MASTER THESIS

High-frequency Market making strategy

Submitted by:

Thu Hang DO ID: 2001586
Hoang Viet LE ID: 2001585

in partial fulfillment of the conditions for the award of the degree
MSc. Quantitative Finance and Risk Management

on

28/01/2019

Supervisor: Dr. **Hans-Jorg von METTENHEIM**

"*A problem well put is half solved.*"

— John Dewey

Abstract

This study explores the use of market making strategy in high-frequency trading practice, typically in the cryptocurrency market as an example for its implementation. From this research, we aim to answer the question whether the market making strategy is a viable one and if it is the case, how it should be interpreted in market context. To properly address this problem, we will review a handful of published research and give the groundwork an explanation from both theoretical and market viewpoint. We are mainly concerned with the work of Avellaneda & Stoikov [3] as it is the foundation of other papers [7][5]. An empirical research is conducted to illustrate the employment on an actual trading engine. The strategy was experimented on Bitmex due to its data availability. From a two-month experiment, we show that the strategy is profitable and works more stably in some unfavourable circumstance than a naive strategy. Furthermore, a comparison on computational cost and precision between two solution proposals are also to be included. We conclude that given two proposals, precision is trade off by computational cost and vice versa. In high-frequency trading manner, computation efficiency is of more importance than precision in our view.

Acknowledgement

We owe great gratitude to Dr. von Mettenheim for his invaluable guidance during the time we did this thesis. Not only did he provided helpful comments on the subject but also was he engaged in the process so that we felt at easy to discuss with him. The completion of this paper could not have been accomplished without his support. We are also sincerely appreciated to all the professors who have been with us during these two years of master. To Dr. Taflin, Dr. Kortchemski, Dr. Aktar, Dr. Manolessou we would like to express our gratitude for their passing knowledge. They have given us priceless academic foundation so that we are capable of conducting this very research.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Technological advances of modern days bring about tremendous impact on financial market and high-frequency trading was one among several breakthroughs. Coined in the 1990s, High-frequency trading (HFT), as the name suggests, is characterised by short term holding period as well as low execution latency. It is a type of algorithmic trading that makes use of highly complex computerised rules to replace human intervenes in making trading decision and execution [1]. Scholars are not agreeing on how short the position holding period and how low the execution latency should be to be recognised as HFT. While its definition remains somewhat ambiguous, HFT strategy shares one thing in common: it strives to earn marginal profits on intense small trades [13]. HFT includes a handful types of strategy: market-making, event arbitrage, statistical arbitrage and latency arbitrage but this paper centers solely the former. Its use is catching more and more public attention and our interest has grown out of this rising trend. Nevertheless, it also gains popularity from debates. On the one hand, advocates claim that market-makers, by constantly posting quotes on the exchange, adds liquidity for a certain asset. On the other hand, naysayers argue that it gives rise to market unfairness in the sense that players can pay to locate their computers closer to the exchange platform thus have timing advantage over average investors [18]. While the strategy remains controversial, this paper aims neither to clarify its pros and cons nor to take side in the argument. It rather tries to shed some light on market-making strategy and its application in practice. In this paper we put together published papers on this topic and share our thoughts on the materials. Moreover, we particularly bring this strategy to use in cryptocurrency market owing to its data availability. By doing so, we hope to sketch a general guidance on how to apply the strategy in practice.

1.1 Structure of the thesis

This paper is organised into six chapters that firstly go through relevant industrial contexts and theoretical works, then through illustrative example ofthe application, and finally through the conclusive discussion.

The first chapter begins by drawing an overview of the target of our research - the market making strategy. A glance of its concepts and key notions will be presented in the following section. The third section will end this chapter by some description of the market where we will implement the strategy, that is the cryptocurrency market. In particular, we will describe the trading platform where we perform the strategy, the Bitmex exchange, and the asset that we choose to trade, the Bitcoin (XBT).

It would be incomplete had we not considered past academic effort in this field. The second chapter is therefore devoted to discuss the works of pioneers who set milestones in building the idea of market making. The chapter presents state-of-art in this type of trading from our point of view. Additionally, the chapter contains mathematical tools that are essential to understand the rationale of this trading mechanism.

Chapter 3 is the opening of the empirical part in our study. The chapter demonstrates the process of getting the dataset which will be of use in the later parts. The source of data, its descriptive statistics, its treatment procedure, etc. are included in this chapter.

Chapter 4 turns the spotlight on indispensable parameters which drive the strategy. In particular, the chapter exhibits a framework to calibrate the model using real data from chapter 3. The chapter contains mathematical explanation on the deriving of the estimators. The last part of this chapter exemplifies the results of the calibration process.

Chapter 5 will justifies the application of this strategy on cryptocurrency market. We will enforce the trading algorithm into the historical dataset in chapter 3 and compare its performance with that of other market making strategy using a set of generally accepted performance indicators.

The ending chapter will sum up our paper and put some words on its limitations. We will also list out recent ideas as for further development of this paper.

1.2 Overview of Market making strategy

Throughout this paper, we will repeatedly come across few fundamental notions in trading. First we go over them to grab a quick understanding of trading principles.

1. **Market order**: A market order will be executed as soon as it lands at the exchange. This order type guarantees to be traded at the best available price on the trading platform.
2. **Limit order**: A limit order restrains itself from being executed at worse-than-quoted price. Using limit orders is a passive way to trade but it shields order-posters from disadvantageous price shock. Limit orders stay in the book until a price "match" comes and fills the orders.
3. **Rebate/Transaction fee**: It costs traders transaction fee to place a market order on the exchange while they are rewared rebate fee for placing limit orders. The fee is usualy proportionate to the transaction amount.
4. **Limit order book**: A list of all existing limit orders in the exchange is the limit order book.
5. **Level of market data**: Exchange engines often provide free information about transactions but not entirely. This applies solely to Level I market data and is also the case in our study. Best bid and ask prices and their size (quantity) are often included in Level I data[15].

Market-making strategy refers to a trading practice in which the executors target to profit from the discrepancy between the ask and bid prices of the trading instrument, obviously the former should be higher than the latter for the sake of positive profit[6]. Traditionally, they are dealers who lay orders in both sides of the limit order book. In this manner, they realise the spread between the offer and bid prices as profit. When dealers place the limit orders frequently on the exchange and intensely in the order book, they supply liquidity for the trading asset. An asset is called liquid when it is sufficiently easy for traders to buy or sell the asset in no time. When the order book has reservations at almost every tick (the minimal increment amount at which the asset can be traded), traders have more options to match their price preference so it is intuitively clear that trades could be done rather effortlessly. Moreover, frequent placing from dealers ensures the continual supply of limit orders for other trade partners. This liquidity servicing function earns them the name market makers, and the pricing strategy in which dealers look for optimal bid-ask spread is termed market making strategy.

1.3 Cryptocurrency Market

1.3.1 The Bitcoin

Bitcoin was created by Satoshi Nakamoto [16] in 2008 with an intention to create a decentralised peer-to-peer electronic cash system that allows transaction to be performed between one party and another without dependency on financial institutions. For this reason, Bitcoin should be mainly used as an alternative currency to pay for goods and services and it will compete directly with fiat currency such as the US dollar and the like. However, ever since its creation, its purpose has gradually deviated from the original one. As the price of the Bitcoin along with its volatility was getting higher and higher, it reached its peak at nearly 20,000 USD in late-2017 (Figure 1.1), the Bitcoin nowadays has become popular among the public as well as investors as a speculative investment and behaves similar to an asset rather than a currency, as suggested by Baur, D.G. et al (2015) [4].

Figure 1.1: Price evolution of Bitcoin since 2017

Source: Bitmex

Due to the similarity with the traditional financial assets, Bitcoin will be the subject of our study to investigate if the standard market making model, which works in the traditional stock market [7], can still perform in the non-traditional market namely cryptocurrency market. Despite the existence of other cryptocurrencies nowadays (Ethereum, Ripple, so on and so forth), we decided to go on with Bitcoin because its capitalisation stays as the largest portion in the market. As of the time of our study, Bitcoin's transactions cover up to 66% market's, which is roughly 228bn USD.

1.3.2 The Bitmex Exchange

In this report, we study the evolution of Bitcoin price of an derivatives exchange called Bitmex. This is one of the largest Bitcoin exchanges based on its trading volume as of June 2019.

Regarding its history, Bitmex is an online crytocurrency derivatives exchange founded in 2014. The exchange operates with no fiat currency involvement, which means the margin and settlement are paid only in Bitcoin, the most popular cryptocurrency in the market. Due to its no fiat policy as well as being owned by an entity incorporated in Republic of Seychelles, a well-known tax shelter, Bitmex is not subject to any regulatory bodies such as the Commodity Features Trading Commission (CFTC) of the United States or the Securities and Futures Commission (SFC) of Hong Kong [2].The exchange requires no know-your-client (KYC) or anti-money laudering (AML) procedures which can be an obstacle for large financial institutions.

Unlike the traditional derivatives market for Bitcoin (CME) where the contract size denominated in Bitcoin and the based currency is fiat money, the Bitcoin derivative contracts in Bitmex use Bitcoin (XBT) as the based currency. Therefore, the profit and loss of Bitmex derivatives contracts is calculated by the inverse of XBT/USD which means the long (short) position in BTC is equal to a short (long) position on USD/XBT and the contract size is measured in USD not XBT.[1]

This core difference in the derivatives contracts of Bitmex leads to the creation of its unique contract type called the perpetual swap contract. The so called perpetual swap contract is basically a cross-currency swap between XBT and USD where the notion amount in XBT is continously rebalanced against a fixed notion amount in USD. In a typical currency swap, the difference in interest rates is exchanged between the holders of long and short positions. In the case of Bitmex, due to the fact that there is no interest rate in Bitcoin market, the mentioned difference in interest rate (or the funding rate, by

[1]Suppose that the price of Bitcoin changes from 10,000 USD to 9,000 USD. In the CME, if we short the futures of 10 XBT then the profit from the position will be:

$$(10,000 - 9,000) \times 10 = 10,000(USD)$$

The portfolio then will consist of 10 XBT and 10,000 USD with the total value of 100,000 USD.

If the same situation happens in Bitmex then a futures contract of 100,000 USD will be shorted, and the profit will be:

$$\Big(\frac{1}{9000} - \frac{1}{10000}\Big) = 1.11111(XBT)$$

The result position will be 11.11111 XBT which is equal to 100,000 USD.

Bitmex) is calculated to prevent the divergence between the swap rate and the reference Bitcoin spot price index of Bitmex (which is based on the Bitcoin spot price of a bucket of a few largest crytocurrency exchanges). As a result, the Bitcoin perpetual contract rate in Bitmex follows closely the general spot price of Bitcoin. In this research, for the sake of simplicity, we will use the rate of the contract instead of the Bitcoin spot price to do the backtest simulation. The creation of the perpetual contract on Bitmex has been a huge success as it is by far the most traded product on Bitmex and contributed to the position of the highest volume exchange in Bitcoin of Bitmex.

There are also a few other differences of Bitmex compared to the traditional market (CME) which might be the reasons for Bitmex's popularity. The first thing is that the Bitmex margin requirement is only 1% which means the maximum leverage can be up to 100 time. Furthermore, the transaction cost in Bitmex is quite low as the transaction fees for market order is 0.075% and the users of limit order can receive the rebate fees of 0.025% instead of having to pay any cost. This rebate scheme along with the Bitmex's support for trading via API lead to a strong incentive for crytocurrency-based high-frequency hedge fund to participate in the market. The contract unit in Bitmex is also significantly lower as it is only 1 USD compared to the 5 Bitcoin units in CME. The tick size is also lower which is only 0.5 USD in comparison with 5 USD of CME. The final characteristic is that the exchange operates 24/7, which is similar to spot exchange, not from Sunday to Friday, 5pm to 4pm of Chicago time in the case of CME.

In conclusion, all of those characteristics is the reason for Bitmex to become one of the top crytocurrency exchange with really high volume and diverse customer base. Alexander, C. et al. (2019) even suggest that the Bitcoin perpetual contract rate can be a leading indicator or have a net spillover effect to the spot price of Bitcoin in other exchanges [2].

Chapter 2

Literature review

2.1 Market-making model

To this point, it is clear that market makers are passive traders. Having posted both buying and selling limit orders, they wait until market orders arrive so that theirs are executed. A dilemma that every market maker faces is the trade-off between the bid-ask spread and the probability of execution. Should the spread be too narrow, quotes are more likely to be executed but on the other hand, market makers are exposed to adverse price jumps. Should it be too wide, market makers receive a thick margin but it is compensated by less likelihood to be executed. In case only one side of the orders is matched (either buy or sell), market makers risk holding or in short of asset for uncertain period until the other side of trade is filled. This waiting time is undesirable because asset price may move unfavourable to them. For instance, the market maker purchased an asset, then the selling price goes lower than the price at which he/she bought. The market maker has two options: either sell the asset forthwith or keep the asset in the portfolio until its value picks up then sell it. While the former results in a direct loss, the latter conceals inventory risk, or the risk that assets stay in the portfolio for indefinite time. One thing to bear in mind is that market makers have no desire for inventory. Their profits arise from the gap between selling and buying prices; left-over assets solely lead to value uncertainty so market maker do not favour inventory. Another concern for market makers is the presence of informed traders, ones with more precise information about imminent price movement. Informed traders trade with informational advantage and do so prior to marker makers, thus inducing adverse selection risk.[17]

As early as 1981, even before the birth of the concept "market maker", Ho & Stoll [9] proposed a pricing method for "dealers". The essence of their work was the use of a utility function whose expected maximum determines the optimal bid-ask quotes.

$$\max_{a,b} E[W_T] \text{ where } W_T = F_T + I_T + Y_T \tag{2.1}$$

Accumulated wealth and inventory compose dealers' utility and are deemed stochastic, driven by transactions and inventory return uncertainty. Arrival rate of market orders is assumed to follow a jump process: every unit time $\mathrm{d}t$, $\mathrm{d}q_b(\mathrm{d}q_a)$ unit of assets are to be bought (sold) with probability $\lambda_a \mathrm{d}t$ ($\lambda_b \mathrm{d}t$). Transaction intensities have linear relationship with the quoting prices a, b:

$$\begin{cases} \lambda_a = \alpha - \beta a \\ \lambda_b = \alpha - \beta b \end{cases} \tag{2.2}$$

Thus, cash F generated from this continual buying/selling act is the sum of trading margin plus self-return. Assets enter and exit the portfolio by $\mathrm{d}q_b$ and $\mathrm{d}q_a$ units, which stand for the change in quantity of assets in the portfolio. When the volume of asset inflow exceeds outflow, inventory amasses. r_I denotes the return on inventory account and $\sigma_I^2 \mathrm{d}Z_I$ reflects inventory return uncertainty, or the variance, where Z_I is a standard Gaussian random variable. The base wealth Y is unaffected by quoting prices but collateralises the short position; collateral can be in form of cash or shares.

$$\mathrm{d}F = rF\mathrm{d}t - (p-b)\mathrm{d}q_b + (p+a)\mathrm{d}q_a \tag{2.3}$$

$$\mathrm{d}I = p\mathrm{d}q_b - p\mathrm{d}q_a + r_I I\mathrm{d}t + \sigma_I^2 I\mathrm{d}Z_I \tag{2.4}$$

$$\mathrm{d}Y = r_Y Y\mathrm{d}t + \sigma_Y^2 Y\mathrm{d}Z_Y \tag{2.5}$$

The last part of this paper presents resolution for this optimisation problem using the well-known Hamilton-Jacobian-Bellman equation. While the model was indeed pioneer, it constitutes profound flaws. Firstly, quoting prices are anchored to a "true" constant asset price and this raises ambiguity for the model. Had buyers known about the "true" price, they would not have traded the asset at a different one. Linearity assumption between quoting prices and arrival rate intensity is also a concern because later in chapter 3, we can see clearly that they have non-linear relation.

Influenced by and elaborated this idea to a further step, Avellaneda & Stoikov published a paper in 2008 [3]. They adopted a generally accepted constant absolute risk-

aversion (CARA) exponential function to measure market makers' satisfaction upon accumulated profit and inventory afore liquidation at T. Each market maker is characterised by distinct risk aversion coefficient γ. The negative sign preceding this positive constant, together with that of the utility function, guarantees the law of diminishing for marginal utility.

$$V(t, x, q, s) = \max_{\delta^a, \delta^b} E_t[-exp(-\gamma(X_T + q_T S_T))] \tag{2.6}$$

The break-through of this model lies at the design of "true" price as the middle point of best buying (lowest bid) and selling (highest ask) prices. Not only does it illuminate the definition of a "true" price but also does it entail the calibration in practice. In their article, Avellaneda & Stoikov chose a standard one-dimensional Wiener process with quadratic variation σ^2 to model the referencing price. While this assumption does not comply with the generally accepted geometrical Brownian motion, it remains adequately rational in infinitesimal time interval as an implication of stationary price evolution in very small time window. This set up also ensures boundedness for utility function.

$$\mathrm{d}S_t = \sigma \mathrm{d}W_t \tag{2.7}$$

The mid-price S, multiplied by units of holding assets q, values the inventory amount though it does not necessary mean that market makers trade cost-free. Let N_t^b and N_t^a denote the number of assets to be bought and sold at time t, q is inherently the difference between the former and the latter: it is the left-over assets remaining from buying and selling activities.

$$q_t = N_t^b - N_t^a \tag{2.8}$$

Buying and selling prices are set δ^b lower and δ^a higher than mid-price S; both are strictly positive. Once limit orders are quoted, market orders come and fill the limit order book. Additional profit $\mathrm{d}X_t$ yields from newly matched selling and buying amount.

$$\mathrm{d}X_t = (S_t + \delta_t^a)\mathrm{d}N_t^a - (S_t - \delta_t^b)\mathrm{d}N_t^b \tag{2.9}$$

Trading volume is unknown prior to t but is unarguably dependent on prices. The pricier the asset is, the less preferable it is in the eyes of traders. Unlike that in Ho & Stoll's model, N_t here is assumed to follow a Poisson process having exponential relationship with δ, the "distance" to mid-price. The intensity of trades $\lambda(\delta)$ is of the following form saying that prices farther away from S_t are less likely to be executed. A and k are parameters

defining the intensity function.

$$\lambda(\delta) = Ae^{-k\delta} \tag{2.10}$$

Last but not least, the authors gave the solution for the equation 2.6 using dynamic programming principle (DPP). Mathematical explanation of this is disclosed in the next section. To sum up, Avellaneda & Stoikov formulated an innovative and flexible framework for market making strategy that stimulates contemporary research on market-making strategy. After this work, several studies emerged attempting to extend the model. One way is to impose dissimilar model supposition for the mid-price S_t. Depending on the nature of the trading asset, mid-price evolution owns distinctive trait thus should be assumed differently. For example, the interest rate is believed to be a mean-reverting process whilst stock price return more than often admits a diffusive process (either with trend or not).

1. Drift-diffusion process:

$$\mathrm{d}S_t = \mu \mathrm{d}t + \sigma \mathrm{d}W_t \tag{2.11}$$

   where μ depicts the price momentum.

2. Mean-reverting process:

$$\mathrm{d}S_t = \theta(\mu(t) - S_t)\mathrm{d}t + \sigma \mathrm{d}W_t \tag{2.12}$$

   where $\mu(t)$ is the long-run equilibrium. This parameter can be time-dependent or not. When it is not, the process is the original Vasicek model and when it is so, the process admits the Hull-White model. θ denotes the reversion rate, or the rate at which the process reverts to the long-run equilibrium.

Likewise, it is inevitable to mention the work addressing other source of risk for market makers, namely adverse selection risk. This risk refers to a phenomenon that market makers have a sell limit order filled just ahead of a price increment or a buy limit order filled before prior to price downturn, hence the name adverse selection. Cartea *et al.* (2015) [5] suggested two approaches to incorporate this risk into the model. One proposal is to assume that market order flow casts a measurable effect ξ to the mid-price: the price moves downward following a purchase and upward succeeding a sale.

$$\mathrm{d}S_t = \sigma \mathrm{d}W_t + \xi^a \mathrm{d}N_t^a - \xi^b \mathrm{d}N_t^b \tag{2.13}$$

The other scheme says the mid-price is affected by a momentary component denoted α_t in addition to the usual long-term effect μ. This model can be viewed as a modification of the usual drift-diffusion model with an additional abrupt drift. The temporary effect can take any model but is usually modeled after a zero-mean reverting process. Owing to the fact that α_t is regulated by unknown random factors, this model is more difficult to implement.

$$\mathrm{d}S_t = (\mu + \alpha_t)\mathrm{d}t + \sigma \mathrm{d}W_t \tag{2.14}$$

All in all, the market making model is dynamic and therefore applicable to a wide range of asset classes. The first and foremost criterion as whether or not to apply the strategy on any asset is to consider its trading mechanism. The model requires the existence of limit order books for market makers to quote their reservation prices and for market orders to fill. Having discussed the Bitmex exchange in chapter 1, we believe that it is appropriate to exercise this methodology on the cryptocurrency market since its trading mechanism resembles that of stock market.

2.2 Mathematical tools

In this section, we review key mathematical tools which are essential in market making model. The goal is to correctly understand their concepts and usages so as to properly develop the framework for application. The whole section concerns the basic model by Avellaneda-Stoikov as a mathematical illustration. The section starts with the most important knowledge in stochastic calculus that is Ito integral. Next, we study the dynamic programming principle and its solution to the market making model. Finally, we revise a classical calibration tool, i.e. Levenberg-Marquardt algorithm.

2.2.1 Ito's Lemma

Needless to say, Ito's Lemma is the heart of stochastic calculus. It concerns the derivative (and integral) of a twice-derivable function V with respect to the vector of n random variables $X_t = (X_t^1, ..., X_t^n)'$ on the filtration generated by X. The general formula states:

$$\mathrm{d}V(X_t) = \frac{\partial V}{\partial X_t} \cdot \mathrm{d}X_t + \frac{1}{2}\sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\frac{\partial^2 V}{\partial X^i \partial X^j}\mathrm{d} < X^i, X^j >_t \tag{2.15}$$

In market making model, we concern the function $V(t, x, q, s)$ which is driven by three

processes (t, x, s) and two of them (x, s) are stochastic. Applying Ito's rule on $\mathrm{d}V(t, x, q, s)$ gives us the following result.

$$\begin{aligned}
\mathrm{d}V(t,x,q,s) =& \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial s}dS_t + \frac{1}{2}\frac{\partial^2 V}{\partial s^2}d < S, S >_t \\
&+ \frac{V(t, x+s+\delta_t^a, q-1, s) - V(t,x,q,s)}{s+\delta_t^a}(s+\delta_t^a)dN_t^a \\
&- \frac{V(t,x,q,s) - V(t, x-s+\delta_t^b, q+1, s)}{s-\delta_t^b}(s-\delta_t^b)dN_t^b \\
=& \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial s}\sigma dW_t + \frac{1}{2}\frac{\partial^2 V}{\partial s^2}\sigma^2 dt \\
&+ (V(t, x-s+\delta_t^b, q+1, s) - V(t,x,q,s))dN_t^b \\
&+ (V(t, x+s+\delta_t^a, q-1, s) - V(t,x,q,s))dN_t^a
\end{aligned}$$

2.2.2 Hamilton-Jacobi-Bellman equation

The Hamilton-Jacobi-Bellman equation [20][10] arose from the Dynamic programming principle dated back in the mid-20th century. It is a nonlinear partial differential equation (PDE) that serves not only as necessary but also sufficient condition for an optimisation problem. The equation plays a vital role in market-making framework so we again state the problem here and give its solution an explanation.

Our aim is to maximise the utility function $U(t, x, q, s)$ at liquidating time T and the control function is subject to a terminal condition as in equation 2.16.

$$\begin{cases} V(t,x,q,s) = \max_{\delta^a, \delta^b} E\big[V(T, X_T, q_T, S_T)|\mathcal{F}_t\big] \\ V(T, X_T, q_T, S_T) = -\exp(-\gamma(X_T + q_T S_T)) \end{cases} \tag{2.16}$$

where δ^a, δ^b are the predictable control processes. Driven by random processes X_t the inventory amount and the price evolution S_t , $V(t, x, q, s)$ itself is random. Being stochastic, the function admits Ito's Lemma.

$$\begin{aligned}
V(t,x,q,s) &= \max_{\delta^a, \delta^b} E[V(T, X_T, q_T, S_T)|\mathcal{F}_t] \\
V(t,x,q,s) &= \max_{\delta^a, \delta^b} E\left[V_t + \int_t^T \mathrm{d}V_s|\mathcal{F}_t\right] \\
\Longrightarrow \max_{\delta^a, \delta^b} E[\mathrm{d}V_t] &= 0
\end{aligned}$$

Applying Ito's rule on $\mathrm{d}V(t,x,q,s)$, taking the expectation and dividing both sides of the equation by $\mathrm{d}t$, we acquire the HJB equation as in 2.17. We highlight the fact that the wealth process X_t changes accordingly to the value of the sale/purchase. When one unit of asset is sold, inventory level decreases by one unit $(q-1)$ and the wealth recognises the value of this sale that is $(s+\delta^a)$. Vice versa, a purchase adds one unit of asset into the inventory $(q+1)$ and the wealth loses an amount of $(s-\delta^b)$. Therefore, $V(t,x,q,s)$ needs to be derivated discretely in terms of x.

$$\begin{cases} 0 = \dfrac{\partial V}{\partial t} + \dfrac{1}{2}\dfrac{\partial^2 V}{\partial s^2}\sigma^2 + \max\limits_{\delta^b} \lambda^b (V(t, x-s+\delta^b_t, q+1, s) - V(t,x,q,s)) \\ \qquad\qquad + \max\limits_{\delta^a} \lambda^a (V(t, x+s+\delta^a_t, q-1, s) - V(t,x,q,s)) \\ V(T,x,q,s) = -\exp(-\gamma(x+qs)) \end{cases} \tag{2.17}$$

To the best of our knowledge, we are aware of two approaches to solve 2.17. We first introduce the one employed by Avellaneda & Stoikov. In view of the exponential utility function, they speculate the ansatz:

$$V(t,x,q,s) = -\exp(-\gamma x)\exp\big(-\gamma\theta(t,q,s)\big) \tag{2.18}$$

Then equation 2.17 is translated to 2.19.

$$\begin{cases} 0 = \dfrac{\partial \theta}{\partial t} + \dfrac{1}{2}\sigma^2\dfrac{\partial^2\theta}{\partial s^2} - \dfrac{1}{2}\sigma^2\gamma\left(\dfrac{\partial\theta}{\partial s}\right)^2 \\ \qquad + \max\limits_{\delta^a} \dfrac{\lambda^a}{\gamma}\left[1 - \exp\Big(-\gamma\Big(s+\delta^a - \big(\theta(t,q,s) - \theta(t,q-1,s)\big)\Big)\Big)\right] \\ \qquad + \max\limits_{\delta^b} \dfrac{\lambda^b}{\gamma}\left[1 - \exp\Big(-\gamma\Big(-s+\delta^b + \big(\theta(t,q+1,s) - \theta(t,q,s)\big)\Big)\Big)\right] \\ \theta(T,q,s) = qs \end{cases} \tag{2.19}$$

Proof: Deriving 2.19 from 2.17.
Each derivative term in equation 2.17 can be derived using the chain rule.

$$\begin{aligned} \frac{\partial V}{\partial t} &= -\gamma V(t,x,q,s)\frac{\partial\theta}{\partial t} \\ \frac{\partial V}{\partial s} &= -\gamma V(t,x,q,s)\frac{\partial\theta}{\partial s} \\ \frac{\partial^2 V}{\partial s^2} &= \gamma^2 V(t,x,q,s)\left(\frac{\partial\theta}{\partial s}\right)^2 - \gamma V(t,x,q,s)\frac{\partial^2\theta}{\partial s^2} \end{aligned}$$

Next, the value function $V(t, x-s+\delta_t^b, q+1, s), V(t, x+s+\delta_t^a, q-1, s)$ are rewritten in form of the base value $V(t, x, q, s)$.

$$\begin{aligned} V(t, x-s+\delta_t^b, q+1, s) &= -\exp\big(-\gamma(x-s+\delta_t^b)\big)\exp\big(-\gamma\theta(t, q+1, s)\big) \\ &= V(t, x, q, s)\exp\Big(-\gamma\big(-s+\delta^b+\theta(t, q+1, s)-\theta(t, q, s)\big)\Big) \\ V(t, x+s+\delta_t^a, q-1, s) &= -\exp\big(-\gamma(x+s+\delta_t^a)\big)\exp\big(-\gamma\theta(t, q-1, s)\big) \\ &= V(t, x, q, s)\exp\Big(-\gamma\big(s+\delta^a+\theta(t, q-1, s)-\theta(t, q, s)\big)\Big) \end{aligned}$$

Finally, substituting them into equation 2.17 and dividing both sides by $-\gamma V(t, x, q, s)$ we get the equation 2.19.

The differential term of $\theta(t, q, s)$ in terms of q lies vividly inside equation 2.17 , Avellaneda & Stoikov suggested asymptotic expansion in the inventory variable q. This action factors out the inventory level in $\theta(t, q, s)$ owing to it being discrete variable.

$$\begin{aligned} \theta(t, q, s) &= \theta^0(t, s) + q\theta^1(t, s) + \frac{1}{2}q^2\theta^2(t, s) \\ \theta(t, q+1, s) &= \theta^0(t, s) + (q+1)\theta^1(t, s) + \frac{1}{2}(q+1)^2\theta^2(t, s) \\ \theta(t, q-1, s) &= \theta^0(t, s) + (q-1)\theta^1(t, s) + \frac{1}{2}(q-1)^2\theta^2(t, s) \end{aligned} \tag{2.20}$$

Thus,

$$\begin{aligned} \theta(t, q+1, s) - \theta(t, q, s) &= \theta^1(t, s) + \frac{1}{2}(2q+1)\theta^2(t, s) \\ \theta(t, q, s) - \theta(t, q-1, s) &= \theta^1(t, s) + \frac{1}{2}(2q-1)\theta^2(t, s) \end{aligned} \tag{2.21}$$

Essentially, we need to answer the optimal terms in equation 2.17 so as to solve it. As they are alike, here we only explain one term denoted as $L(\delta^a)$; the other is analogue. The left-hand side of equation 2.21 is exchanged by its right-hand side into the optimising function.

$$L(\delta^a) = \frac{Ae^{-k\delta^a}}{\gamma}\left[1 - \exp\Big(-\gamma\big(s+\delta^a-\theta^1(t, s)-\frac{1}{2}(2q-1)\theta^2(t, s)\big)\Big)\right] \tag{2.22}$$

The optimal point is satisfied if and only if $\frac{\mathrm{d}L}{\mathrm{d}\delta^a} = 0$, the critical point is unique.

$$\begin{aligned} \iff 0 \quad &= \frac{Ae^{-k\delta^a}}{\gamma}\left[-k + (k+\gamma)\exp\Big(-\gamma\big(s + \delta^a - \theta^1(t,s) - \frac{1}{2}(2q-1)\theta^2(t,s)\big)\Big)\right] \\ \iff \frac{k}{k+\gamma} \quad &= \exp\Big(-\gamma\big(s + \delta^a - \theta^1(t,s) - \frac{1}{2}(2q-1)\theta^2(t,s)\big)\Big) \end{aligned} \tag{2.23}$$

Consequently, switching the result as in equation 2.23 into the original one 2.22 caters us the optimal solution $L^*(\delta^a)$

$$\max_{\delta^a} L^*(\delta^a) = \frac{Ae^{-k\delta^a}}{\gamma}\left(1 - \frac{k}{k+\gamma}\right) = \frac{A}{k+\gamma}e^{-k\delta^a} \tag{2.24}$$

Swapping optimal terms into equation 2.19 we acquire the following equation. Noticeably, the last term $\frac{A}{k+\gamma}\big(e^{-k\delta^a} + e^{-k\delta^b}\big)$ is no longer dependent on the inventory level.

$$\begin{cases} 0 = \dfrac{\partial\theta}{\partial t} + \dfrac{1}{2}\sigma^2\dfrac{\partial^2\theta}{\partial s^2} - \dfrac{1}{2}\sigma^2\gamma\left(\dfrac{\partial\theta}{\partial s}\right)^2 + \dfrac{A}{k+\gamma}\big(e^{-k\delta^a} + e^{-k\delta^b}\big) \\ \theta(T,q,s) = qs \end{cases} \tag{2.25}$$

At this stage, each derivative term of $\theta(t,q,s)$ is replaced by those of the asymptotic expansion[1]. Then, we have an equation from which we can factorise the terms into two groups of q and q^2. The former infers equation 2.26 and the latter yields equation 2.27.

$$\begin{cases} 0 = \dfrac{\partial\theta^1}{\partial t} + \dfrac{1}{2}\sigma^2\dfrac{\partial^2\theta^1}{\partial s^2} \\ \theta^1(T,s) = s \end{cases} \tag{2.26}$$

$$\begin{cases} 0 = \dfrac{\partial\theta^2}{\partial t} + \dfrac{1}{2}\sigma^2\dfrac{\partial^2\theta^2}{\partial s^2} - \dfrac{1}{2}\sigma^2\gamma\left(\dfrac{\partial\theta^1}{\partial s}\right)^2 \\ \theta^2(T,s) = 0 \end{cases} \tag{2.27}$$

We can quickly see that $\theta^1(t,s) = s$ and $\theta^2(t,s) = -\frac{\gamma\sigma^2}{2}(T-t)$ are the solutions for these PDEs. Substituting them into 2.22 we finally attain the results for δ^a, δ^b as in equation 2.28.

$$\begin{aligned} \delta^{a*} &= \frac{1}{\gamma}\ln\left(1 + \frac{\gamma}{k}\right) - \frac{\gamma\sigma^2}{4}(2q-1)(T-t) \\ \delta^{b*} &= \frac{1}{\gamma}\ln\left(1 + \frac{\gamma}{k}\right) + \frac{\gamma\sigma^2}{4}(2q+1)(T-t) \end{aligned} \tag{2.28}$$

Inasmuch as this method grants closed-form formulas, it is indeed computationally

[1] Refer to equation 2.25

beneficial, especially in HFT where speedy algorithm is a must. Nonetheless, the approximation procedure is sill extensively challenging when it comes to sophisticated assumption on the mid-price process. Tapia (2015) [7], on the other hand, presented another way to untangle the puzzle. First of all, the author capped the inventory level by a lower bound $-Q$ and an upper bound Q, that is to say $q \in \{-Q, ..., Q\}$. When inventory reaches the upper (lower) bound, the marker maker is prohibited to make any additional purchase (sale). The symmetric bound is rational as we know that market maker does not want inventory, thus q should revert around 0. As a result, Tapia introduced a system of PDEs instead of just one as in 2.17.

For $|q| < Q$:

$$\begin{aligned} 0 = \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial s^2}\sigma^2 &+ \max_{\delta^b} \lambda^b (V(t, x - s + \delta_t^b, q + 1, s) - V(t, x, q, s)) \\ &+ \max_{\delta^a} \lambda^a (V(t, x + s + \delta_t^a, q - 1, s) - V(t, x, q, s)) \end{aligned} \tag{2.29}$$

For $q = Q$:

$$0 = \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial s^2}\sigma^2 + \max_{\delta^a} \lambda^a (V(t, x + s + \delta_t^a, q - 1, s) - V(t, x, q, s)) \tag{2.30}$$

For $q = -Q$:

$$0 = \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial s^2}\sigma^2 + \max_{\delta^b} \lambda^b (V(t, x - s + \delta_t^b, q + 1, s) - V(t, x, q, s)) \tag{2.31}$$

subject to the final condition $V(T, x, q, s) = -\exp(-\gamma(x + qs))$. Rather than just the wealth process, the whole mark-to-market value of the portfolio $(x + qs)$ is factored out leaving the unknown function only the time-decaying part to consider.

$$V(t, x, q, s,) = -\exp(-\gamma(x + qs))v_q(t)^{-\gamma/k} \tag{2.32}$$

Tapia has kindly provided us the full proof of this method in his paper so we will not repeat the work here but to recall the numerical solution in 2.33.

$$\begin{aligned} \delta^{a*} &= \frac{1}{\gamma}\ln\left(1 + \frac{\gamma}{k}\right) + \frac{1}{k}\ln\left(\frac{v_q(t)}{v_{q-1}(t)}\right) \\ \delta^{b*} &= \frac{1}{\gamma}\ln\left(1 + \frac{\gamma}{k}\right) + \frac{1}{k}\ln\left(\frac{v_q(t)}{v_{q+1}(t)}\right) \end{aligned} \tag{2.33}$$

where $v_q(t)$ is an element of the vector $v(t) = \exp(-M(T-t)) \times (1, ..., 1)'$ with

$$M = \begin{pmatrix} \alpha Q^2 & -\eta & 0 & ... & ... & ... & 0 \\ \eta & \alpha(Q-1)^2 & -\eta & 0 & ... & ... & 0 \\ 0 & -\eta & \alpha(Q-2)^2 & -\eta & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & ... & ... & ... & 0 & -\eta & \alpha Q^2 \end{pmatrix}$$

where $\alpha = \frac{k}{2}\gamma\sigma^2$ and $\eta = A(1+\frac{\gamma}{k})^{-(1+\frac{k}{\gamma})}$.

From mathematical stand-point, the solution is undeniably more precise now that Tapia did not use approximation technique in the process. Despite being so, the method is critically troublesome regarding implementation in HFT. The algorithm involves matrix exponential at every time moment that market makers wants to find new optimal quotes. Usually, this should take place very shortly, it might even happen every second. Given a large inventory level bound or a long time-to-maturity exponentiating factor, it would be almost impossible to achieve the result in a blink of an eye (millisecond or nanosecond for instance). For this reason, Tapia included a section where he examined the function in infinite time horizon[8]. In this case, the formulas for δ^a, δ^b are independent of time-to-liquidation factor $(T-t)$, plus, it is needless to compute the whole matrix of $v_q(t)$.

$$\begin{aligned} \delta^{a*} =& \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) - \frac{2q-1}{2}\sqrt{\frac{\sigma^2\gamma}{2kA}\left(1+\frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}} \\ \delta^{b*} =& \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + \frac{2q+1}{2}\sqrt{\frac{\sigma^2\gamma}{2kA}\left(1+\frac{\gamma}{k}\right)^{1+\frac{k}{\gamma}}} \end{aligned} \tag{2.34}$$

The method is extensively applicable for different mid-price assumptions.

2.2.3 Levenberg-Marquardt algorithm

The algorithm was first formulated by Levenberg (1944) [12] and then by Marquardt (1963) [14]. Its goal is to find solution for a nonlinear least square problem, which is typically encountered in parametric calibration. Given a set of n pairs of data points $(x_i, y_i)_n$ and a presumed fitting function $y = f(x; \boldsymbol{\beta})$ whose parameters are a vector $\boldsymbol{\beta}$, the algorithm seeks to minimise the sum of square of errors $SSE(\boldsymbol{\beta})$ between the empirical data points and the assumed function. By changing values of $\boldsymbol{\beta}$ sequentially, the algorithm

iteratively diminishes the sum of square of residuals until the best-fit curve is found.

$$\min_{\boldsymbol{\beta}} SSE(\beta) = \min_{\boldsymbol{\beta}} \sum_{i=1}^{i=n} \Big(y_i - f(x_i; \boldsymbol{\beta}) \Big)^2 \tag{2.35}$$

We denote d as the updating vector for the parameter $\boldsymbol{\beta}$. The algorithm starts with an initial guess for $\boldsymbol{\beta}$ and d and keeps running so long as the norm of updating vector d is greater than the critical threshold ϵ. For each independent data point x_i, $\boldsymbol{J}_i$ is the first-order derivative of function $f(x_i; \boldsymbol{\beta})$ with regards to the parametric vector $\boldsymbol{\beta}$ at that point. Tt is also the $i-th$ row of the Jacobian matrix of function $f(x_i; \boldsymbol{\beta})$ with regards to the parametric vector $\boldsymbol{\beta}$.

Algorithm 1: Levenberg-Marquardt algorithm

Data: Set n pairs of data points (x_i, y_i)
Result: $\boldsymbol{\beta}$
Init: $\boldsymbol{\beta} = (1, ..., 1)'$; $d = (1, ..., 1)'$;
while $||d|| > \epsilon$ **do**
  for $i = 1$ **to** n **do**
    $\boldsymbol{J}_i = \dfrac{\partial f(x_i; \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$;
    $\hat{y}_i = f(x_i; \boldsymbol{\beta})$;
  end
  $r = y - \hat{y}$;
  $d = -(\boldsymbol{J}'\boldsymbol{J} + \lambda I)^{-1}\boldsymbol{J}'r$;
  $\boldsymbol{\beta} = \boldsymbol{\beta} + d$;
end

The damping factor λ marks the difference between this and other least-square algorithm. When $\lambda = 0$, the algorithm is identical to Gauss-Newton method. Its existence is to protect the algorithm to reach singularity for $\boldsymbol{J}$.

Chapter 3

Data preparation

3.1 Data description

Having discussed the methodology, we arrive at the stage where necessary information is identified for implementation. Two pieces of data are required: (i) the mid-price trajectory S_t and (ii) the volume of market orders for each price N_t. The Bitmex exchange publicly offers historical data dating back to 2014. Transaction details are stored daily and are accessible at `https://public.bitmex.com/?prefix=data/`. The exchange houses two sets of data that are limit order quotes and trading logs (or market order)[1]. As market and limit orders are different in nature, their data structures are disparate. Table 3.1 and 3.3 list out all the information fields and their meanings.

Quote data table preserves the position of order book periodically. Each row represents a snapshot of the order book at one moment.

Table 3.1: Quote data table

	Field name	Type	Description
1	timestamp	datetime	The moment at which the limit order book position is recorded, with precision to millisecond.
2	symbol	string	Ticker representing quoting instrument.
3	bidSize	integer	Total amount of contract at each bid price level
4	bidPrice	float	Lowest bid price
5	askSize	integer	Total amount of contract at each ask price level
6	askPrice	float	Highest ask price

[1]From this point onward, we refer them as "quote" and "trade" to be coherent with the source.

Table 3.2: Quote data desciptive statistics

	bidSize	bidPrice	askSize	askPrice
count	153,632,023	153,632,023	153,632,023	153,632,023
mean	510,475	10,201.1	490,938	10,201.8
std	671,002	1,373.97	589,888	1,374.04
min	1	7,435	1	7,438
25%	67,753	9,286.5	63,253	9,287
50%	326,359	10,335	303,473	10,336
75%	805,455	11,278	789,317	11,278.5
max	42,870,290	13,919.5	41,882,526	13,920

Trade data table records every single market order that takes place. Each row marks a request arriving at Bitmex and obviously it is possible that more than one order reach the exchange at the same time.

Table 3.3: Trade data table

	Field name	**Type**	**Description**
1	timestamp	datetime	The moment at which the transaction occurs, with precision to millisecond.
2	symbol	string	Ticker representing trading instrument, we only concern the ticker XBTUSD.
3	side	string	The side of the trade, either Buy or Sell.
4	size	integer	Contract size of the trade
5	price	float	The price that the trade was executed

Table 3.4: Trade data descriptive statistics

	price	size
count	20,150,159	20,150,159
mean	10,460.3	13,629.9
std	1,390.62	50,565.4
min	7,435	1
25%	9,520.5	250
50%	10,623.5	1,604
75%	11,455.5	9,000
max	13,920	20,000,000

3.2 Data pre-processing

Given the immense size of tick-by-tick transaction data as well as the restraint of our computer capacity, we limit the time frame of study from June 1^{st} to July 31^{st} 2019 or 61 trading days in total. As perpetual contract is our prime trading asset, we filter out irrelevant tickers from the data set. The instrument is labelled XBTUSD on Bitmex.

The purpose of quote data table is to track the mid-price evolution S_t, it is not given explicitly though. Instead, S_t is infered from bidPrice and askPrice:

$$\text{MidPrice} = \frac{\text{bidPrice} + \text{askPrice}}{2}$$

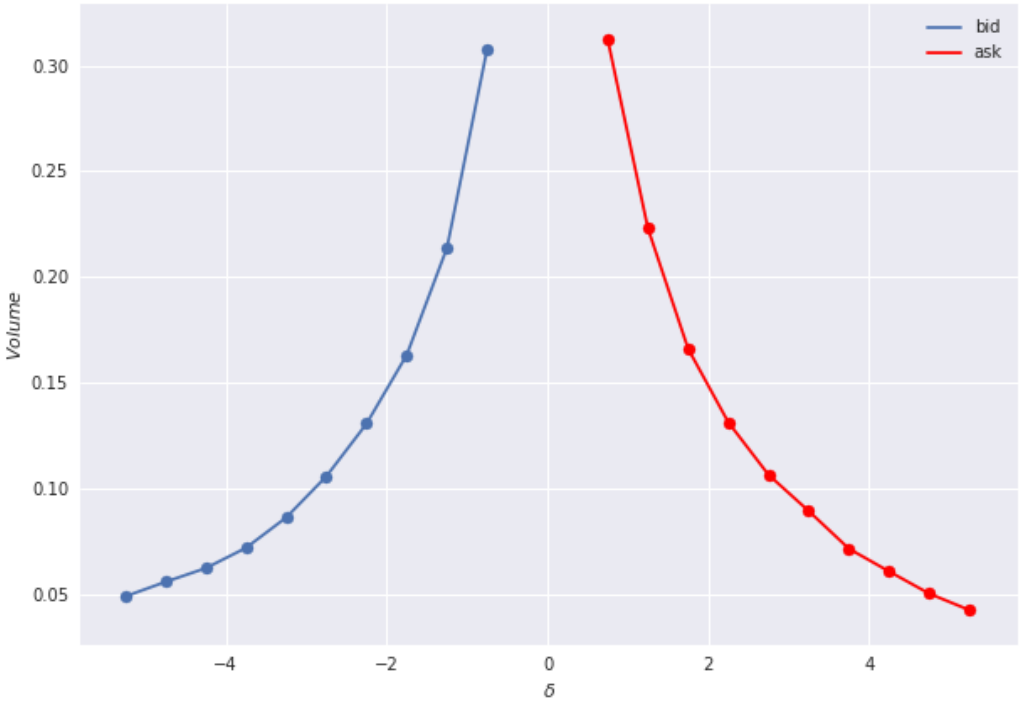
Due to the recurrent feature of quote data table, there exists duplicate entries for mid-price. We therefore remove the redundant ones in order to cut down table size without loss of information. All the consecutive rows where mid-price remains unchanged are discarded from the data table. Thus, the final quote data table shows only the changes of mid-price.

Regarding the trade data, we are indifferent about the arriving orders so long as they occur at the same time and price (of course each side is treated separately). Consequently, all those trades are aggregated to one: the trading volume is the summation of those individual trades.

$$\text{Trading volume}^a(t, t+\Delta T, S) = \sum_{t}^{t+\Delta T} \text{size } \mathbf{1}_{\text{price} = S} \mathbf{1}_{\text{side = Sell}}$$

Figure 4.4 justifies the assumption about the exponential relationship between the quoting price and possibility of execution.

Figure 3.1: Relationship between distance to midprice and volume of trade



Chapter 4

Calibration

The last few parts of our research have been dealing with theoretical part of market making model. We are now aware of possible mathematical solutions as well as how to derive them. The remaining task concerns the practical side of the model; this chapter is meant for empirical study given the dataset from chapter 3. The solutions of the Avellaneda-Stoikov model as in 2.28 and its expansions 2.33, 2.34 are characterised by four parameters: (i) the risk aversion coefficient γ, (ii) the volatility of the mid-price process σ and (iii, iv) the two parameters that measure the trading intensity A and k. The risk aversion coefficient γ represents the risk-averse level of market makers so it varies among individuals, apparently it can be arbitrary. As a result, the remaining parameters to be calibrated are σ, A and k. We start with a step-by-step guideline for calibration process:

1. Mid-price model selection and calibration: We need to choose a model that suits the mid-price process, possible options are: diffusion process (with or without drift), diffusion process with adverse selection, Ornstein-Uhlenbeck process, etc.. Aside from the volatility parameter σ, each model assumption contains other distinctive parameters and will be discussed in detail later on.
2. Trading intensity $\Lambda(\delta)$ estimation: The intensity is different for each level of distance from the mid price δ as portrayed in its formula[1].
3. Calibration for the trading intensity parameters A and k using the estimate $\hat{\Lambda}(\delta)$: Using logarithmic transformation, we modify the exponential relationship to a simpler linear relation.

[1]Refer to equation 2.10.

4.1 Methodology review

This section is dedicated to convert mathematical procedure to algorithmic one, relevant mathematical explanation will be presented in case of necessity.

4.1.1 Calibration of the mid-price model

As mentioned in the literature review, we understand that the mid-price process theoretically can take any assumption on its evolution. The hypothesis shows our belief on certain feature of the price dynamic. In this section, we focus on two prominent models: (i) the classical diffusion price model (one-dimensional Brownian motion) and (ii) the adverse selection model (diffusion process with jumps). We chose these two because of their distinctive calibration method.

Diffusive model

We recall the assumption proposed by Avellaneda & Stoikov (2008) [3]. They theorised that the price process can be modelled by a Brownian motion in extremely short-term:

$$\mathrm{d}S_t = \sigma \mathrm{d}W_t \tag{4.1}$$

This formula suggests the relationship between σ and $\mathrm{d}S_t$:

$$\begin{aligned} Var[\mathrm{d}S_t] &= E[(\mathrm{d}S_t)^2] - E[\mathrm{d}S_t]^2 \\ &= E[\sigma^2 \mathrm{d}t] - 0 \\ &= \sigma^2 \mathrm{d}t \end{aligned} \tag{4.2}$$

$$\Rightarrow \sigma^2 = \frac{Var[\mathrm{d}S_t]}{\mathrm{d}t} \tag{4.3}$$

As a result, the estimation of the model will be the same as the classical volatility estimator:

$$\hat{\sigma^2}_n = \frac{1}{n\mathrm{d}t} \sum_{k=1}^{n} (S_{t_k} - S_{t_{k-1}})^2 \tag{4.4}$$

Adverse selection model

This model is an extension of the classic diffusion process [7][5]. Other than the diffusive term, the price process is added a new component that represents market instantaneous impact due to MOs. The newly added item is the combined outcome of

two jump processes that model selling $\xi^a \mathrm{d}N_t^a$ and buying $\xi^a \mathrm{d}N_t^b$ MOs during unit time interval. With the inclusion of the market impact, the mid price is modelled as follows:

$$\mathrm{d}S_t = \sigma \mathrm{d}W_t + \xi^a \mathrm{d}N_t^a - \xi^a \mathrm{d}N_t^b \tag{4.5}$$

If we consider selling and buying acts grant similar effect to the mid-price process, or $\xi^a = \xi^b$, then equation 4.5 can be rewritten as:

$$\begin{aligned} \mathrm{d}S_t &= \sigma \mathrm{d}W_t + \xi \mathrm{d}N_t^a - \xi \mathrm{d}N_t^b \\ &= \sigma \mathrm{d}W_t + \xi(\mathrm{d}N_t^a - \mathrm{d}N_t^b) \end{aligned} \tag{4.6}$$

When we consider the historical trading data in chapter 3, $\mathrm{d}N_t^a$, $\mathrm{d}N_t^b$ and $\mathrm{d}S_t$ are obviously given. They are respectively the additional sales, purchases filled in the order book and the first-difference of the mid-price process every unit time $\mathrm{d}t$. Therefore, we can estimate ξ and σ by considering a linear regression as follows:

$$\mathrm{d}S_t = \beta_0 + \beta_1(\mathrm{d}N_t^a - \mathrm{d}N_t^b) + \epsilon_t \tag{4.7}$$

Then β_1 is the estimate for ξ and β_0 should be very close to 0 as we expect no trending effect in this model. The residual term ϵ_t is inherited from $\sigma \mathrm{d}W_t$ whose expectation is 0 and variance is $\sigma^2 \mathrm{d}t$. We thence can estimate σ from the residuals ϵ_t:

$$\hat{\sigma^2}_n = \frac{Var[\epsilon]}{\mathrm{d}t} = \frac{1}{n\mathrm{d}t} \sum_{k=1}^{n} \epsilon^2 \tag{4.8}$$

4.1.2 Estimation methods for trade intensity $\Lambda(\delta)$

We enter the second step of where we have to find an estimate for trade intensity $\Lambda(\delta, 0, t)$. This parameter bespeaks expected number of market orders land at the exchange in a time interval $[0, t]$; it varies accordingly to the distance to midprice δ. The parameter regulates the behaviour of Poisson process $N_t^{a,b}$.

$$N_t \sim \mathcal{P}(\Lambda(\delta, 0, t))$$

According to Tapia (2015), there are two approaches to solve this calibration problem and they will be the subject of discussion in the following parts of of our study. As mentioned earlier in this chapter, the intensity is highly dependent on distance to midprice δ so it is a must that we estimate it for every δ independently.

Estimate by counting trades

One way to estimate $\Lambda(\delta, 0, T)$ is to add up the number of trades that were executed at that very level of δ during the time $[0, T]$. Because trades are easily observable on Bitmex, here we take the role of an external observer who can detect every deal happened in each unit time of our observation window. The unit time is resulted from the discretisation of $[0, T]$ into n units; each unit considers the time length of ΔT. We denote X_k as the total volume of trades that arose in the period k-th of $[0, T]$ for $k \in \{1, 2, ..., n\}$, or the interval $[(k-1)\Delta T, k\Delta T[$. We end up with n observations; the intensity is estimated by the mean of all of those:

$$\hat{\Lambda}_n(\delta) = \frac{1}{n} \sum_{k=1}^{n} X_k \tag{4.9}$$

This is an unbiased estimator that converges almost surely to the true value of $\Lambda(\delta, 0, T)$ by strong law of large number.

Estimate using waiting time

The other approach to find the estimator $\hat{\Lambda}(\delta, 0, T)$ is gauging the time-to-execution of orders at the distance δ. The approach exploits the relationship between Poisson distribution and exponential distribution in queuing theory. The waiting time is defined as $X_k := \min(\tau_n, T)$ where τ_n is an exponential distributed random variable associated to the parameter $\Lambda > 0$ that we are trying to estimate.

$$\tau_n \sim \mathcal{E}xp(\Lambda)$$

X_k is the k-th observation in the time unit $[(k-1)\Delta T, k\Delta T[$[2]. The estimator for the intensity Λ in this case is calculated by dividing the total odds that the waiting time is less than T, or the order is executed before reaching maturity, to the total waiting time.

$$\hat{\Lambda}_n(\delta) = \frac{\sum_{k=1}^{n} \mathbf{1}_{\{X_k < T\}}}{\sum_{k=1}^{n} X_k} \tag{4.10}$$

This approach has an advantage over the previous one under circumstances that we are not authorised to see other transactions in the exchange therefore can construct the estimator based on merely our observable trading history.

[2]We use the same discretisation principle as in the previous approach

4.1.3 Calibration for A and k

After the intensity $\Lambda(\delta)$ is estimated, the only remaining task is to calibrate the two parameters A and k which defines the intensity function. Results from section 4.1.2 are prerequisites of this calibration process. We consider the intensity for bid and ask transactions in the interval $[0, \Delta T]$:

$$\Lambda^a(\delta, 0, \Delta T) = \int_0^{\Delta T} Ae^{-k\delta + k(S_{u+\Delta T} - S_0)} du \tag{4.11}$$

and

$$\Lambda^b(\delta, 0, \Delta T) = \int_0^{\Delta T} Ae^{-k\delta - k(S_{u+\Delta T} - S_0)} du \tag{4.12}$$

If we assume the same parameters A and k for both bid and ask trading intensity then , we can combine the equation 4.11 and 4.12 into one function $\Lambda(\delta)$. This eases the calibration process.

$$\Lambda(\delta, 0, \Delta T) = \int_0^{\Delta T} Ae^{-k\delta + k(S_u - S_0)} du \tag{4.13}$$

As a result,

$$\begin{aligned} E[\Lambda(\delta, 0, \Delta T)] &= E\left[\int_0^{\Delta T} Ae^{-k\delta + k(S_u - S_0)} du\right] \\ &= Ae^{-k\delta} \int_0^{\Delta T} E[e^{k(S_u - S_0)}] du \end{aligned} \tag{4.14}$$

Taking logarithm of both sides we have:

$$\log(E[\Lambda(\delta, 0, \Delta T)]) = \log(A) - k\delta + \log\left(\int_0^{\Delta T} E[e^{k(S_u - S_0)}] du\right) \tag{4.15}$$

From section 4.1.2, we have $\hat{\Lambda}(\delta)$ as the estimate of $E[\Lambda(\delta, 0, \Delta T)]$ for each $\delta \in \{\nu, ..., j_{\max}\nu\}$ where ν is the minimum change in price. Substituting the estimate into equation 4.15 gives us:

$$\log(\hat{\Lambda}(\delta)) = \log(A) - k\delta + \log\left(\int_t^{t+\Delta T} E[e^{k(S_u - S_t)}] du\right) \tag{4.16}$$

The simplest method to estimate A and k from the equation 4.16 as suggested by Laruelle (2013) [11] is to apply linear regression to the function; we consider:

$$\log(\hat{\Lambda}(\delta)) = \beta_0 + \beta_1 \delta + \epsilon \tag{4.17}$$

Then $A \approx e^{\hat{\beta}_0}$ and $k = -\hat{\beta}_1$. In this way, the calibration technique have neglected the last part of equation 4.16 so the result is not the closest estimate.

A more precise method to calibrate for A and k is to minimise the sum of squares of residuals:

$$SSE(A,k) = \sum_{j=1}^{j_{\max}} \left(\log(\hat{\Lambda}(\delta_j) + k\delta_j - \log(A) - \log\left(\int_0^{\Delta T} E[e^{k(S_u - S_0)}]du \right) \right)^2 \tag{4.18}$$

There are several optimisation methods that can be utilised but here we choose the method of Levenberg-Marquardt as the minimiser. As for this method, the Jacobian matrix is required and it will be different depending on the choice of mid price model.

Let us take the classic model of Avellaneda & Stoikov (2018) [3] where the mid price is modelled by a Brownian motion:

$$\mathrm{d}S_t = \sigma dW_t \tag{4.1}$$

Then the integral part of the equation 4.18 can be solved as follows:

$$\begin{aligned} \int_0^{\Delta T} E[e^{k(S_u - S_0)}]du &= \int_0^{\Delta T} E[e^{k\sigma W_u}]du \\ &= \int_0^{\Delta T} e^{\frac{k^2\sigma^2}{2}u}du \\ &= \frac{2}{k^2\sigma^2}(e^{\frac{k^2\sigma^2}{2}\Delta T} - 1) \end{aligned} \tag{4.19}$$

As a result, the residual function is as follows:

$$res(A,k) = \log(\hat{\Lambda}(\delta_j) + k\delta_j - \log(A) - \frac{2}{k^2\sigma^2}(e^{\frac{k^2\sigma^2}{2}\Delta T} - 1) \tag{4.20}$$

The Jacobian matrix in this case will be a $(2 \times j_{max})$ matrix whose j-th row is defined as:

$$\mathbf{J}(j,A) = \frac{\partial res}{\partial A} = -\frac{1}{A} \tag{4.21}$$

$$\mathbf{J}(j,k) = \frac{\partial res}{\partial k} = \delta_j + \frac{2}{k} - k\sigma^2\Delta T\left(1 + \frac{1}{e^{k^2\sigma^2\Delta T/2}}\right) \tag{4.22}$$

The last stage is to employ algorithm 1 to find the estimates for (A,k).

4.2 Calibration results

Coming to this part of the study, we finally apply the calibration method as previously mentioned to the dataset of Bitcoin that we have collected. The data-set consists of the Level I order book data (price and volume at the best bid and best ask level) and the tick-by-tick trading data in the Bitmex exchange from 01/06/2019 to 31/07/2019. .

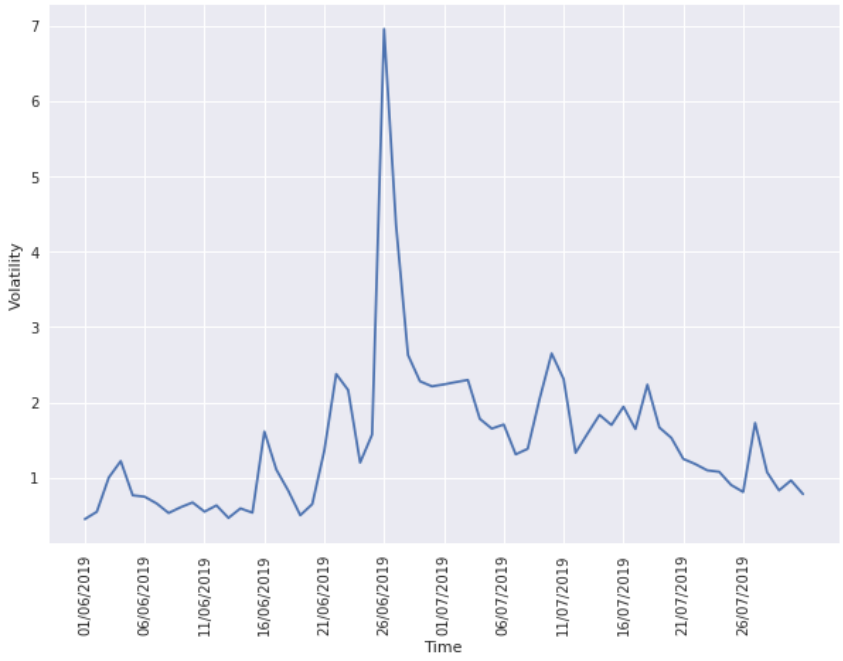
4.2.1 Calibration of volatility

We start with the traditional model of Avellaneda Stoikov where the mid price follows the Brownian motion. The volatility in this case as we have analysed previously will be estimated by the formula:

$$\hat{\sigma^2}_n = \frac{1}{n\Delta T}\sum_{k=1}^{n}(S_{t_k} - S_{t_{k-1}})^2 \tag{4.4}$$

This basically means that the volatility is basically the variance of the difference in mid price divided by ΔT. In our calibration model, we choose ΔT to be equal to 1 second and also take 1 second as the unit of time in our whole study. Volatility will be estimated day-by-day so we have $n\Delta T = 86,400$ as it represents the number of seconds in one day. The mid price data as obtained after the data pre-processing procedure are resampled into the interval of 1 second. The results of our calibration is as follows:

Figure 4.1: Daily Volatility-Avellaneda model



From the figure 4.1, we can see the evolution of the volatility in time. Before the middle of June, the secondly volatility of each day is only around 1 to 2 USD per second. However, as the price of the Bitcoin peaked during the middle of June and that there are a lot of volume during the period, the volatility of the mid price can go up to 7 USD per second. The volatility in this case therefore is clearly not a constant and there are several statistical models that can be applied to this problem. However, due to the scope of our study, the topic of volatility modeling is not covered in our research.

Next, we also try to calibrate for the volatility with the inclusion of adverse selection where the mid price follows:

$$\mathrm{d}S_t = \sigma \mathrm{d}W_t + \xi(\mathrm{d}N_t^a - \mathrm{d}N_t^b) \tag{4.6}$$

Here $\mathrm{d}N_t^a$ and $\mathrm{d}N_t^b$ are the number of trades in each side that happened during every ΔT and can be estimated by their definition. As $\mathrm{d}N_t^a$ and $\mathrm{d}N_t^b$ are estimated, we then used the method of linear regression without constant to estimate the remaining parameter of ξ and σ where ξ equal to the coefficient β_1 of the regression and σ can be estimated by the remaining residual of the regression. The calibrated volatility under this model when compared to the model without market impact is as follows:

Figure 4.2: Comparing results of two volatility model

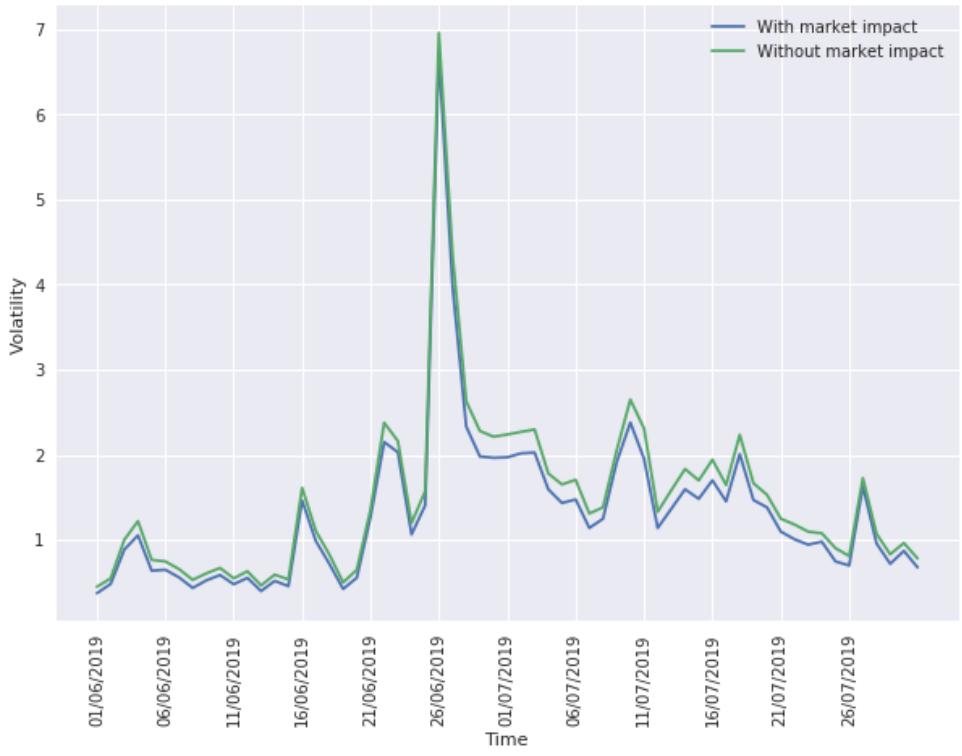


Figure 4.2 shows the comparison of the two volatility calibrated under the two models of mid price. Despite the difference between the two model, the calibrated volatility are quite similar. With the inclusion of the market impact factor, the volatility of the adverse selection model is only slightly lower than the standard model and they are highly correlated. In conclusion, based on the dataset the we used, the result suggests that there is not much improvement including the market impact factor into the model.

4.2.2 Estimation of trading intensity $\Lambda(\delta)$

In the previous part of this sections, we have discussed two popular methods that can be utilised to estimate of the trading intensity which based on the counting of trade or based on the waiting time until the orders are executed. In our study, we mainly focus on the first method where we count the number of trade that has happened during each ΔT. The estimator of the trading intensity $\Lambda(\delta)$:

$$\hat{\Lambda}_n(\delta) = \frac{1}{n}\sum_{k=1}^{n} X_k \tag{4.23}$$

where: X_k is the observation of trade during the period ΔT. Using the method discussed previously, we estimate $\hat{\Lambda}_n(\delta)$ for each day for a fixed list of $\delta \in \{-k\Delta S, (-k+1)\Delta S, ..., k\Delta S\}$ where ΔS is the minimum difference in price that can be posted (for Bitmex, $\Delta S = 0.5$ for the price of Bitcoin).

Algorithm 2: Algorithm to estimate $\Lambda(\delta)$

```
Result: Λ̂(δ)
Defining a list of δ
Resampling Midprice for ΔT = 1s
foreach δ do
    foreach Midprice S_t do
        if δ > 0 then
            Λ̃=Sum all trades at price ≥ S_t + δ between t and t + ΔT
        else
            Λ̃=Sum all trades at price ≤ S_t + δ between t and t + ΔT
        end
    end
    Λ̄ = mean of Λ̃
end
Λ̂(δ) = list of Λ̄ at each δ
```

Figure 4.3: Daily trading Intensity

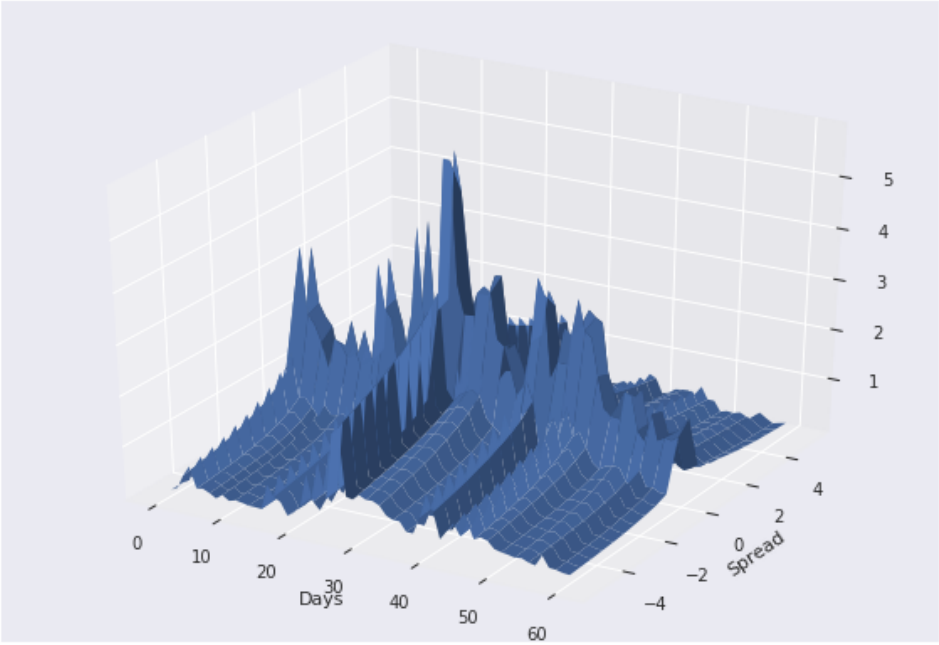


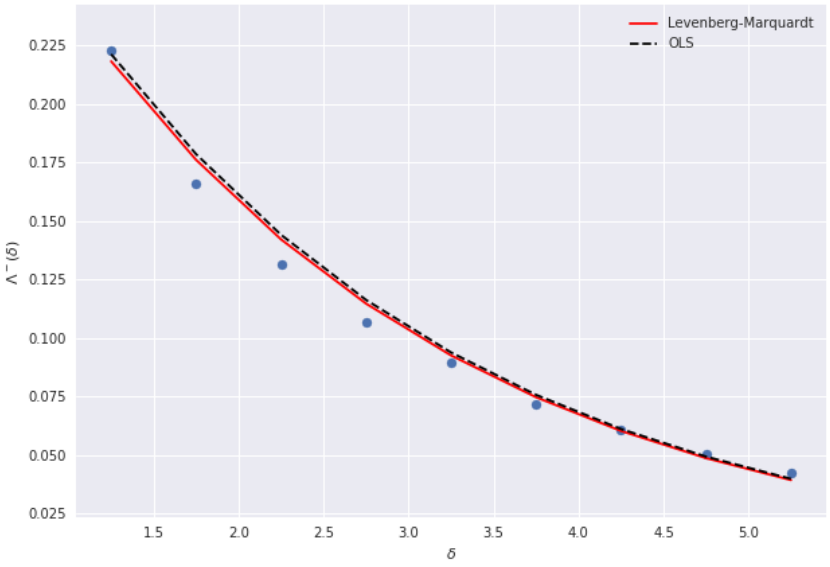
Figure 4.3 shows the daily estimated $\Lambda(\delta)$ from the Bitcoin historical data during 61 days from the beginning of June to the end of July with the following parameters: $\Delta T = 0$, $\delta \in \{-5.25, -4.75, ..., 5.25\}$ and $n = 86,400$ as the number of seconds in one day.

After the approximation of $\Lambda(\delta)$, the remaining task is to calibrate for the parameters A and k. As discussed previously, there can be several methods than can be used to solve the calibration problems. In this study, we focus on the two methods: one method is the simple linear regression and the other is the Levenberg-Marquardt algorithm which can be considered as one of the best optimisation methods for non-linear least square problem. Both methods will be used to solve the function (4.16). The results of the calibration for A and k by both method is demonstrated in figure 4.4a and 4.4b. From the two figures, it can be quite easy to realise that the results are not much different between the two methods. The parameter k in this case for both method is estimated to be 0.397. Regarding the parameter A, the estimated values are slightly different as the

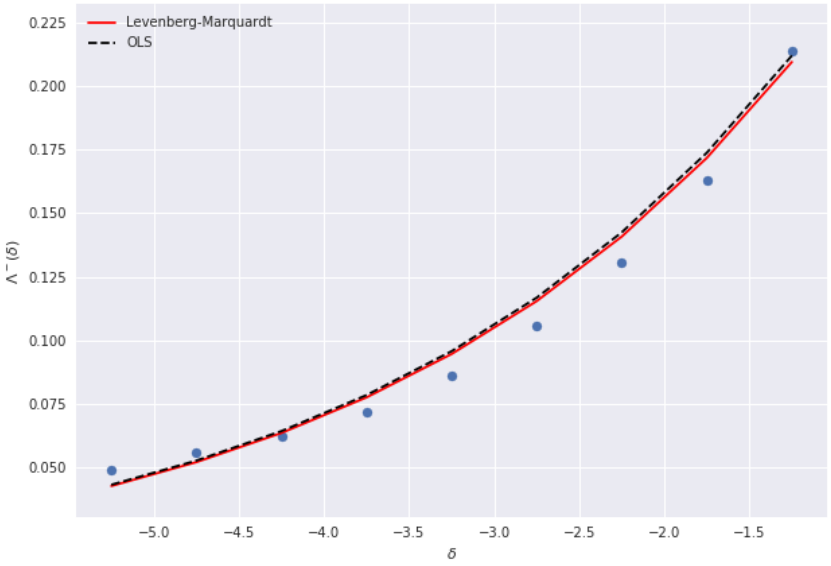
OLS method gave 0.34 whereas the Levenberg-Marquardt gave 0.3441.

Figure 4.4: Calibrated Filled rate for one day 01/06/2019.

(a) Ask side



(b) Bid side



As both methods try to minimise the sum of squares of errors (SSE) with the same number of variables, it is possible to compare between the two. The SSE of the Levenberg-Marquardt is 0.0889 in comparison with 0.0903 of the OLS. From the result of the SSE, we can see that the Levenberg-Marquardt approach is slightly better than the OLS. This is as expected due to the fact that the approach of Levenberg-Marquardt still takes into account the integral $\int_0^{\Delta T} E[e^{k(S_u - S_0)}]du$. The OLS method, on the other hand, try to ignore the integral so that the problem can be solved linearly. Despite being slightly better in result, the calculation of the Levenberg-Marquardt can be more problematic than the OLS, especially in the case of more complex mid price model where the integral of $\int_0^{\Delta T} E[e^{k(S_u - S_0)}]du$ has not closed form solution. If the complexity of the model is taken into consideration then the OLS method is likely to be preferable due to its simplicity.

Chapter 5

Backtesting

Before launching any trading strategy, we need to check whether it is applicable in practice. This verification process is called backtesting and its outcome justifies the use of the trading strategy. The backtesting procedure involves feeding historical data to the trading strategy to find out how well or how badly the strategy would have worked should the past rewound. Even though the past cannot tell the future, it is still a powerful tool to analyse the feasibility of the strategy.

5.1 Backtesting Methodology

Backtesting is performed on the dataset as we described in chapter 3, from 01/06/2019 to 31/07/2019. We first uncover significant parametric setting for this part. More specifically, we will unveil all the aspects that market makers encounter and are required to settle when putting market making strategy into practice.

The first element is all about quoting prices. The last chapter has clarified how to calibrate mandatory parameters in market making model and this part adds some details on the implementation in HFT. At the beginning of a day, we use the historical data of the previous day to calibrate the parameters σ, A and k using the process we have mentioned in chapter 4. They serve as the inputs to deduce optimal spread $\delta^{a,b}$ for the whole day so they will be kept constant during this trading day. The parameters can only be updated at the beginning of the next day. In this manner, we avoid look-ahead bias, which means using future information to predict at current time. Two other components in determining quoting prices are the midprice S_t and inventory level q. They should be the latest recorded values so every time new order prices should be set, we recalculate

these two factors. In this study, we assume that there is no delay in getting the current mid-price and posting the orders to the exchange so the input mid price S_t will be the mid price at that very moment. The buying and selling quotes are δ^b lower and δ^a higher than this current mid-price. The remaining input for the optimal spread is the risk aversion coefficient γ. The parameter γ is the parameter for the utility function and it represents the risk aversion of the investors. This parameter is arbitrary and in this study, we fix this parameter as $\gamma = 0.001$.

The frequency and volume of trades are the second aspect to consider. The former relates to how often market makers post limit orders and the latter is about the size of trades that market makers intend to go on. Updating time is the duration between two consecutive limit orders. Operating the strategy in HFT implies that this interval should be very small. For this reason, we scheduled that new quotes are updated every second. Furthermore, this choice agrees with ΔT that we used to calibrate our parameters in chapter 4. The other facet, order size, calls for delicate attention. Too small order size is easily filled entirely so it does not reflect well market operation. In fact, we cannot neglect the existence of partially filled orders. Furthermore, small order size is not favoured by big investors whose capital is hefty. On the other hand, too big order size is likely to influence the market in some way because market participants can observe it and this may even distort the predictability of the model. Having considered these two effects, here we limit the maximal order size by 0.1 XBT and it is also the first order size. This amount is neither too small nor too big; it falls in the third quartile of order size distribution[1]. When orders are partially filled, our decision on order size in the upcoming second (when updating limit orders) will be affected but not in the very current second. The next order size is determined as the greater between maximum order size (0.1 XBT) and the absolute value of the current inventory q. This means that the inventory will never surpass the maximum order size and the inventory risk will be minimised at all cost.

Thirdly, we cannot neglect one main contribution to the profit of market makers, which is the rebate fee. The fee is the money that market makers are given instead of being charged when using limit orders. It is to encourage them to post limit orders thus furnishing more liquidity to the market. The Bitmex exchange fixs the rebate fee at 0.025% of the amount was executed, i.e. we are rewarded 0.025% of the executed orders.

Execution priority is a very eminent factor that we save for the last so as to emphasise its importance. This term refers to two executing rules: (i) orders are executed sequen-

[1]Refer to Table 3.3

tially along the depth of the limit order book and (ii) orders of more than one market makers compete with another to win the privilege to be executed first. Because our study focuses on Level I of limit order book, the former phenomenon hardly affects our strategy. Thus, we mainly concern the latter one. It is intuitively understandable that first-priority market maker is the most advantageous because his/her orders are almost surely filled (either entirely or partially) so long as a price match exists. This reduces vastly inventory risk and ensures at least some gain for the market maker. Vice versa, unprioritised orders are inferior thus stipulating lower chance to be executed. As a result, we backtest and study two versions of the strategy. One version where we have the first priority and the other where we have no priority at all. First-priority assumption permits our orders to be executed when there are market orders whose price match current or deeper level of the limit order book whilst no priority assumption forbids doing so at the current level.

Algorithm 3: Market making algorithm

Input : $\sigma, k, A, T, t, \Delta T$, size

while $t < T$ **do**

Look up for the last mid price S_t

Calculate the optimal δ^a, δ^b for the next trade[a]

buy size = max(size,abs(q))

sell size = max(size,abs(q))

foreach $MO \in [t, t + \Delta T]$ **do**

if *MO_type = sell and* $S_t - \delta^b \geq$ *MO_price and* $q < Q$ *and buy size* > 0 **then**

if *buy size* $\leq$ *MO_size* **then**

$PnL = PnL + buy\ size - buy\ size \times (S_t - \delta^b) \times (1 - rebate)$

$buy\ size = 0$

break

else

$PnL = PnL + MO\_size - MO\_size \times (S_t - \delta^b) \times (1 - rebate)$

$buy\ size = buy\ size - MO\_size$

continue

end

else if *MO_type = buy and* $S_t + \delta^a \leq$ *MO_price and* $q > -Q$ *and* *sell size* < 0 **then**

if *sell size* $\geq$ *MO_size* **then**

$PnL = PnL - sell\ size + sell\ size \times (S_t + \delta^a) \times (1 + rebate)$

$sell\ size = 0$

break

else

$PnL = PnL - MO\_size + MO\_size \times (S_t + \delta^a) \times (1 + rebate)$

$sell\ size = sell\ size - MO\_size$

continue

end

else

$t = t + \Delta T$

end

end

end

[a]Depending on the choice of solution formula, we substitute different equations in this

5.2 Backtesting results

This section opens with the naive market making strategy. The strategy will serve as the benchmark so that we can compare our strategy with. We list out common indicators to gauge strategy performance: Sharpe ratio, profit factor, maximal drawdown, so on and so forth.

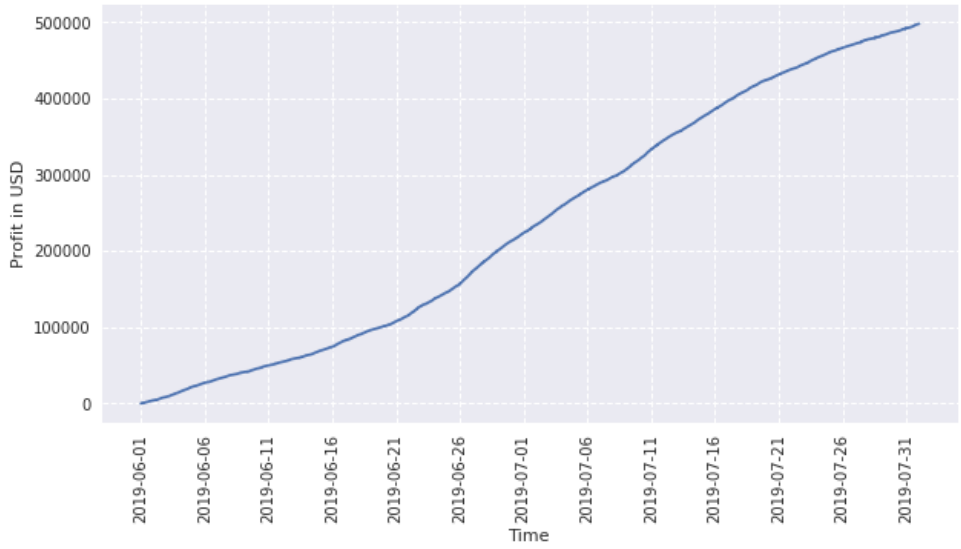
The naive strategy is a trading plan in which market makers quote prices at minimal distance from the mid-price. Usually, this distance is one half of a tick size and is 0.25 USD on Bitmex exchange as for XBT[2]. Implementing this trading scheme means that at every updating time, we post new limit orders at 0.25 USD further away (lower for bid and higher for ask) from the mid-price. As having mentioned in the previous part, we examine the prioritised vs. unprioritised trading patterns to see its effect on the strategy.

Figure 5.1a visualises the cumulative profit of naive strategy with first-priority during the backtesting period. The non-decreasing trend implies that this strategy conceals no loss during the period. Figure 5.1b tells the same story since we spot only positive bars.

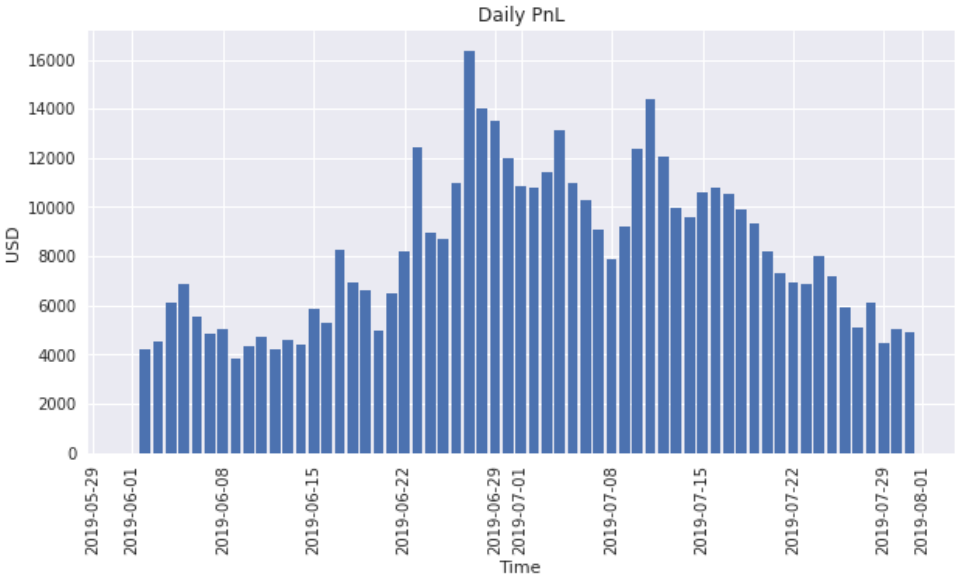
[2]Tick size on Bitmex is 0.5 USD

Figure 5.1: P&L (in USD) of naive strategy with first-priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit



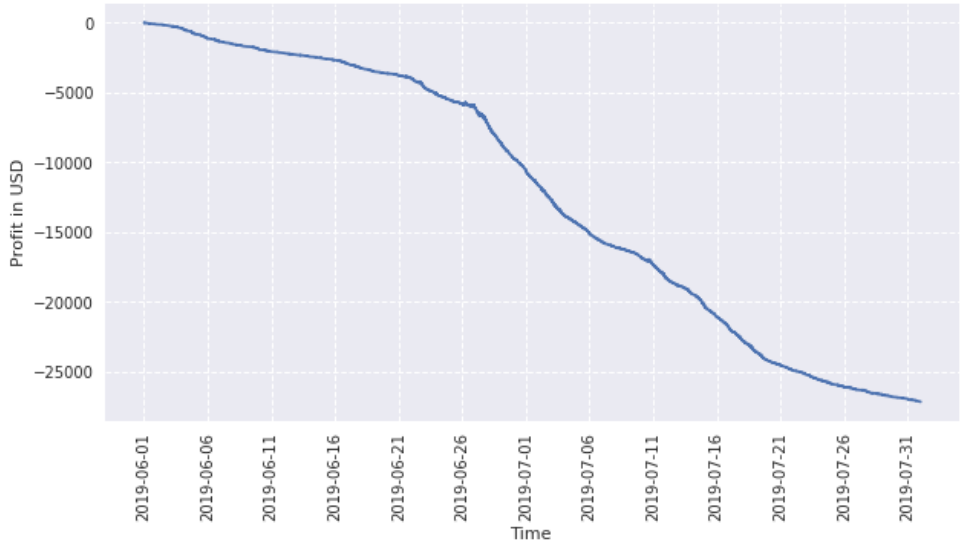
(b) Daily profit



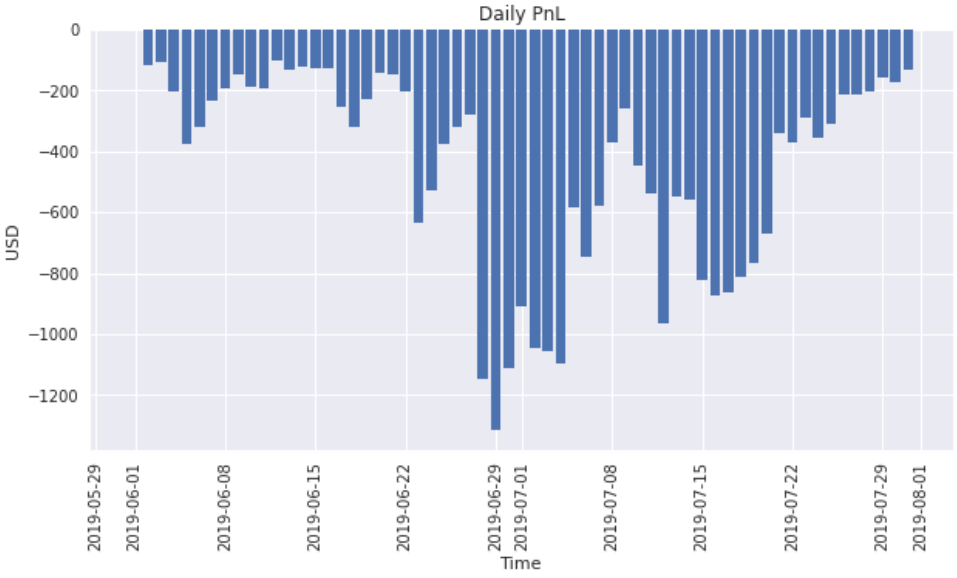
Nevertheless, the priority assumption is almost irrealistic for the majority of investors. This assumption is only satisfied for certain investors with extreme technology advance. We therefore compare with the case in which we are inferior in the executing hierarchy. Figure 5.2a is almost the mirror path of the last strategy; we cannot detect signs of winning for the whole period. Figure 5.2b also illustrates daily loss of this strategy.

Figure 5.2: P&L (in USD) of naive strategy with no priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit



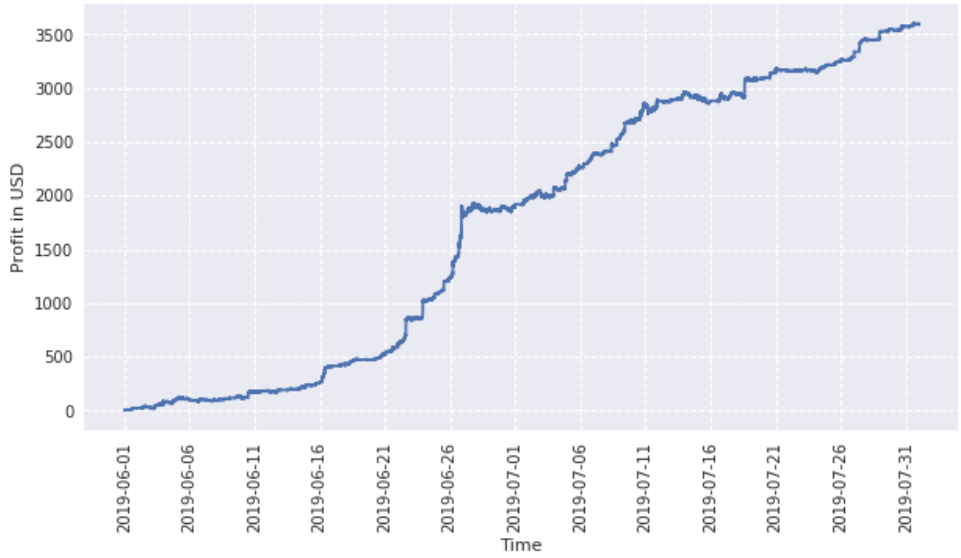
(b) Daily profit



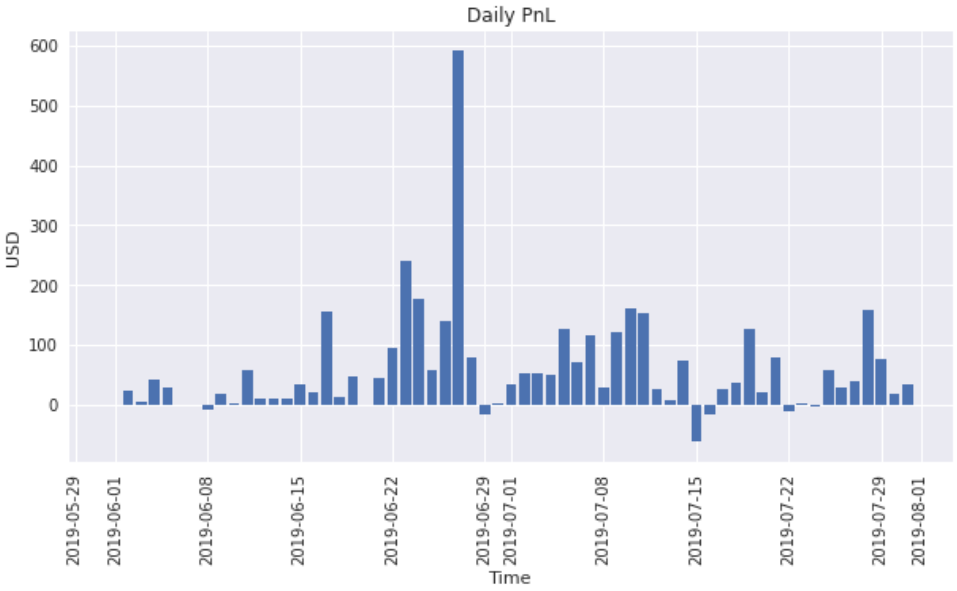
Avellaneda-Stoikov model does not outperform the naive strategy under the first-priority assumption. We do lose in some days. Even so, the rising trend of cumulative profit suggests overall winning position.

Figure 5.3: P&L (in USD) of Avellaneda-Stoikov strategy with first-priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit



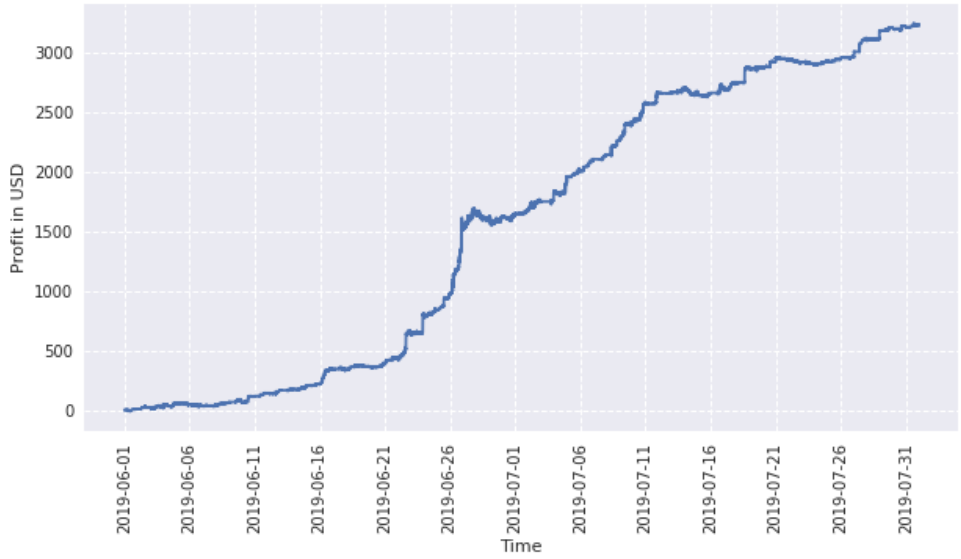
(b) Daily profit



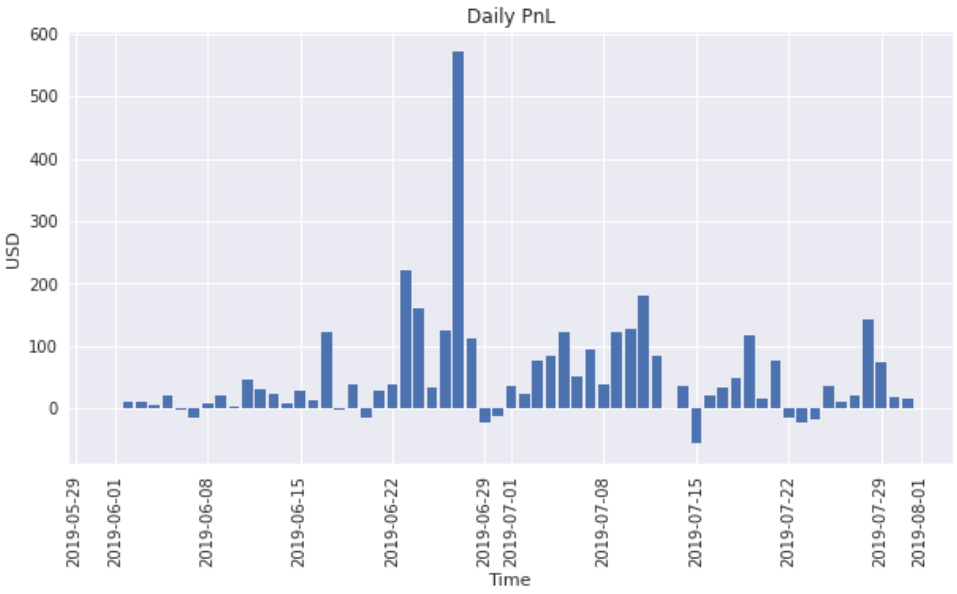
While Avellaneda-Stoikov model cannot justify itself under first-priority assumption, it does so in worse case when market makers have no privilege in executing order. That Figure 5.4 resemble Figure 5.3 hints that execution priority has minute or no effect on our strategy. As a result, the strategy is more robust and applicable to a wider range of investors, i.e. to those with no executing privilege.

Figure 5.4: P&L (in USD) of Avellaneda-Stoikov strategy with no-priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit



(b) Daily profit



Before coming to next part, we need to have some words on the solutions revealed by Tapia. From equation 2.33 we can visualise the algorithm to obtain $\delta^{a,b}$.

Algorithm 4: Algorithm to calculate optimal $\delta^{a,b}$ from the solutions by Tapia

Result: δ^a, δ^b
Input : $A, k, \gamma, \sigma, Q, T, t$
Init: matrix M of size $\big((2Q+1) \times (2Q+1)\big)$
$\alpha = \frac{k\gamma}{2}\sigma^2 \quad \eta = A\big(1+\frac{\gamma}{k}\big)^{-\big(1+\frac{k}{\gamma}\big)}$
for $i = 0$ **to** $2Q+1$ **do**
  for $j = 0$ **to** $2Q+1$ **do**
    if $i = j$ **then**
      $M[i,j] = \alpha(Q-i)^2$
    else if $j = i-1$ *or* $j = i+1$ **then**
      $M[i,j] = -\eta$
    else
      $M[i,j] = 0$
    end
  end
end
$v = e^{-M(T-t)} \times (1, .., 1)'$
$\delta^a = \frac{1}{\gamma}\ln(1+\frac{\gamma}{k}) + \frac{1}{k}\ln\Big(\frac{v[q+Q+1]}{v[q+Q]}\Big)$
$\delta^b = \frac{1}{\gamma}\ln(1+\frac{\gamma}{k}) + \frac{1}{k}\ln\Big(\frac{v[q+Q+1]}{v[q+Q+2]}\Big)$

Applying this algorithm involves matrix calculation, typically matrix exponential. The bigger the matrix's size, the more time it costs to execute the algorithm. One special trait of cryptocurrency is that we can trade the asset at fractional unit so the inventory level q should be accounted in consistent counting logic. In particular, the unit of inventory level is not one unit of asset but as fraction of asset. Inventory level therefore should be quite sizable to account for a large fraction of asset. This leads to computation difficulty. As we try to examine a large inventory bound Q, the size of the matrix M increases substantially. An inventory bound of 1,000 takes us almost 2.5 seconds to finish the execution. If we increase the number of interval to 2000, the execution time will be around 14.7 seconds. This executing time is by far more costly comparing to the closed-form solution in equation 2.28, which is around 1.96 microseconds. Of course, our algorithm is far from optimal as it is coded in Python, not other faster low level programming languages and there might be better techniques to deal with the matrix

calculation. However, in the sense of high-frequency trading, this can still be a huge problem as usually the execution must be completed within millisecond. Due to heavy computing problem in the execution, this method by Tapia despite its accuracy will not be included in our historical backtesting. Due to its execution inefficiency, we opt to investigate the closed-form formula in infinite time horizon by Tapia.

Figure 5.5: P&L (in USD) of Tapia strategy with first-priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit

(b) Daily profit

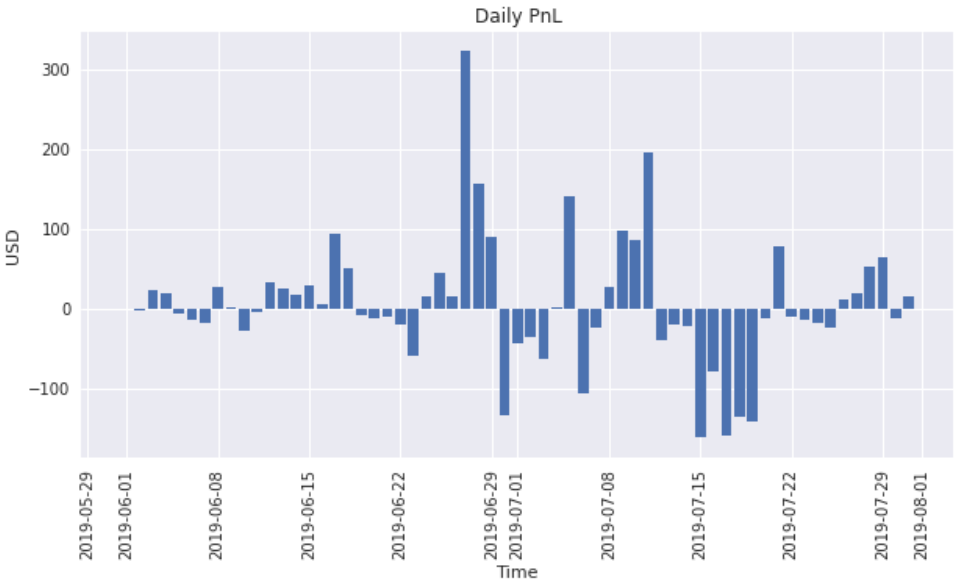
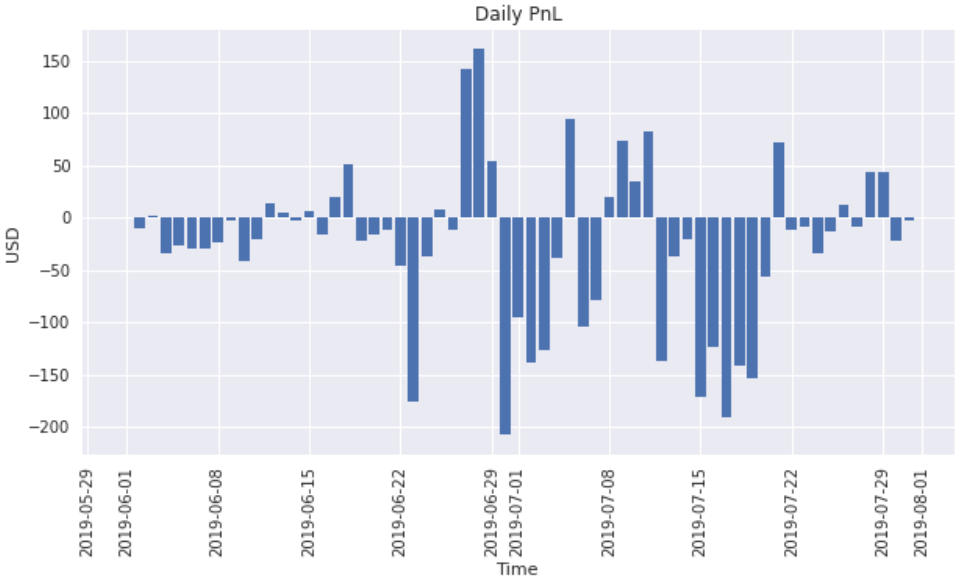


Figure 5.5 and 5.6 depicts an odd result. Neither of the two situations realised markable profit. In extremely unfavourable case, it performs not much better than the naive strategy.

Figure 5.6: P&L (in USD) of Tapia strategy with no-priority in order execution from 01/06/2019 to 31/07/2019.

(a) Cumulative profit

(b) Daily profit



Finally, in this part we compare performance indicators of those models. We begin by redefining a notion that is slightly different to common understanding.

$$\text{Daily Return} = \frac{\text{Daily P\&L}}{\text{Daily investment}} \quad [3]$$

We consider a set of three well-known ratios:

$$\text{Sharpe ratio} = \frac{\text{Total return}}{\text{Return volatility in the period}}$$

$$\text{Profit factor} = \frac{\text{Total winning value}}{\text{Total losing value in absolute term}}$$

$$\begin{aligned}\text{Maximum drawdown} &= \frac{\text{Peak} - \text{Valley}}{\text{Peak}} \\ &= \max_{\tau \in (0,T)} [\max_{t \in (0,\tau)} P(t) - P(\tau)]\end{aligned}$$

where: $P(t)$ is value of the portfolio at time t.
Table 5.1 demonstrates that A-S strategy works somewhat similarly in both extreme case where naive strate can only bring about profit in extremely favourable case (with priority).

Table 5.1: Performance statistics

	Naive strategy		A-S strategy		Tapia Inf strategy	
Indicator	Priority	No priority	Priority	No priority	Priority	No priority
Annual Return	46945.09%	-2546.65%	341.03%	310.71%	24.07%	-155.52%
Sharpe ratio	62.55	-28.08	13.90	12.80	1.06	-7.02
Profit factor	Inf	0	31	18.84	1.25	0.38
Maximum drawdown	5.39%	2,896.06%	10.24%	12.66%	82.89%	169.82%

[3]Here we accept the total value of all limit orders that we have posted during the day as proxy for daily investment.

Chapter 6

Conclusion

In the final chapter we shall summarise all the work we have put together up until now and point out the limitation of our research.

Firstly, by an intensive review of noble academic works relating to market making strategy along with essential mathematical explanation, we have verified its appropriateness from theoretical viewpoint. Avellaneda & Stoikov have well established a framework to answer the need of dealers: how to earn the widest spread from limit order placements. They founded a groundwork for further development; Tapia [7], Cartea et al. [5] later have developed the model extensively both in model assumption and in resolution tactics.

Secondly, we have drawn a picture of the application procedure from calibration process to backtesting. Based on an illustrative example in cryptocurrency market, the strategy conclusively performs better than the naive one in the sense that it is stably profitable regardless of order priority. This feature is severely important in highly competitive market where there are countless traders thus remaining to the top-priority one in every trade is almost impossible. Moreover, that the model can be interpreted in algorithm means that it is absolutely feasible to employ the strategy automatically, and even in high-frequency provided that technology requirement is satisfied.

Thirdly, we would like to highlight the pros and cons of two schemes to solve the dynamic programming equation. The asymptotic technique proposed by Avellaneda & Stoikov, in spite of its approximation, is still reliable for it being profitable in most trading days. Moreover, the closed-form solutions guarantees its feasibility in HFT. On the other hand, Tapia suggested a more precise method to the DPE problem but it is

disadvantageous in HFT. This is due to its computational costs, both in terms of time and memory. Even the approximation in infinite time horizon does not work well as we could see the performance result was worse compared to a closed-form solution (by Avellaneda-Stoikov).

Our study is also restricted to non-trivial limitations. Due to its inefficiency in real-time trading, we have excluded the matrix exponential method (by Tapia) from back-testing. One significant assumption in our research is that trading on Bitmex suffers no latency while in fact, no exchange can offer transactions without latency [19]. That we use a one-day look-back time frame for calibration and use the parameters for next trading day is another a bold assumption. Finally, that we arbitrarily chose a definite value for risk aversion coefficient γ deserves some attention because it is clearly one of the driving factors for deciding $\delta^{a,b}$.

We reserve some final words for further discussion.

- As we have mentioned above, the matrix exponential remains the most challenging part in applying Tapia's solution. Due to its complexity, we did not include in our scope of research but we are highly encouraged to study the matter in other papers.
- We omit the reason for why the approximation formula in infinite time horizon by Tapia performs worse than that of Avellaneda-Stoikov because it is not the main target of our paper. Yet we see that this matter deserves a thorough explanation.
- This paper assumes the constant volatility [3][7] but realistically it is not the case. There can be several models for volatility than can be applied to enrich the study.
- When an asset is too liquid, the spread will be too marginal so market makers may want to make their profit from volume of trades. This is another development for the original model [5] that calls for attention.

References

[1] Aldridge, I. *High-Frequency Trading: A Practical guide to Algorithmic trading.* Wiley Trading, 2010, pp. 13–35.

[2] Alexandera, C., Choi, J., Park, H., and Sohn, S. Bitmex bitcoin derivatives: Price discovery, informational efficiency and hedging effectiveness. *Journal of Futures Markets* (2019).

[3] Avellaneda, M., and Stoikov, S. High-frequency trading in a limit order book. *Quantitative Finance* (2008).

[4] Baur, D., Lee, A., and Hong, K. Bitcoin: Currency or investment? *SSRN Electronic Journal* (01 2015).

[5] Cartea, A., Jaimulgal, S., and Penalva, J. *Algorithmic and High-frequency Trading.* Cambridge University Press, 2015, pp. 246–266.

[6] Cartea, A., Jaimungal, S., and Ricci, J. Algorithmic trading, stochastic control, and mutually-exciting processes. *SIAM Review, Forthcoming* (2018).

[7] Fernandez-Tapia, J. *Modeling, Optimisation and estimation for the on-line control of trading algorithms in limit-order markets.* 2015, pp. 29–47, 83–100.

[8] Guant, O., Lehalle, C.-A., and Fernandez-Tapia, J. Optimal portfolio liquidation with limit orders. *SIAM Journal on Financial Mathematics 3*, 1 (Jan 2012), 740764.

[9] Ho, T., and Stoll, H. R. Optimal dealer pricing under transactions and return uncertainty. *Journal of Financial Economics 9* (1981).

[10] Kirk, D. *Optimal Control Theory: An Introduction.* Dover Publications, Inc., 2004, pp. 53–93.

[11] Laruelle, S. Faisabilit de lapprentissage des paramtres dun algorithme de trading sur des donnes relles. *Hors-Srie Microstructure des Marchs N1* (2013).

[12] Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics, pg. 164-168* (1944).

[13] Maclntosh, J. G. High frequency Traders: Angel or Devils? *C.D. Howe Institute Commentary* (2013).

[14] Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics, pg. 431-441* (1963).

[15] Milton, A. Description of order book, level i and ii market data. https://www.thebalance.com/order-book-level-2-market-data-and-depth-of-market-1031118, 2020.

[16] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Technical Report* (2008).

[17] O'Hara, M. *Market Microstructure Theory*. Wiley, 1998.

[18] Prial, D. Controversy over high-frequency trading is hardly new. https://www.foxbusiness.com/features/controversy-over-high-frequency-trading-is-hardly-new, April 2014. Accessed on 2019-10-30.

[19] Segdwick, K. Order speed analysis reveals the fastest cryptocurrency exchanges. https://news.bitcoin.com/order-speed-analysis-reveals-the-fastest-cryptocurrency-exchanges/, Oct 2018.

[20] Yong, J., and Zhou, X. *Stochastic Controls : Hamiltonian Systems and HJB Equations*. Springer, 1999, p. 157215.