Adaptive Market Making via Online Learning

Jacob Abernethy
Computer Science and Engineering University of Michigan

Satyen Kale
IBM T. J. Watson Research Center

Team 2
Nefedova Maria, Matveeva Anna,
Sushkevich Lev, Sobakinskikh Ilya

Research Seminar, HSE
29.01.2022

Introduction

In this paper we focus on constructing MM strategies that achieve **non-stochastic** guarantees on profit and loss. We begin by proposing a class of market making strategies, parameterized by the choice of bid-ask spread and liquidity, and we establish a data-dependent expression for the profit and loss of each strategy at the end of a sequence of price fluctuations.

The Market Execution Framework

Assumptions of our model. In the described framework we make several simplifying assumptions on the trading execution mechanism, which we note here.

(1) The trader pays neither transaction fees nor borrowing costs when his cash balance is negative.

(2) Market orders are executed at exactly the posted market price, without “slippage” of any kind. This suggests that the market is very liquid relative to the actions of the MM.

(3) The market allows the buying and selling of fractional shares

(4) The price sequence is “exogenously” determined, meaning that the trades we make do not affect the current and future prices.

Spread-based Strategies

[maintain a fixed bid-ask spread]

We consider market making strategies parameterized by a window size b {1d, 2d, . . . , B}, where B is a multiple of d.

Before round t, the strategy S(b) selects a window of size b, [a_t, a_t + b], starting with a1 = p_1.

For some fixed liquidity density parameter A, it submits a buy order of A shares at every price p such that p < a_t and sell order of A shares such that p > a_t + b.

Depending on the price in the trading period p_t, the strategy adjusts the next window by the smallest amount necessary to include p_t

Spread-based Strategies

Algorithm 1 Spread-Based Strategy $S(b)$

Receive parameters $b > 0$, liquidity density $\alpha > 0$, inital price p_1 as input. Initialize $a_1 := p_1$.
for $t = 1, 2, \ldots, T$ **do**
  Observe market price p_t
  If $p_t < a_t$ **then** $a_{t+1} \leftarrow p_t$
  Else If $p_t > a_t + b$ **then** $a_{t+1} \leftarrow p_t - b$
  Else $a_{t+1} \leftarrow a_t$
  Submit limit order L_{t+1}: $L_{t+1}(p) = 0$ if $p \in [a_{t+1}, a_{t+1} + b]$, else $L_{t+1}(p) = \alpha$.
end for

Meta–Algorithm

We choose a pool of N strategies parametrized by spread window b.

Then we want to design an algorithm that achieves almost as much payoff as the best strategy (for some spread window b)

The proposed meta-algorithms:

- MMMW
- MMFPL
- FPL
- Uniform

The proposed MMW and MMFPL, FPL algorithms achieve regret bound of $\sqrt{\log(N)T}$.

Algorithm 2 Low regret meta-algorithm

1: Run every strategy $S(b)$ in parallel so that at the end of each time period t, all trades made by the strategies and the vectors H_{t+1}, C_{t+1} and $V_{t+1} \in \mathbb{R}^N$ can be computed.

2: Start a regret-minimizing algorithm $\mathcal{A}$ for learning from expert advice with one expert corresponding to each strategy $S(b)$ for $b \in \mathcal{B}$. Let the distribution over strategies generated by $\mathcal{A}$ at time t be w_t.

3: **for** $t = 1, 2, \ldots, T$ **do**

4: Execute any market orders from the previous period at the current market price p_t so that the inventory now equals $H_t \cdot w_t$. The cash value changes by $-(H_t \cdot (w_t - w_{t-1}))p_t$.

5: Execute any limit orders from the previous period: a w_t weighted combination of the limit orders of the strategies $S(b)$. The holdings change to $H_{t+1} \cdot w_t$, and the cash value changes by $(C_{t+1} - C_t) \cdot w_t$.

6: For each strategy $S(b)$ for $b \in \mathcal{B}$, set its payoff in round t to be $V_{t+1}(b) - V_t(b)$ and send these payoffs to $\mathcal{A}$.

7: Obtain the updated distribution w_{t+1} from $\mathcal{A}$.

8: Place a market order to buy $H_{t+1} \cdot (w_{t+1} - w_t)$ shares in the next period, and a w_{t+1} weighted combination of the limit orders of the strategies $S(b)$.

9: **end for**

MMMW

1. Initialize weights of each strategy to 1/N

2. In round t update the weights to $w_{t+1}(b) := w_t(b)\exp(\eta_t(V_{t+1}(b) - V_t(b)))/Z_t,$

   $V_t(b)$ is the payoff of strategy b in round t

   Z_t is a normalizing constant so that w_{t+1} is a distribution

   $$\eta_t = \min\left\{\sqrt{\frac{\log(N)}{t}}, \frac{1}{G_t}\right\}, \text{ where } G_t = \max_{\tau \leq t, b, b' \in \mathcal{B}} |V_\tau(b) - V_\tau(b')|$$

FTL (Follow the leader)

In each round simply choosing the action of the strategy that has the greatest payoff for the previous rounds

That is, we choose to use the strategy i_t = argmax(V_t(b)) during round t+1, where V_t(b) is the payoff of strategy b in round t and i_t the i-th agent.

MMFTL (Market Making Follow-The-Perturbed-Leader)

Just like FTL we choose the strategy which has the greatest payoff in the previous rounds.

When comparing the strategies, we add a random component p(b) to the previous payoff.

$$V_t(b) + p(b) \geq V_t(b') + p(b')$$

The random component is a sample from the exponential distribution with mean ƛ=1/η

Choose $\eta = \frac{1}{2G}\sqrt{\frac{\log(N)}{T}}$. *Then the regret of MMFPL is bounded by* $7G\sqrt{\log(N)T}$.

In the experiments $\eta = \sqrt{\frac{\log(N)}{T}}$

Uniform

Simple average of each strategy.

Basically MMMW without updating the weights.

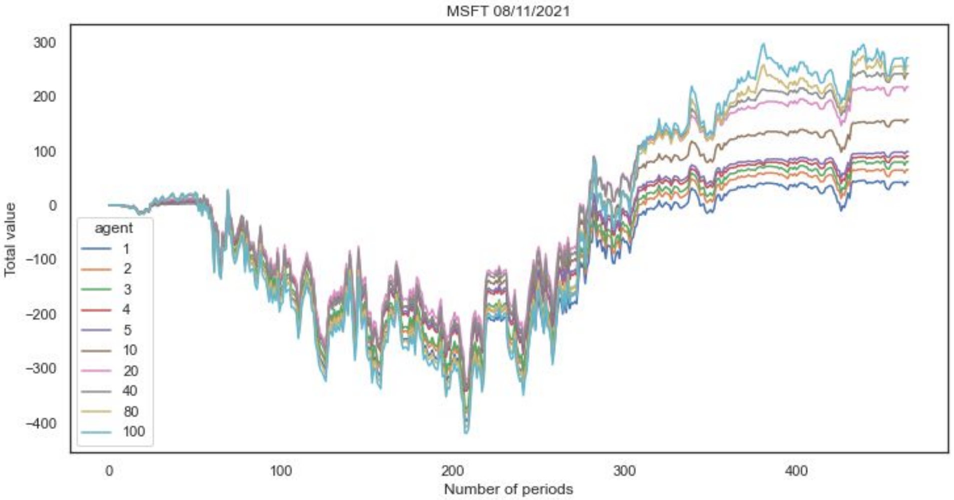
Experiments

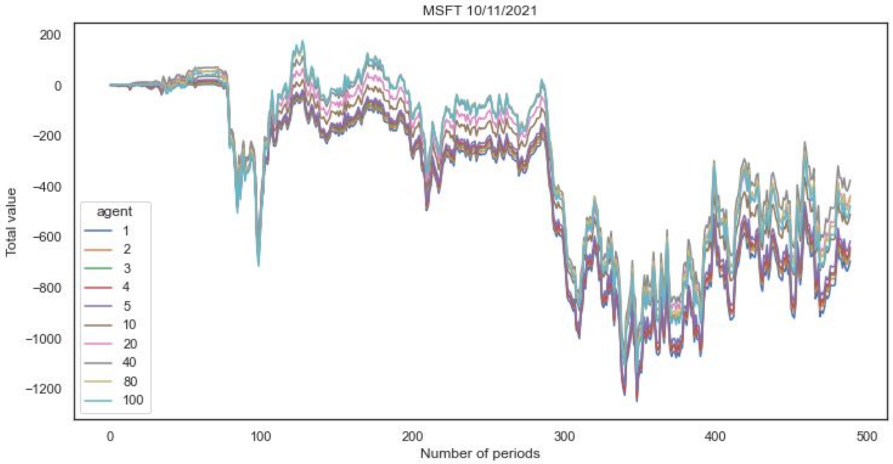
Authors conducted experiments with stock price data obtained from http://www.netfonds.no/. Authors downloaded data for the following stocks: MSFT, HPQ and WMT for May 8 and 9, 2013. The data consists of trades made throughout a given date in chronological order. Authors implemented MMMW, MMFPL, simple Follow-The-Leader2 (FTL), and simple uniform averaging over all strategies. Authors compared their performance to the best strategy in hindsight.

We have taken the data for the same stocks from finnhub for the time period from
from Nov 8 to Nov 12, 2021 due to absence of market data from May 8 to May 13, 2013.

Experiments

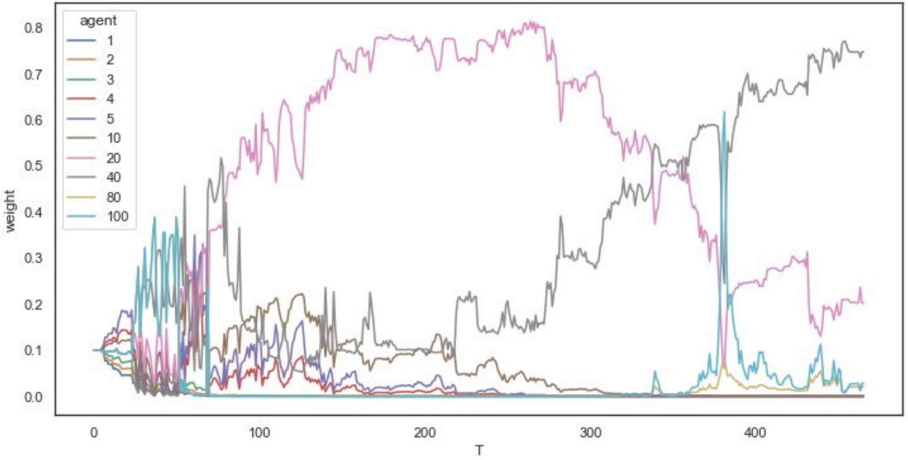
Agent with a spread of 100 was better on Nov 8, and with a spread of 40 was the best on Nov 10





MMMW weight evolution

MSFT on 08/11/21



Experiments

Same conclusions can be reached.

MMMW algorithm performs nearly as good as the Best strategy

MMFPL does not perform well.

The Best strategy differs for each day

	Symbol	Date	T	Best	MMMW	MMFPL	FTL	Uniform
0	HPQ	2021-11-08	398	-10.010	-13.469740	-18.175	-24.820	-14.6260
1	HPQ	2021-11-09	405	10.495	7.492334	2.155	4.800	5.5210
2	HPQ	2021-11-10	406	10.410	6.126768	13.635	18.615	3.7535
3	HPQ	2021-11-11	391	11.840	6.557494	6.760	7.110	4.5750
4	MSFT	2021-11-08	511	270.850	238.107957	185.135	187.750	151.9830
5	MSFT	2021-11-09	543	138.131	137.823051	46.766	73.481	8.9326
6	MSFT	2021-11-10	558	-375.230	-375.250606	-471.895	-467.549	-551.1577
7	MSFT	2021-11-11	532	233.150	232.881449	78.700	55.840	119.5125
8	WMT	2021-11-08	422	11.478	9.376191	-17.087	-9.782	-9.5433
9	WMT	2021-11-09	410	55.880	53.390251	49.310	24.885	31.6045
10	WMT	2021-11-10	453	98.725	94.653497	3.965	5.835	37.8350
11	WMT	2021-11-11	401	41.215	38.827534	29.345	23.765	18.5919

Code

Available on github

The spread based strategy is implemented in **Agent.py**

Data collection and calculations for plots and results table is in **NIPS.ipynb** jupyter notebook

The collected stock data is in **stocks.csv** file.

https://github.com/thxi/hse_nis