aggressive order limits (accept more slippage)
 - Must trade all remaining shares at $t=120$ (no matter how poor the price)
- Use Reinforcement Learning to find optimal aggressiveness (=limits).
- Exploit Market/Orderbook Features
 - Indication of price fall or rise?!
 - More people trying to sell or to buy?!

Roadmap: Implementation

- ✓ Collect data
- ✓ Implement OrderbookContainer
- ✓ Implement OrderTradeSimulator
- Train RL-Agent on repeating data window
 - ✓ Actions: $\text{limit} = \text{ask} + a \mid a \in [\dots, -1, 0, 1, \dots]$
 - ✓ Costs: Slippage as from initial *ask*
 - ✓ States: [Remaining time, Remaining Volume]
 - Find optimal Q function for this data window
 - Learned strategy must have small costs
- Train RL-Agent on various, non-overlapping data windows
 - Expand State dimension: spread, ask-bid-ratio, ...
- Train RL-Agent on various currencypairs/stocks simultaneously

Questions?

