UNIVERSITY OF LIVERPOOL

Liquidity Prediction in Limit Order Book Markets

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

Keren Dong

February 10, 2015

Contents

Illustrations

List of Figures

List of Tables

Preface

This thesis is primarily my own work. The sources of other materials are identifed. This work has not been submitted for any other degree or professional qualification except as specified.

Abstract

Limit order book markets are a rich research area, not only because these markets generated huge amounts of data (at an exceedingly high rate), but also because the fine level of detail that their data enables one to explore market microstructure in unprecedented ways.

Due to the large quantity and rich details of the data in such market, one has to leverage the power of computers to perform both the analysis and modeling work. This calls for both new algorithms and infrastructure to perform the computing tasks effectively and efficiently. Motivated by the questions and challenges I see there, I started my research first from a engineering perspective and then moved to a quantitative perspective. My aim was to find my way through this newly emerging area and develop a systematic approach to seek, study and solve the potential questions in it. I will graph and explain my findings and results in this thesis, hoping that they will help and inspire further research work.

To discipline and guide myself with a clear goal in the long journey exploring the world of limit order book markets, I focus on liquidity modeling. I try to predict trading volume from a daily scale to intra-day distributions, with the aim to design trading algorithms to reduce transaction costs and market impact. Within a microstructure context, I try to model the self-exciting nature of trading events with both a stochastic process approach and a statistical approach. Prediction methods are proposed to help trading algorithms to react to big trade events in real time.

I use two different modelling approaches. One is based on stochastic processes that have nice mathematical properties, while the other one is driven by statistics extracted directly from the data. I try to examine them in a unified and scientific way so that it is easy to compare the strengthes and weaknesses of each of them. Empirical findings are given to support the rationale behind all of the proposed algorithms.

Acknowledgements

Pursuing a PhD degree is never meant to be an easy task, at least so it seems to me. I started knowing little about the research area I would jump into, and without correctly estimating the difficulties and stresses I would have. It has taken days and nights of frustration, depression and struggle. Fortunately I have had lots of support and help from many amazing people, without whom I would not have got through this long journey.

Above all, I would like to thank my primary supervisor, Rahul Savani. Rahul has always been supportive whenever I had troubles in either research or life. Without his encouragement in the darkest period of my study, I might have already given up half way. We have had lots of decent discussions which helped me to understand the problems better and learn new technologies quickly.

I would like to thank my sponsoring company, Arctic Lake Solutions, for both the economic sponsorship and academic guidance when I was working in their office. In particular, Niels Buhl gave me many directions and inspirations for my research, his professional and hard working style has always impressed me and motivated me to go further. Nicolas Colson has always been helpful for all sorts of aspects. He is the one that I would first go to seek help whenever I got stuck in my research or was stuck while writing or making presentations. Johan Munk is the most reliable people I have seen, and has always been in charge of the most critical development work in Arctic Lake, and kept us going through difficult days.

I thank the University of Liverpool for providing me with the opportunity and academic environment and support for doing my research. My advisors, Xiaotie Deng, Martin Gairing, and Mingyu Guo are all great people, from whom I have gained a lot of insights and advice for how to do good academic work.

Last but not least, I thank my parents for the unconditional support throughout my study. They always make me feel like I am not alone and would not lose hope even in the end of the world.

Chapter 1

Introduction

In this thesis, I focus on a liquidity study in limit order book markets. Limit order books play a central role in modern exchanges' trading activities. With the rapid development of electronic trading in the last decade, the trading volume of financial markets has been increasing very quickly. Employment of computers in such activities has become more and more important for banks, brokers, hedge funds, etc. This trend brings three changes for the trading industry:

- Markets have been generating trading data at an exceedingly fast speed, this requires the participants to be able to process and analyze huge amounts of data to find better trading ideas and signals.
- Trades need to be executed in an algorithmic strategy to be cost effective and reduce market impact. The larger the trade is, the more desirable to design and optimize the algorithms for this execution.
- The exposure of the full depth of order books to market participants makes it not only possible, but also critical to study and understand the microstructure of markets deeply, especially for those with direct market access (DMA) to exchanges.

Besides, the electronization of trading also stimulates high frequency trading and low latency reaction to market events. This newly developing area poses a lot of challenges to computing infrastructure, strategy design, policy decisions and risk management. A significant research effort is required to meet the new requirements.

I am thus attracted by these new questions and challenges in the algorithmic trading area. I understand there are both technologies that haven't been developed for some time and are still worth being investigated, and the potential for new methodologies to explore and understand the problems better.

To discipline myself with a clear and definite goal in a PhD study that faces an extremely big data world and a great many potentially useful theories and technologies, I chose liquidity modeling as my major research target and focused on understanding the problems then finding and optimizing the solutions in this particular topic.

1.1 Research Question

In this thesis I try to build predicative algorithms for modeling liquidity in limit order book markets. These algorithms are to help trading applications to optimize their trading activities and minimize transaction costs.

A trading application cannot just trade as much as it needs at the price it sees in the market. It needs to know how much quantity is offered at that price. If there is less offered than it needs, it has to either wait until somebody appears to offer the rest, or suffer a worse price. The application needs to be able to judge how liquid the market is and predict when will the necessary quantity arise in the market to make the right decision.

However, as apposed to price, liquidity in the limit order book markets is not a simple concept but composed by different properties. It involves properties such as how much quantity is offered in the limit order book, how much volume was traded in a past time, how much will be traded in a future time, etc. To predict liquidity for a trading application, one has to be precise about what is needed and how does that affect the trading strategy.

I approached the research question by breaking down liquidity modeling into tasks at different level. I first tried to predict daily total volume to have a view of how much volume will be traded in total for the trading day. Then I worked on modeling intraday volume distribution, to study how will the volume be distributed within market hours. After that I moved on to microstructure level, to study how do trade events affect each other with respect to liquidity. At last I developed algorithms to predict the impact made by trade events with big sizes.

1.1.1 Total Volume

Forecasting how much volume will be traded in total before market open helps us to have a view of how liquid the market will be. A liquid market makes it easier for a trader to make a trade and have lower transaction costs. A good prediction of the total volume can help the trade strategy to optimize its schedule and resource allocation in advance.

The daily total volume is a low frequency data, i.e. it has one data point per day. It can be modeled by standard time series techniques. But it is still worth to investigate which technique works best, and is there any space for improvement. I will show how do I model and predict it in Chapter (2).

1.1.2 Intraday Volume Distribution

To execute an intraday order, a trading strategy needs to model the intraday volume distribution to decide when and how to issue the order. If the intraday volume distribution follows some pattern rather than a pure randomness, a trading strategy must have a good model to capture that pattern.

I started by doing empirical study on real market data to identify whether there is any pattern in the intraday volume distribution. Then I applied traditional trading strategies to get performance benchmarks on the data. The challenge is to develop new strategies to improve the performance. I will demonstrate what I have achieved for this in Chapter (3).

1.1.3 Microstructure Modeling

By looking into the microstructure level of the market data, i.e. individual trade events record, one would question that is there any relationship between these events, does one event have impact to the market and the following events?

Being able to properly model these events helps us to better understand how does the market evolve. And having a quantitative model can help the trading strategies to have optimal control at the microstructure level.

In particular, I tried to model liquidity at this level to identify if the market data presents some predictable features. I will show this work in Chapter (4).

1.1.4 Impact of Big Trade Events

The individual trade events are different against each other in many aspects, e.g. they have different timestamps, different prices and sizes, etc. The ones with big sizes can reveal more information and may have impact on the market. When such events occur, one can question that how much impact they will have on the market, and how long will that impact last? I will address these questions in Chapter (5).

1.2 Background

1.2.1 Limit Order Book Markets

I will introduce the limit order book and its role for algorithmic trading players in this section. In a limit order book market, the exchange maintains a central order book for each listed security. Each order book consists of a ladder of prices, where each level of the ladder is a queue of the orders sent from the participants waiting to be matched to the others. The orders coming from the buy side are bid orders, while those from the sell side are ask orders. The levels corresponding to the highest/lowest prices of the bid/offer orders are called the top of the order book, while the prices are the best bid/offer and the difference between those two is called spread.

Participants post two types of orders to the exchange to show their interest to trade:

1. Limit Order - an order to be executed at the specified price. If the exchange cannot match the price at the moment, it will put the order into a queue in the order book. This allows the trader to have a fixed execution price, which results in lower risk and cost. Because the execution time is nondeterministic, this is not in favour of an urgent/fast trading requirement.

2. Market Order - an order to be executed immediately at the current market price. The actual execution price depends on the top of order book, and the speed of transmitting

the order to the exchange. The slower the transmission, the higher risk that the price (the best bid/offer) moves away from the one at the moment the decision was made.

Table 1.1 gives an example to illustrate the state of a limit order book.

Bid Size	Bid Price	Offer Price	Offer Size
10	98	101	20
20	97	102	35
20	96	103	30

TABLE 1.1: Sample Limit Order Book

From Table 1.1 we see the best bid price is 98, while the best offer price is 101. These two levels are called the "top of book". The difference between these two, which is 3 in this example, is called the spread. With the order book as is, there is no trade. To make a trade happen, either side has to cross the order book by issuing a market order or a limit order that hits the best price of the other side. For example, if a bidder placed a market order with size 10, it would trigger a trade at offer price 101 with size 10 and the order book would be changed to:

Bid Size	Bid Price	Offer Price	Offer Size
10	98	101	**10**
20	97	102	35
20	96	103	30

Table 1.2: Order Book After a Market Order from Bid with Size 10

If the bidder places a further market order with size 20, it would hit the offer side first at price 101, with the result of wiping out all the quantities at that level, and filling another 10 at price 102. The order book would then look like:

Bid Size	Bid Price	Offer Price	Offer Size
10	98	102	**25**
20	97	103	30
20	96		

Table 1.3: Order Book After a Market Order from Bid with Size 10

Such dynamics of the order book evolution make up the microstructure of a market that reflects the economic value of the trading activities. Because the exchange will generally broadcast every change in a limit order book to all the subscribers (that have direct market access), it can generate an exceedingly large amount of data per day. Analyzing and modeling such big data bring a lot of research challenges.

I focused my research on liquidity modeling. In this area, there are many interesting problems that one could work on. I will list some of the most recent work with respect to this area below, to outline a bigger picture of what people do and why this kind of work is important.

1.2.2 Liquidity Takers

One remarkable feature of liquidity in financial markets is that it is neither constant nor random over time, but exhibits strong seasonal patterns. Understanding such patterns can be of great benefit to traders that need to take liquidity from the market. One typical application of such study is to reduce transaction costs and market impact. The more a particular trader participates in the marketable volume, the more impact it would impose on the price and the easier for others to detect his/her purpose and leverage that to go against him/her.

To avoid such potential risks, one has to be able to model the liquidity properties in advance. Incorporating such modeling into trading decisions can naturally make it a control problem. Junca (2011) gives an example of formulating the minimization of transaction cost of a risky asset as an impulse control problem, and characterizes the value function using the viscosity solutions framework.

There is a lot of other work for determining and predicting the seasonal patterns of liquidity, such as Almgren and Chriss (2001), Białkowski et al. (2008) and Humphery-Jenner (2011) all provide models for such patterns. Another attraction of modeling liquidity patterns is that they can be separated from the price process, and are much easier to forecast than price. Practitioners such as brokers can use such techniques to optimize their executions and be cost effective. I will explain more about this issue later in, Chapter 3.

Note that it is still valid to link liquidity study back to the price process as they essentially share the same underlying driving factors - good stocks attract good prices and big trades and vice versa. For example, Lehalle and Tapia (2011) proposes a model that links the liquidity distribution pattern of a portfolio to the price of the limit orders that have to be sent to the limit order book. Fruth et al. (2011) provide two liquidity based models to study price manipulation and optimal portfolio liquidation strategies.

1.2.3 Liquidity Makers

Besides takers that consume liquidity from the markets, there are also makers that provide liquidity. For that type of player, it is critical to understand when is the best time to add liquidity, what are the best prices and spreads depending on market environment, etc. For example, a market maker does not want to be part of a price crash because they are generally exposed to adverse selection risk in such situations.

For such participants, it is often critical to be able to react to markets as fast as they can, which requires technology investment in low latency and high frequency trading. The electronization of limit order book markets stimulates this trend further. For readers who are interested in this area, I recommend Loveless (2013) which present a very good illustration of how the underlying infrastructure works for making such trading activities possible.

1.2.4 Seek and Hide

The limit order book markets are far from fully transparent. Thanks to the sophisticated order types introduced by exchanges to accommodate the needs of various market participants, there are lots of seeking and hiding games in its microstructure. For example, Winne and D'hondt (2007) use regression models on 2002-2003 Euronext equities data. This data set allows hidden depth to be directly observed. They find that more than 45% of the depth at the top five levels of the LOB is hidden and that the iceberg order size is six times greater than a normal order.

Hautsch and Huang (2012) build a Bayesian model with a Bernoulli likelihood function using logit multiple regression, where the probability of an order being executed with hidden liquidity can be predicted by eight predictors, including distance from the mid price, the size of the spread, and the lagged return. Christensen and Woodmansey (2013) present an algorithm based on joint-distributed distribution estimation for detecting hidden orders in GLOBEX an electronic trading platform. For the sake of low latency execution, they also outline a computational implementation of the algorithm that leverages cloud computing to accelerate the model estimation.

1.3 Related Work

Research work in liquidity modeling area has been made on optimization and simulation. Lehalle (2013) gives a good survey about market microstructure knowledge with respect to intra-day trading processes. It confronts the recent advances in optimal trading, order book simulation and optimal liquidity seeking with the reality of trading in an emerging global network of liquidity.

The methodologies that have been employed in the previous research work with respect to liquidity modeling are inline with the other quantitative financial modeling and engineering exercises. As Meucci (2011) outline, there are two main streams in the quantitative financial modeling world: the "$\mathbb{Q}$" area to "extrapolate the present" and the "$\mathbb{P}$" area to "model the future". The differences of the two are:

- "$\mathbb{Q}$" - Its theory is mainly based on risk-neutral stochastic processes, which are arbitrage-free and do not reward risk, a.k.a martingales. A martingale is a process P_t with constant expected value: $P_0 = \mathbb{E}\{P_t\}, t \geq 0$. Such models are generally time-continuous, and rich in mathematical analytics. They are often used on the sell side for derivatives pricing. The risk-neutral assumption makes them easy to evolve without much input from real world data, which gives a large space for mathematicians to apply techniques in stochastic processes, partial differential equations, Ito calculus, etc. to not only be able to price existing financial products, but also design and implement new products. This approach uses data to calibrate the models, which means the structure of a model is not changed during calibration, only the parameters of the model are varied to find the optimal set.

- "$\mathbb{P}$" - This approach emphasizes the data more than theoretical assumptions. Letter $\mathbb{P}$ denotes the real probability distribution of the data. The real distribution is estimated from market data using multivariate statistics and econometric techniques. Modeling in this area uses discrete-time series like ARMA, GARCH, etc. Data drive the modeling process in the sense that the structures of models depend on the estimation result on data, there can be multiple candidates of models in the estimation process and only the one that best fits the data will be picked up. Historically, this approach has been employed more by the buy side for quantitative risk and portfolio management. However, with the recent rapid development of machine learning and big data technologies, it is becoming more and more valuable and attracting considerable interest.

Though each of the two approaches has strong and weak sides and a typical application area, there is not a hard line that limits one to use them in the other's area. Actually, it makes more sense to keep both in mind when addressing a specific research task and let each of them to cover their strong points in the problem. Following this principle, I list the state-of-the-art of research with a particular focus on liquidity study and modeling below.

1.3.1 Stochastic Modeling

The typical approach in the quantitative financial world is driven by mathematical modeling with stochastic processes. This approach does not rely much on historical data and provides possibilities to understand the underlying nature of markets not obviously observable from data.

In terms of liquidity modeling and, in particular, limit order book modeling, there are lots of studies that try to approach solutions from this direction. Cont et al. (2010) propose a continuous-time stochastic model for the dynamics of a limit order book. The model strikes a balance between three desirable features: it can be estimated easily from data, it captures key empirical properties of order book dynamics and its analytical tractability allows for the fast computation of various quantities of interest without resorting to simulation. Cont and Larrard (2012b) propose and study a simple stochastic model for the dynamics of a limit order book in which arrivals of market orders, limit orders and order cancelations are described in terms of a Markovian queueing system.

Besides the work focussing on stochastic modeling listed above, a subset of stochastic processes has drawn a lots of interest in this area recently, i.e. point processes family. Bayraktar and Ludkovski (2014) consider a point process based approach to model the intensity of order arrivals depending on the liquidation price for solving optimal liquidation problems in limit order books. Cont and Larrard (2012a) propose a model for the dynamics of a limit order book in a liquid market where buy and sell orders are submitted at high frequency. Hewlett (2006) fit a bivariate Hawkes process to arrival data for buy and sell trades in FX markets to predict the future imbalance of buy and sell trades conditional on the history of recent trade arrivals.

1.3.2 Statistical Inference

Compared with building models from some ideal assumptions and risking not capturing the full picture of the data, another approach is to develop models from the basis of statistics directly obtained from the data.

The emerging technologies in this area, such as machine learning and big data, provide great potential for financial practitioners too. Nevmyvaka et al. (2006) present a large-scale empirical application of reinforcement learning to optimize execution in limit order book markets. Kearns and Nevmyvaka (2013) provide three examples of applying machine learning technologies to high frequency trading strategies. Their work suggests that machine learning provides a powerful and principled framework for trading optimization via historical data, but not necessarily profitable execution.

The fact that this approach is heavily driven by large amounts of data does not mean that it is not useful in real time trading strategies. Loveless et al. (2013) proposed online algorithms that act extremely fast on large amounts of data received in a short time. Additionally, as mentioned above, Christensen and Woodmansey (2013) make a good example for how to coordinate statistical inference modeling and low latency trading technologies.

1.4 Methodology

Due to the data-driven nature of my research question, I decided to take empirical study as the first principle. That is, the ideas I propose have to be supported and verified by the data. I spent lots of energy setting up the data processing framework and finding evidence in the huge amount of data I have. By looking into the very details of raw data, and carefully designing the testing process, I found it often revealed unexpected difficulties with respect to problems that at first appeared to be not so difficult. Although sometimes such difficulties can be frustrating, I actually think it is the only way to really understand the problems and find effective solutions.

On the basis of solid analytical work, I tried both the "$\mathbb{Q}$" and "$\mathbb{P}$" modeling approach to my goals. The reason that I tried both ways is that with the "$\mathbb{Q}$" approach it is easier to construct mathematical models, which enable us to analyze theoretical properties, perform simulations and test in a more controllable way, while with the "$\mathbb{P}$" approach it is easier to seek help from the data itself and quicker to get some meaningful results out of the apparent mess. I understood both approaches have limits and weaknesses so I kept myself open to new ideas from both sides.

In terms of the "$\mathbb{Q}$" method, I chose to study a family of point processes to model the naturally time-discrete trading events in the market, i.e. Hawkes processes. This technology has been used in seismology for many years and recently drawn interests in the financial modeling world too. I examined the goodness of fit of such models with real data, and proposed algorithms to make prediction strategies.

The attractive side of the "$\mathbb{Q}$" method is that it does not require too much data. It puts more effort on the mathematical modeling part, which often requires sophisticated equations rather than actually processing data and finding clues in the numbers. This of course reduces the need of big data processing in the early stage of research work. One can make up the models with theoritical work and verify on small data set. Once the idea gets confirmed it can then be applied on larger scale of data. The mathematical work also makes it more easier and clearer to build solid theorical foundation for the models. In the later chapters, I will how this approach work in my research.

For the "$\mathbb{P}$" side, I went with massive empirical analysis on real data. I took the evidence I found in the data as the first place driver for modeling and algorithm design. While the modeling itself can be straightforward, I use the advantage that I have in the quality and quantity of data to approach questions that have not been studied yet.

Because of the data driven nature of the "$\mathbb{P}$" method, it does not require much work on mathematical derivation, but focus more on data itself. It relies more on the data processing infrastructure. The more convenient and faster the infrastructure can process data, the easier the researchers can approach the problem and get solution. Although the infrastructure itself is not the focus of this thesis, I do get lots of help in my research with a good underlying system that we build at Arctic Lake. Thanks to the computing power we haver there, I was able to run many empirical experiments on the data sets and got lots of hints out of the process. I will show my work in the later chapters, and that most of the time strategies with simple statistics modeling can present unanticipated power and good results.

1.5 Contribution

Because of the data-driven nature of the research question, the first effort of my research work is to present analysis on real market data, to make good foundation for the later work. I have done massive empirical study on two data sets, one comes from London Stock Exchange (LSE) and the other is from National Stock Exchange (NSE). Both data sets have detailed limit order book events information, which helps me to verify my prediction algorithms in finest details.

It is very useful for a trading strategy to have a view on how liquid the market will be before the market is open. This view can help the strategy to better schedule its trading activities and allocate the resources. I have proposed an algorithm to predict the daily total volume for this purpose. The algorithm uses an ARMA model with a decomposition of market trend and individual stock specific trend. It systematically outperforms all benchmark models.

For those market participants who have large orders to trade, they need to both reduce transaction costs and minimize market impact. To achieve that, they split the large orders into small chunks and trade them in sequence. This type of task requires a good modeling of intraday liquidity distribution, so that the small chunks of a large order can follow the distribution to hide themselves from the counterparties. I have done empirical analysis for the traditional strategy on real data and found it is very difficult to improve that strategy. I have overcome the difficulty by proposing a bidirectional trading strategy, which dynamically adjusts balance between the volume bought and sold to stick to the target order quantity. This bidirectional strategy shows a significant performance improvement.

Trade events in financial market come with different properties. Those events that have large sizes or special prices can make impact to the market and affect trading strategies from counterparties. I have applied a stochastic model named Hawkes process to study intraday liquidity data, and verified that the NSE data present the effect that trade events tend to be self-exciting.

Following the above finding, I tried to model the liquidity impact of the trade events that have large sizes. I have developed a prediction algorithm to forecast the liquidity impact of such events. The algorithm is built on historical statistics and uses a spread-based classification. It provides reasonable prediction results, and provides a foundation for trading strategies that are sensitive to liquidity impact.

1.6 Organization

I organize this thesis as follows:

Chapter 2 presents prediction algorithms for total daily volume. It outlines the evaluation framework for the algorithms, and gives empirical study on market data. In this chapter I develop a new algorithm based on ARMA model with a decomposition of market feature and stock specific feature. I show that this algorithm outperforms the benchmark ones.

Chapter 3 models intraday volume distribution. Based on the models, it proposes methods to improve VWAP trading strategies. It gives analysis of how VWAP trading strategies work, and shows massive empirical analysis on market data from two exchanges. In this chapter I develop a bidirectional VWAP trading strategy that trades at both buy and sell side. This strategy gives significant improvement over the traditional strategies.

Chapter 4 uses the Hawkes processes to model liquidity in microstructure level. It describes the theoretical foundation of the Hawkes processes, and both parametric and non-parametric kernel estimation methods for the Hawkes processes. In this chapter I apply the Hawkes processes on market data from National Stock Exchange (NSE) and show that there is a self-exciting feature in the data.

Chapter 5 gives prediction algorithms for the liquidity impact of big trade events. It gives empirical analysis to show that big trade events do have impact on liquidity. In this chapter I propose four algorithms to predict the impact. I show that all the four algorithms give reasonable results, while the one built on historical statistics with a spread-based classification performs the best.

Chapter 6 concludes and outlines the future research work.

Chapter 2

Predicting Daily Total Volume

2.1 Introduction

In this chapter, I try to build a predictor for daily total volume, which provides a base for the further modeling of intraday liquidity dynamics. I demonstrate a first attampt to build the forecast models for trading with a data driven approach. I develop the models with empirical finding and benchmark them with real data. I will show in later chapters the impact of the results we get in this chapter. The most important contribution I make in this chapter is an ARMA based forecasting model, with a decomposition of the market feature and stock specific feature, that makes the best prediction of the daily total volume.

Historically, volumes have not drawn as much interest as prices or volatility in financial theory study. However, it is unquestionable that volumes play an important role in the trading process and contribute to price and volatility dynamics. It also seems easier to analyze and predict volumes than prices, in the sense that volumes are less likely to be volatile than prices, especially for liquid stocks.

For the sake of algorithmic trading strategies, the daily total volume is obviously the first quantity that one would think of when it comes to quantifying the intra-day liquidity of a financial stock. Predicting the daily total volume is often the first step that a trading strategy will do to optimize the resource allocation, in favour of minimizing market impact and reducing transaction cost. A good prediction of the daily total volume can also offer hints for volatility, in the sense that more trading activity could imply significant economic news or a high incentive to trade from big institutions, and thus more movements of price. And as we will see in the next chapter, such predictions have a big impact on a so called VWAP (Volume Weighted Average Price) trading strategy.

Both academia and practitioners have been making efforts in the pursuit of a better modeling approach for these kinds of forecasts. Hautsch (2002) proposed a strategy that models intraday liquidity with the concept of durations using autoregressive conditional duration (ACD) that was originally proposed by Engle and Russell (1998). Although this strategy does not make predictions directly for the end-of-day volume, its idea to forecast the waiting time for a predetermined volume can already help a lot for any trading algorithm that requires a liquidity model. Similarly, Białkowski et al. (2008) used the autoregressive moving average

(ARMA) and self-exiting autoregressive (SETAR) models to predict the intraday volume in a dynamic manner. Białkowski et al. (2008) also introduced a way to decompose the intraday liquidity dynamics into two parts, one reflecting the whole market trend and the other representing the stock specific pattern, for the sake of improving VWAP trading strategies. Darolles and Fol (2003) shared the same idea but with a focus on predicting the daily volume, i.e. the decomposition was done on the daily rather than the intraday level. Kissell (2013) proposed a forecasting approach based on an ARMA model. They fed factors including moving average daily volume, a historical look-back period, a day of week effect or a lagged daily volume term into the ARMA model.

I extend the work of Kissell (2013) in a way that decomposes the volume into two parts, one is the total volume of all stocks, and the other is the percentage of each stock. I predict these two parts separately and then consolidate the results to forecast the volume of each stock respectively.

I show that the this new decomposing based model systematically outperforms all the other models on the two data sets we have.

2.2 Modeling

Although at first glance the task is quite simple: to predict a single number as the total end-of-day volume for an stock before market open, there is no need for a sophisticated formation process model and not much data is needed as input (in comparison with the tick data amount for intraday price/volume dynamics), we still need to approach it in a systematic manner. I first try to answer these questions:

- What information do we have when making the prediction?
- Which information do we need to make the prediction?
- What technologies do we use for the modeling?
- How do we evaluate the performance of our models?

The conclusions from these questions will outline the task precisely and direct us to a numerical practice with real market data. I will discus and analyze them in the following sections of this chapter.

2.2.1 Information Selection

First we need to review the information we have before starting forecasting. In a stock market, for any stock, we would know the following information of the last trading day before market open:

- Price - Including Open/Close/High/Low price
- Volume

- Maybe some news

In addition, we also know all of these for any historical day. So for any one of them, we can calculate a moving average based on historical data. To simplify the question, we first try to focus on the types of the four kinds of information, to see what could contribute to the forecasting task.

First let us look at price. The analysis of volume and price has no doubt been a research topic for the last few decades, and there are many empirical studies that prove they have a positive correlation. We refer the reader to the work of Karpoff (1987) and Chandrapala (2011). The basic conclusions from Karpoff (1987) were:

- A small volume is usually accompanied by a fall in price.
- A large volume is usually accompanied by a rise in price.
- A large increase in volume is usually accompanied by a large price change.
- A large volume is usually followed by a rise in price.
- If the volume has decreased (increased) for five straight trading days, the price will tend to fall (rise) over the next four trading days.

The above suggests that it is more common to view volume as a factor of price change, not the opposite. Plus, the price process is a vastly complicated research problem. Thus, we decide not to take price into account when predicting volume.

The next information is historical volume. This is obviously quite important information, and what other researchers are actually focussed on when doing the prediction task.

The last one is news. The relationship between news and finance has recently drawn a lot of interest from researchers, such as Bollen et al. (2010), Nagar and Hahsler (2012), Zhang (2013), Aase (2011), Schumaker et al. (2009). These were all focussed on the relationship to price, none was for volume. Although the analysis for linking the news to volume sounds interesting, I do not have access to good data source for news, so this was left out of my choices.

With all the above concerns, I decided that I were going to forecast volume with historical volume data. The next step is to formalize the input and output for the models.

2.2.2 Definitions

My aim is to predict daily total volume for a set of stocks $\{s_t\}_{t \in 1,\dots,N}$ where N is the number of stocks. I use L days of historical data to build moving averages. For each stock s_t, the prediction is a function $f(\tilde{v}_t)$ to transform a vector of historical volume data $\tilde{v}_t$ to the volume for the target day $\hat{v}_t$.

$$F(\tilde{v}_t) = \hat{v}_t, \tilde{v}_t = \begin{pmatrix} v_{t-1} \\ v_{t-2} \\ \vdots \\ v_{t-L} \end{pmatrix} \tag{2.1}$$

If we group the N stocks together, we get a prediction function $F(\tilde{V}_t)$ that transforms the matrix $\tilde{V}_t$ to a vector output $\hat{V}_t$ as the daily volume of the respective stocks.

$$F(\tilde{V}_t) = \hat{V}_t, \tilde{V}_t = \begin{pmatrix} v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L} \\ v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L} \\ \vdots \\ v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L} \end{pmatrix}, \hat{V}_t = \begin{pmatrix} \hat{v}_{1,t} \\ \hat{v}_{2,t} \\ \vdots \\ \hat{v}_{N,t} \end{pmatrix} \tag{2.2}$$

For each day in the test set, we have the recorded total volume $v_{i,t}$ for stock s_i, this enables us to measure the prediction error $\epsilon_{i,t}$, we define $\epsilon_{i,t}$ as percentage relative to $v_{i,t}$.

$$\epsilon_{i,t} = 100 \times \frac{v_{i,t} - \hat{v}_{i,t}}{v_{i,t}} \tag{2.3}$$

Thus if we have M days of test data, we have an error matrix ε.

$$\varepsilon = \begin{pmatrix} \epsilon_{1,t-1}, \epsilon_{1,t-2}, \ldots, \epsilon_{1,t-M} \\ \epsilon_{2,t-2}, \epsilon_{2,t-2}, \ldots, \epsilon_{2,t-M} \\ \vdots \\ \epsilon_{N,t-1}, \epsilon_{N,t-2}, \ldots, \epsilon_{N,t-M} \end{pmatrix} \tag{2.4}$$

From this matrix, it is obviously not easy to get an idea about the performance of a prediction model, because there are many numbers when M and N are large. We need a function to simplify the matrix to a single number for that purpose. I define the error function as $Err(\varepsilon)$. I will give several candidates for $Err(\varepsilon)$ later.

With the above definitions we have a way to define the optimal forecasting algorithm.

Definition 2.2.1. Having $L + M$ days of data for N stocks, an optimal algorithm for forecasting daily total volume is a function $F(\tilde{V}_t) = \hat{V}_t$ that minimizes the error measurement $Err(\varepsilon)$, where $\varepsilon = 100 \times \frac{V - \hat{V}}{V}$

In this chapter I propose four algorithms to predict the total volume, and I try to find the best one among the candidates.

2.2.3 Forecasting Models

Lagged

I start from a naive model that basically forecasts the volume as the number for the last day. I call this one Lagged algorithm.

$$F(\begin{pmatrix} v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L} \\ v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L} \\ \vdots \\ v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L} \end{pmatrix}) = \begin{pmatrix} v_{1,t-1} \\ v_{2,t-1} \\ \vdots \\ v_{N,t-1} \end{pmatrix} \tag{2.5}$$

On the face of it, this one does not make much sense because it uses very little historical information, however it will give us a baseline for the worst case.

Moving Average

Going along with the idea of the Lagged algorithm that uses historical data as a prediction indicator, we can extend it further to use a moving average of the whole L days of data. I define m as the function to calculate an average of L historical volume $\{v_{i,k}\}_{k\in 1,\ldots,L}$ for each stock s_i, so the algorithm can be presented as:

$$F(\begin{pmatrix} v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L} \\ v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L} \\ \vdots \\ v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L} \end{pmatrix}) = \begin{pmatrix} m(v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L}) \\ m(v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L}) \\ \vdots \\ m(v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L}) \end{pmatrix} \tag{2.6}$$

For the average function m, we have two options:

- Mean - We take the mean of $\{v_{i,k}\}_{k\in 1,\ldots,L}$ as the average, and call this algorithm MA-Mean.
- Median - We take the median of $\{v_{i,k}\}_{k\in 1,\ldots,L}$ as the average, and call this algorithm MA-Median.

Without applying them to real data, we do not know which one is a better choice. I will compare the differences in these two approaches in an empirical study later.

ARMA

The moving average model does not take prediction errors into account. As we will see in the empirical study, these errors present a strong autoregressive feature. This suggests extending the model into a ARMA (Autoregressive Moving Average) model.

Following the classic ARMA model I define the forecasting algorithm as:

$$F(\begin{pmatrix} v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L} \\ v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L} \\ \vdots \\ v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L} \end{pmatrix}) = \begin{pmatrix} m(v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L}) + \beta_1\epsilon_{1,t-1} \\ m(v_{2,t-2}, v_{2,t-2}, \ldots, v_{2,t-L}) + \beta_2\epsilon_{2,t-1} \\ \vdots \\ m(v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L}) + \beta_N\epsilon_{N,t-1} \end{pmatrix} \tag{2.7}$$

where $\epsilon_{i,t-1}$ represents the error terms and β_i is the autoregressive sensitivity parameter. In the classic ARMA model, the error terms $\epsilon_{i,t-1}$ are calculated as:

$$\epsilon_{i,t-1} = v_{i,t-1} - \hat{v}_{i,t-1} \tag{2.8}$$

However as suggested by Kissell (2013), this will cause persistence of the error term as it includes the previous day's error in the forecast, which they had found is not necessarily needed for a more accurate estimation. They used a definition that only counts the difference of $v_{i,t}$ and the moving average part of $\hat{v}_{i,t}$, as follows:

$$\epsilon_{i,t-1} = v_{i,t-1} - m(v_{i,t-2}, v_{i,t-3}, \ldots, v_{i,t-L-1}) \tag{2.9}$$

I follow this definition in this chapter. This also makes the computation easier because we can calculate the moving average and the autoregressive part separately.

ARMA With Decomposing (ARMA-DECOMP)

The models we have discussed so far all forecast volume for each stock s_i independently. One could imagine the liquidity of the stocks can also be affected by the market state. If there was some macroeconomic news, the whole market can be stimulated to be more or less liquid than usual, this of course would have an impact on the individual stocks. Therefore we introduce a model that takes this concern into account.

Firstly, I model the total daily volume of the whole market as a ARMA process as described in Section (2.2.3). Let $v_t^{market} = \sum_i^N v_{i,t}$, the prediction of v_t^m is:

$$\hat{v}_t^M = m(v_{t-1}^{market} + v_{t-2}^{market} + \ldots + v_{t-L}^{market}) + \beta^m \epsilon_{t-1}^{market} \tag{2.10}$$

For each stock, I model its percentage in the total market daily volume $p_{i,t}$ as the same ARMA process:

$$\hat{p}_{i,t} = m(p_{i,t-1} + p_{i,t-2} + \ldots + p_{i,t-L}) + \beta_i \epsilon_{i,t-1} \tag{2.11}$$

So the actual prediction of daily volume for the stocks is:

$$F(\begin{pmatrix} v_{1,t-1}, v_{1,t-2}, \ldots, v_{1,t-L} \\ v_{2,t-1}, v_{2,t-2}, \ldots, v_{2,t-L} \\ \vdots \\ v_{N,t-1}, v_{N,t-2}, \ldots, v_{N,t-L} \end{pmatrix}, \begin{pmatrix} \hat{p}_{1,t} \\ \hat{p}_{2,t} \\ \vdots \\ \hat{p}_{N,t} \end{pmatrix}) = \begin{pmatrix} \hat{p}_{1,t}\hat{v}_t^{market} \\ \hat{p}_{2,t}\hat{v}_t^{market} \\ \vdots \\ \hat{p}_{N,t}\hat{v}_t^{market} \end{pmatrix} \tag{2.12}$$

2.2.4 Evaluation Method

I first define the prediction error $e_{i,t}$ for day t of stock s_i as the log of the percentage between the actual volume $v_{i,t}$ and the predicted volume $\hat{v}_{i,t}$.

$$e_{i,t} = ln \frac{v_{i,t}}{\hat{v}_{i,t}} \tag{2.13}$$

For stock s_i, I define the total prediction error across all days as the standard deviation of the single day error:

$$\tilde{e}_i = \sqrt{\frac{\sum_t^M (e_{i,t} - \bar{e}_i)^2}{M}} \tag{2.14}$$

For the whole test data set, I define the final error term as the simple average of the error of each stock:

$$E = \frac{\sum_i^N \tilde{e}_i}{N} \tag{2.15}$$

I take the value of E as the performance indicator for each forecasting algorithm.

To decide which average method (mean and median) and what is the best number of historical days L, I follow an empirical approach. I do this by running both methods while varying L from 1 to 30 across the whole data set, and dynamically update the historical window so it always contains the L days before each testing day. I choose the L value that presents the best performance out of this process. I will explain this process in the next section.

2.3 Empirical Study

I have two data sets for performing empirical work. I will first introduce each of the data sets and then carry on to the empirical findings in this section.

The first data set is the market data of stocks listed in the NIFTY index of the National Stock Exchange (NSE), from January 2, 2012 to April 8, 2013, counting 301 effective trading days.

The other one is the market data of 20 stocks listed in the London Stock Exchange (LSE), from January 3, 2010 to August 28, 2010, counting 167 effective trading days.

2.3.1 Moving Average Methods

First we want to examine the effect of the moving average methods and the impact of the length of look-back periods L. Because we need to vary L, the initial days for building the moving average profile for different value of L differ.

I vary the length of look-back periods by taking the odd numbers from 3 to 29, so it is always easy to determine the median of the look-back periods. I fix the start of the testing periods to the 30th day of each data set, so that each iteration has exactly the same set of test data. So in our case, the number of days in test data is 272 for NSE and 138 for LSE. I show the prediction performance of both moving mean and moving median methods for the two data sets in Figure (2.1) and Figure (2.2).

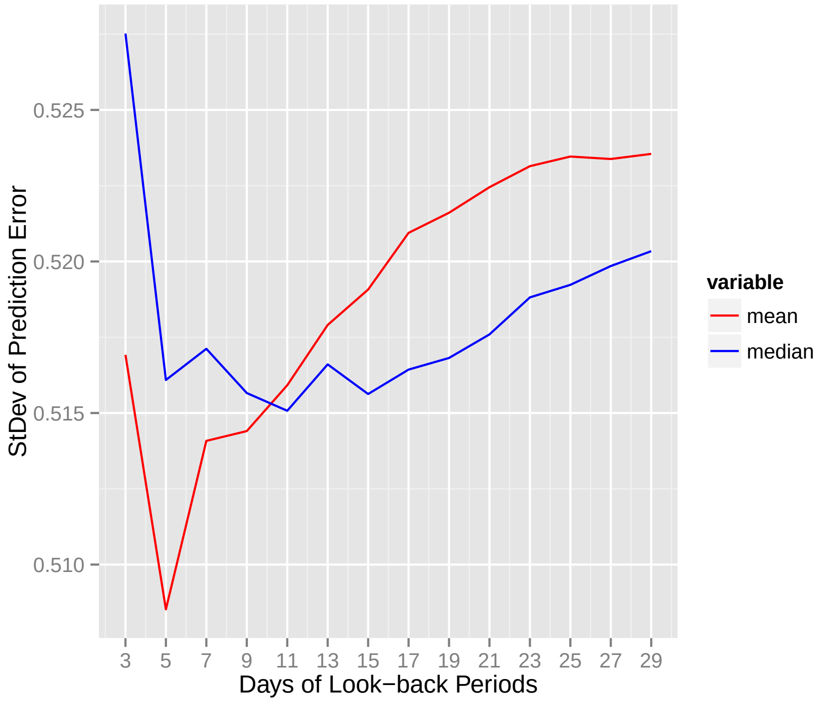


FIGURE 2.1: Prediction of Daily Volume with Moving Average (NSE)

Figure (2.1) outlines the results for the NSE data. For each value of L ranging from 3 to 29, we calculate the standard deviation of prediction error over the 272 testing days. From the figure we see that the mean method performs best when the length of the look-back period is set to 5, while the median method has 11 as the optimal value. It is clear that the best way to average is the mean method with a 5-day look-back period.

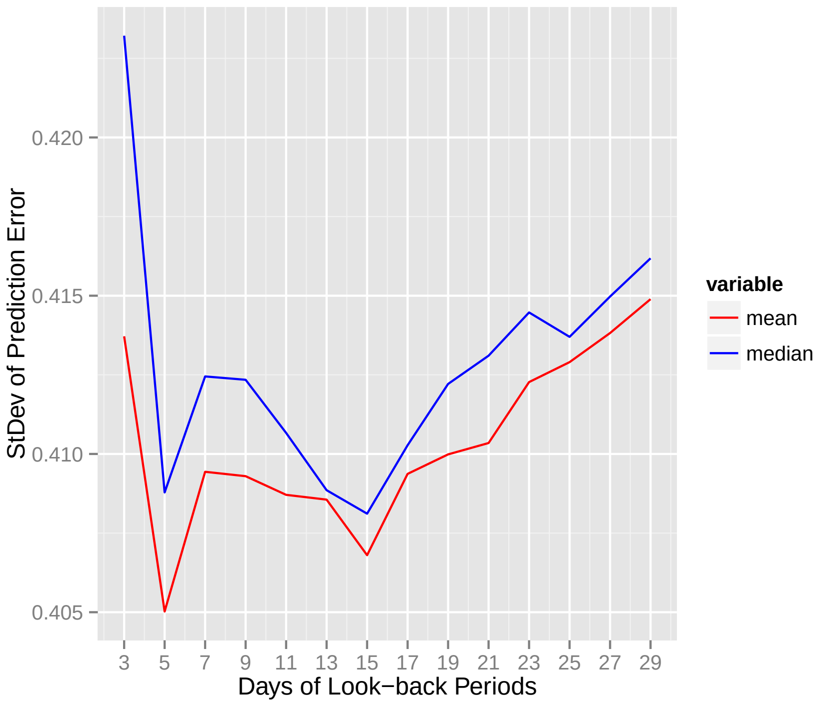


FIGURE 2.2: Prediction of Daily Volume with Moving Average (LSE)

Figure (2.2) represents the output of the LSE data. For each value of L ranging from 3 to 29, we calculate the standard deviation of prediction error over the 138 testing days. It is inline with that of the NSE in the sense that the mean method outperforms the median method. The best setup turns out to be the mean method with a 5-day look-back period.

2.3.2 Error Terms

To determine the proper setup of ARMA model for the error terms predictor, I do the standard analysis on the error data, i.e. ploting the autocorrelation (ACF) and partial autocorrelation (PACF) functions of the error terms produced by the moving average models. Figure (2.3) is the ACF plot, while Figure (2.4) represents the PACF plot.

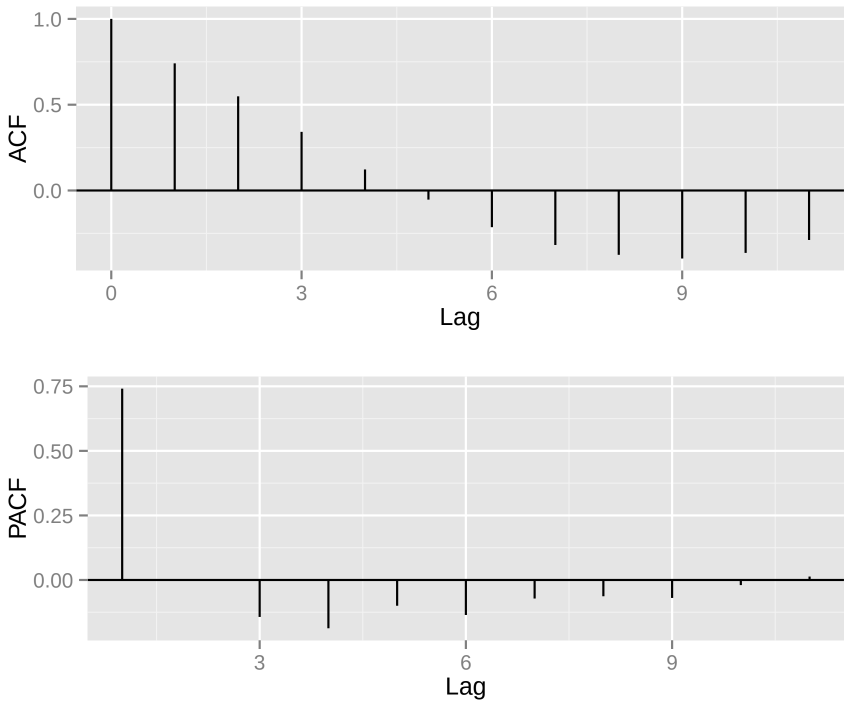


FIGURE 2.3: ACF and PACF of Prediction Errors of MA-MEAN Model (NSE)

Figure (2.3) is the one for the NSE data, from which we can see a clear autoregressive feature.

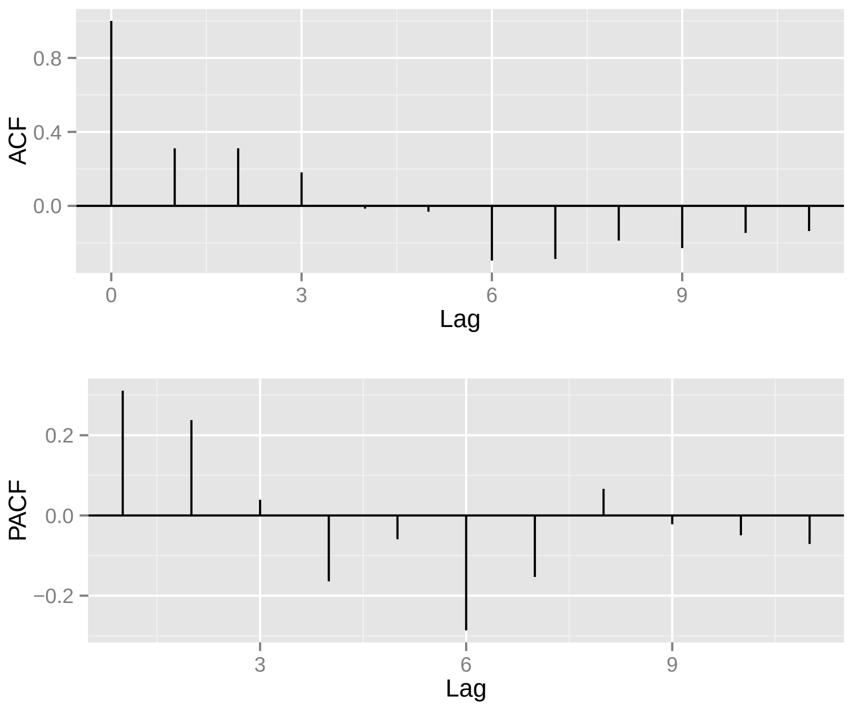


FIGURE 2.4: ACF and PACF of Prediction Errors of MA-MEAN Model (LSE)

Figure (2.4) shows the one for the LSE and once again confirms an autoregressive feature of the error terms.

The evidence we see from these figures suggests modeling the error terms with an AR model.

2.3.3 Summary of Daily Volumes

Before we jump into the model implementations and performance tests, we would like to have a view on the numbers for the volumes. I therefore take some statistics on the data sets we have and list the summary below.

Table (2.1) represents the statistics summary for the NSE data. The table is sorted by the average daily volume of each stock. Each row represents a stock while the last one is the information of the total volume of all stocks in the NIFTY. The numbers in the first column are the average daily volumes of the whole 301 days for the corresponding stock. The second column represents the average percentage of the volume for each stock in the sum. It is the arithmetic average of the daily percentage for all the 301 days. The third and fourth column measure the volatilities of the numbers in the first and second column, respectively. We define the volatility here as the standard deviation of the log ratios between the daily measures and the average measure, presented as a percentage of the average measure. These numbers tell how the numbers fluctuate during the full period.

From the table we see two features:

- The total volume of all stocks is much more stable than the one of each individual stocks, near 10% less volatile than the least fluctuated stock.
- The percentage of the volume in total for each stock varies less than the volume itself.

Symbol	$\bar{Daily\ Vol}$	$\bar{Percent}(\%)$	Vol StDev(%)	Percent StDev(%)
GRASIM	72271.602	0.1	53.998	52.112
ASIANPAINT	80510.759	0.1	55.464	51.405
ULTRACEMCO	176338.353	0.1	67.449	63.933
SIEMENS	214571.925	0.2	62.807	60.735
DRREDDY	335046.614	0.3	52.005	50.442
ACC	345738.257	0.3	55.891	50.784
HEROMOTOCO	368946.473	0.3	49.496	48.332
BAJAJ-AUTO	434851.859	0.3	49.995	48.603
PNB	666901.855	0.5	54.662	52.584
KOTAKBANK	721634.730	0.6	57.905	54.124
LUPIN	747233.349	0.6	55.882	51.626
SUNPHARMA	786446.606	0.6	47.733	43.566
BANKBARODA	799487.232	0.6	60.180	51.769
MARUTI	809033.743	0.6	55.188	54.131
RANBAXY	844543.988	0.6	52.952	49.968

BPCL	947589.320	0.7	66.616	62.628
GAIL	1092650.328	0.8	50.694	45.756
INFY	1177619.261	0.9	53.890	51.250
M&M	1232748.784	0.9	48.468	39.746
TCS	1250417.232	0.9	56.500	48.675
HCLTECH	1261101.697	0.9	64.828	58.618
WIPRO	1466407.490	1.1	53.733	51.992
LT	1589210.224	1.2	43.685	41.343
CIPLA	1610910.178	1.2	55.752	50.066
COALINDIA	1911433.739	1.4	50.282	44.537
SESAGOA	1948706.162	1.5	44.767	44.738
AXISBANK	1984365.880	1.5	42.235	38.908
RELINFRA	2105503.560	1.6	36.095	35.546
JINDALSTEL	2216658.801	1.7	57.704	53.351
HINDUNILVR	2282491.967	1.7	52.856	48.140
AMBUJACEM	2371611.452	1.8	51.119	47.804
SBIN	2447830.635	1.9	40.293	39.137
HDFCBANK	2663834.091	2.0	42.338	33.701
HDFC	2896218.967	2.1	51.699	43.563
TATAPOWER	2980003.676	2.2	53.098	45.631
POWERGRID	3092314.490	2.3	56.327	48.545
RELIANCE	3101789.066	2.4	38.528	32.545
ONGC	3220842.589	2.4	51.583	44.167
ICICIBANK	3356336.714	2.5	42.740	37.236
NTPC	3676125.282	2.6	73.151	64.995
TATASTEEL	3828208.328	3.0	33.469	31.932
CAIRN	3922954.730	2.5	63.786	54.721
BHEL	4271832.519	3.2	52.029	47.060
BHARTIARTL	5079555.556	3.8	57.070	52.371
IDFC	6143510.033	4.7	39.644	33.459
ITC	6368947.763	4.7	49.628	36.885
DLF	6766916.456	5.1	45.569	40.340
HINDALCO	7174421.863	5.4	42.171	35.694
TATAMOTORS	10339395.548	7.7	43.904	37.863
JPASSOCIAT	18099749.245	13.7	35.772	28.917
All	133283770.971	100.0	25.095	N/A

Table 2.1: Statistics of Daily Volumes of the NSE

Table (2.2) shows the same information for LSE data. It confirms the same pattern that we see in Table (2.1).

Symbol	$\bar{DailyVol}$	$\bar{Percent}(\%)$	Vol StDev(%)	Percent StDev(%)
BANKBARODA	1011879.802	0.2	52.647	52.443
GAIL	1266103.371	0.2	44.580	41.491
ACC	1824505.443	0.3	48.636	46.980
CIPLA	2035243.078	0.4	41.559	47.013
ASIANPAINT	2764894.192	0.5	45.440	45.475
GRASIM	2951859.856	0.5	50.297	50.726
BAJAJ-AUTO	3109411.246	0.6	43.132	38.545
HEROMOTOCO	3700574.174	0.6	37.748	30.325
BPCL	5075458.018	0.9	37.772	29.761
BHARTIARTL	5207226.114	0.9	40.647	30.968
HINDALCO	6667037.371	1.2	37.695	29.858
HCLTECH	9805202.090	1.8	38.820	36.748
HDFCBANK	11376395.611	2.0	40.755	35.498
AMBUJACEM	13984190.641	2.5	36.712	29.091
BHEL	18957191.150	3.4	40.244	40.221
DRREDDY	23654371.329	4.4	34.658	40.807
AXISBANK	33743249.078	5.8	42.154	28.643
DLF	107145878.275	18.6	39.255	28.740
CAIRN	121634437.431	20.2	48.829	27.334
COALINDIA	215962899.737	35.1	48.433	21.224
All	591878008.006	100.0	35.650	N/A

Table 2.2: Statistics of Daily Volumes of the LSE

These findings confirm the idea of decomposing the prediction into two parts (one for total volume and the other for percentage), which would reduce noise and capture changes in the two directions respectively.

2.4 Performance

In this section, I address the performance of the prediction models discussed above. The results are a first step towards forecasting the liquidity of the market and important criteria in making further predictions of intra-day volume distribution.

This empirical study includes the tests on the two data sets we have, one for the NSE and the other for the LSE. I run each of the 5 models, i.e. Lagged, MA-MEDIAN, MA-MEAN, ARMA, ARMA-DECOMP, on the same testing periods of the data day by day, measure the log ratio between the actual daily volume and the predicted volume as the forecasting error, and gather all the errors to ascertain the average and the standard deviation of the errors.

As we see in Section (2.3.1), the optimal length of look-back periods of the MA-MEAN model differs to the one of MA-MEDIAN model. Therefore I always use the 24th day of the test data as the starting point for each iteration, which leaves the length of the period for building the initial moving average long enough for both models in both data sets. This ensures that all models use the same subset of the respective data set for evaluation.

For each model, I follow this evaluation process:

- Build the initial predictors for each stock using the data from the 1st to 23rd days. Note, for some cases the very first several days are actually dropped, e.g. for the Lagged model, we only need the 23rd day to forecast the volume for the first testing day.
- Take the actual total volumes of the first testing day, measure the prediction errors for each stock.
- Update the predictors with the volumes of the first testing day.
- Move on to the next day and repeat the above process.
- In the end we take average and standard deviation of all the measured errors as the final performance indicators.

2.4.1 Performance on NSE data

Table (2.3) represents the result obtained from NSE data. It shows both the average and standard deviation of the error in the unit of percentage. The Lagged model, which is supposed to be a lower bound benchmark, performed the worst, as expected. The two moving average based models got better, while the one using mean as the averaging method presented better average error, it did a bit worse with standard deviation. The ARMA model produced a 4 percent improvement on average error and also a better standard deviation, which is very impressive. Finally, the revised ARMA model with decomposition generated a further enhancement for both average and standard deviation.

Name	Average Error(%)	StDev of Error (%)
Lagged	53.372	10.208
MA-MEDIAN	51.246	8.573
MA-MEAN	50.596	9.385
ARMA	46.490	8.439
ARMA-DECOMP	46.285	8.224

Table 2.3: Performance of The Prediction Models (NSE)

2.4.2 Performance on LSE data

Table (2.4) shows the result for LSE data. The fact that the numbers here look better is because I have less data in the LSE data set than that for NSE. Other than that, it presents the same pattern as Table (2.3). Once again, the ARMA model with decomposition outperforms all the others.

Name	Average Error(%)	StDev of Error (%)
Lagged	46.600	5.704
MA-MEDIAN	40.852	5.583
MA-MEAN	40.493	5.852
ARMA	38.973	4.925
ARMA-DECOMP	38.894	4.832

Table 2.4: Performance of The Prediction Models (LSE)

The two tables confirm that our ARMA model with decomposition does make a systematic improvement for total volume prediction. The difference between ARMA-DECOMP and ARMA is not very big though. This can be explained by the fact that they both share the ARMA model as the underlying model, which determins the base of the performance. Nonethless, both Average Error and StDev of Error of ARMA-DECOMP outperforms ARMA in the two sets, which suggest good confidence of the improvement.

2.5 Conclusion

In this chapter I have studied algorithms for predicting daily total volume. A good prediction of the daily total volume gives the trading applications a view of how liquid the market will be before market open, and thus helps to optimize schedule and resource allocation for trading strategies in advance. The work in this chapter is a foundation for the further research work of intraday liquidity modeling. I have followed a data-driven principal to design the algorithms and select the optimal solution. A framework has been proposed to test and evaluate the performance of the algorithms. Empirical studies have been given to illustrate the rationale and design of each algorithm. With all the analysis, I have developed a new algorithm to forecast the daily total volume. The algorithm is based on an ARMA model with a decomposition of the market feature and the individual stock specific feature. The algorithm systematically outperforms the other classic algorithms.

Chapter 3

Improving VWAP trading strategies

3.1 Introduction

In this chapter, I model the intraday liquidity distribution in limit order book markets. I use the models to improve the VWAP trading strategies, which are commonly used to minimize transaction costs and market impact of large orders. I present massive empirical analysis on the market data from two exchanges, the National Stock Exchange (NSE) and the London Stock Exchange (LSE). Based on the findings, I develop a bidirectional VWAP trading strategy that significantly outperforms the traditional VWAP strategies.

In particular, it becomes a big challenge when the amount of intraday trading data is increasing dramatically. Bigger data comes with finer details. It enables us to have high precision of back test results, which makes the testing process better, but also more difficult to beat benchmarks because the randomness of testing results gets much smaller in large scale of data. I will show this difficulty in following sections.

I approach the challenge with an empirically driven method and develop our strategies based on the findings we get from the data. Finally, I propose a new bidirectional strategy, which I call buy/sell strategy, that systematically beats all the other ones we know of on both market data sets from two exchanges.

In industry, there is a known benchmark for such types of trading activities, which is Volume Weighted Average Price (VWAP). The executed VWAP for the given order in the target period is compared with the real VWAP for all orders in that period. The trader tries to match as closely as possible the real VWAP. Using a perfect prediction of the relative volume distribution, the difference is zero, which is the ideal result of such a task. To make the prediction better, there are things like price dynamics, volatility, spreads, etc. that may be taken into account when modeling. However, it turns out that integrating all this information into the prediction model is not an easy job. Normally, the industry applications take a static moving average curve for the intra-day relative volume distribution, derived from past historical days, as the prediction model of VWAP trading. I show analysis for such a static model and try to develop dynamic models using the information that comes from limit order books. As a result, we present a newly designed strategy that we call buy/sell strategy, this systematically performs better than the common strategy based on the static model.

Volume is an important market characteristic for practitioners who aim to lower the market impact of their trades. This impact can be measured by comparing the execution price of an order to a benchmark price. The larger the price difference, the higher the market impact. Volume Weighted Average Price (VWAP) is one such benchmark. The VWAP of a stock over a period of time is the average price paid per share during a given period.

Given a period of N trading days, for each day i the stock has price p_i and traded volume v_i, we have its VWAP P_{VWAP} defined as:

$$P_{VWAP} = \frac{\sum_{i}^{N} p_i \times v_i}{\sum_{i}^{N} v_i} \tag{3.1}$$

VWAP strategies are designed to spread orders over a period of time and target an average execution price close to or better than the VWAP benchmark. Given their popularity, VWAP strategies have been widely studied by financial researchers. One basic example of such strategies is a static strategy which attempt to achieve optimal execution by distributing an order through-out the day so as to mimic the historical-average intra-day distribution of volume. Konishi (2002) provides a static optimal strategy that minimizes the expected squared execution error and shows that price volatility tends to have a positive correlation with volume. McCulloch and Kazakov (2007) uses classic mean-variance optimization to develop strategies that exploit expected price drift by optimally front-loading or back-loading traded volume away from the minimum VWAP risk strategy.

For static strategies, the volume distribution is calculated before market open and fixed until market close. There are also dynamic strategies that can adapt during the day. Białkowski et al. (2008) decomposes traded volume into two parts: one static part reflects changes due to market evolution and one dynamic part describes the intra-day variation pattern. They use ARMA and SETAR (self-extracting threshold autoregressive model) models to depict the dynamic part. Humphery-Jenner (2011) presents a time bins strategy that adjusts volume allocation for each bin dynamically incorporating intra-day market data. Kakade et al. (2004) provides an online algorithm to capture realistic characteristics of market microstructure in a competitive ratio setting.

A more interesting one is given by Nevmyvaka et al. (2006), which applies a machine learning technology, reinforcement learning, to reduce transaction costs for order placement in a limit order book market. Its goal is different to ours in the sense that it focuses on implementation shortfall instead of VWAP. Nonetheless, one can still take inspiration from this for designing trading strategies in limit order book markets.

3.2 Statistics of Market Data

I have two data sets for analysis. One is from Reuters that consists of 20 stocks listed on the London Stock Exchange (LSE), from January 3, 2010 to August 28, 2010, counting 167 effective trading days. The other one is collected directly from the National Stock Exchange (NSE), from January 2, 2012 to April 8, 2013, counting 301 effective trading days. The LSE data does not give full depth of order book, but only quotes and the top levels of the order book in time series, while the NSE set has everything needed to reconstruct the full order book. Thus we use the NSE data as our primary data set and the LSE data just to provide secondary data points.

The NSE data consists of timestamped series of order book events. There are four types of events:

1. New Order - Any new order accepted and added to the order book.
2. Order Modification - Any order modified and added to order book. It contains the new and old image (i.e. price and quantity) of the order.
3. Order Cancellation - Any order cancelled.
4. Execution - An order in the order book gets executed fully or partially.

The dynamics of full level depth order book can be reconstructed from this data. Among the four types of events, execution represents actual trades in the market and gives the strongest information. The other three are also important in the sense that they give the full view of the market structure at any given moment. Having such a view can be critical when making trading strategy decision. in particular, the spread and the total order quantities at the top levels of the order book reveal lots of information about the market. The first intuition is to look at the statistics of the top levels of the order book.

Each event changes the order book. To gain a feel for the order book attributes in a given time period, i.e. spread, the top levels of the order book, I take the value at each time an event happens, weighted by time, sum all values up and divide by the total time. This way I have a time weighted average value for that period.

To have a sense of how liquid an instrument was traded, I calculate execution ratio as the number of shares traded per share in the top 5 levels of its order book, for both bid and offer side. This quantity is measured for each trade, and for a given time period, I use the same method mentioned above to get a time weighted average value for that period.

Table (3.1) shows the average spread and the total order quantity of the top 5 levels of the order book for each stock for the NSE data.

	Volume	Spread	Top 5 Bid Size	Top 5 Ask Size	Execution Ratio
ASIANPAINT	2351.66	40.34	91.33	191.26	0.02
GRASIM	2635.36	33.02	178.81	186.11	0.02
SIEMENS	6730.62	6.94	314.66	447.61	0.07

ULTRACEMCO	7060.24	18.91	149.92	264.91	0.07
ACC	10193.98	9.93	372.19	331.13	0.01
HEROMOTOCO	10591.09	12.07	295.27	295.13	0.01
DRREDDY	12805.48	11.59	287.55	292.35	0.01
KOTAKBANK	19517.57	6.09	892.08	793.18	0.01
RANBAXY	21720.56	3.84	1025.89	926.38	0.03
BAJAJ-AUTO	21739.89	8.71	267.29	413.04	0.01
MARUTI	22925.01	6.45	496.59	723.93	0.02
PNB	25160.10	6.26	873.52	925.05	0.01
HCLTECH	26565.87	3.42	673.13	984.71	0.04
LUPIN	27250.60	4.35	609.19	788.08	0.03
BANKBARODA	27626.91	6.28	595.00	748.60	0.02
BPCL	28464.55	3.68	1532.46	1552.98	0.06
GAIL	28844.64	3.47	1196.12	1531.09	0.11
SUNPHARMA	32468.28	4.41	590.33	847.69	0.03
M&M	38082.83	4.43	665.48	1086.76	0.03
WIPRO	38945.58	2.62	1345.01	1745.69	0.04
INFY	39302.78	6.88	426.09	412.44	0.02
LT	45528.76	5.54	525.29	625.50	0.02
CIPLA	53733.60	2.76	1947.04	2417.01	0.06
TCS	59418.02	4.02	704.73	564.90	0.03
AXISBANK	59822.68	5.44	694.54	867.45	0.02
RELINFRA	62593.95	2.82	1465.42	1399.10	0.03
SESAGOA	75323.68	1.84	5064.82	5080.30	0.05
SBIN	79600.78	6.33	507.07	651.13	0.02
COALINDIA	81136.73	1.96	2490.08	2089.48	0.10
RELIANCE	83077.68	3.02	1285.29	1318.31	0.04
HDFC	85541.39	2.67	1870.98	1263.71	0.03
ICICIBANK	88247.87	3.64	1039.14	1071.35	0.02
AMBUJACEM	93961.02	1.56	5814.86	5869.24	0.16
HINDUNILVR	98033.62	2.17	1925.25	1845.73	0.08
HDFCBANK	99023.97	2.75	2082.02	2315.21	0.04
POWERGRID	103734.34	1.19	18287.27	19569.53	0.22
TATAPOWER	107653.41	1.28	12565.51	14047.43	0.15
NTPC	109883.93	1.41	11828.96	11056.59	0.13
JINDALSTEL	129708.46	2.78	2159.43	1856.71	0.03
BHEL	143579.12	1.39	7892.17	7440.62	0.10
CAIRN	144048.58	1.49	4447.39	4021.13	0.14
ONGC	152105.87	1.53	5975.68	3991.02	0.17
BHARTIARTL	173646.95	1.56	5245.00	5717.42	0.07
IDFC	176003.06	1.27	12868.57	13363.44	0.10

TATASTEEL	204173.55	1.63	3819.27	4771.22	0.08
DLF	264665.82	1.30	13011.76	12631.98	0.09
ITC	329748.14	1.35	10149.20	8985.42	0.10
HINDALCO	384696.23	1.16	29482.15	29460.17	0.14
TATAMOTORS	428815.12	1.27	9123.32	13127.16	0.10
JPASSOCIAT	557105.36	1.08	92796.76	83794.59	0.16

As shown in Table (3.1), there is an obvious trend that the securities having more liquidity tend to have a smaller spreads and bigger sizes at the top levels of the order book.

3.3 Liquidity Strategies to Reduce Market Impact

There are cases where practitioners aim to lower the market impact of their trades. Intuitively, the larger the trade, the larger the market impact it could bring. Big market impacts can hurt the trader in the sense that it reveals private trading interests to the public, giving counter-parties the chance to take advantage of them. To reduce such risks, we first need to quantify the market impact. We can measure the market impact by comparing the execution price of an order to a benchmark price. The larger the price difference, the higher the market impact. Volume Weighted Average Price (VWAP) is one such benchmark. The VWAP of a stock over a period of time is the average price paid per share during a given period.

Definition 3.1. A trade $\sqcup_i$ is defined by its time t_i, price p_i and quantity v_i as a tuple $< t_i, p_i, v_i >$.

Given that there are N trades in total in the trading period, we have VWAP P_{VWAP} defined as:

$$P_{VWAP} = \frac{\sum_{i}^{N} p_i \times v_i}{\sum_{i}^{N} v_i} \tag{3.2}$$

With this benchmark, we can design strategies to split a large order into small chunks and spread them over a period of time to target an average execution price close to or better than the VWAP benchmark.

The market impact measured by VWAP is time related. One can have orders that give impact in 5 minutes, 1 hour, or 1 day. The VWAP approach is sufficiently flexible to measure the impact in any arbitrary time period. I focus on intraday liquidity modeling, so I use daily VWAP as our benchmark.

As we can see from the definition, the goodness of this VWAP approach is that it decouples the volume distribution and the price process. One can catch the VWAP perfectly if having a perfect prediction of the proportion series $\frac{v_{i\in 1,\ldots,N}}{\sum_i^N v_i}$, without knowing the price series $p_{i\in 1,\ldots,N}$. However, to make the perfect prediction, one has to predict both N and each v_i, and divide the target order quantity accordingly. This is very difficult, because N is the total number of trades, which depends on the decisions of the other participants, as does v_i. To minimize the impact of market noise, one has to simplify the prediction process.

Ideally, we would like to model the proportion series $\frac{v_{i\in 1,\ldots,N}}{\sum_i^N v_i}$ as a continuous distribution. Given that the intra-day trading hours start at T_{start} and ends at T_{end}, we model the distribution as $f(t)$, where $t \in (T_{start}, T_{end})$. The integration $\int_{T_{start}}^{T_{end}} f(t)\mathrm{d}t$ equals to 1. If we divide the trading hours $T_{end} - T_{start}$ into $\hat{N}$ equal length bins, where each bin b_i starts at $\hat{t}_i$ and ends at $\hat{t}_{i+1}$, we can then have prediction $\phi_i = \int_{\hat{t}_i}^{\hat{t}_{i+1}} f(t)\mathrm{d}t$ as an approximation of $\int f(t)\mathrm{d}t$.

Definition 3.2. $\phi_i = \int_{\hat{t}_i}^{\hat{t}_{i+1}} f(t)\mathrm{d}t$ is the prediction of the volume proportion traded in bin b_i, where b_i starts at $\hat{t}_i$ and ends at $\hat{t}_{i+1}$.

With the discrete prediction series ϕ_i, we have a VWAP tracking strategy for any given target order quantity. The strategy works as follows, divide the target quantity Q into $\hat{N}$ sub-orders $o_{i\in 1,\ldots,\hat{N}}$ to be executed in bin b_i, where each sub-order has quantity equal to $Q * \phi_i$. In this setting, we only have to predict the series ϕ_i. We can either predict all ϕ_i in advance, which we call static prediction, or we can use the intra-day information up until the time before $\hat{t}_i$ and dynamically predict ϕ_i, which we term dynamic prediction.

One can have a naive strategy that simply distributes the order evenly during the day. We will first check this naive linear strategy as a basic benchmark. Figure (3.1) presents the distribution of volume using this strategy for 4 sample stocks listed in NSE, i.e. ACC, PNB, RELIANCE and WIPRO which are marked as title for each sub-figure. The 4 stocks are used for illustrative purposes throughout this chapter.

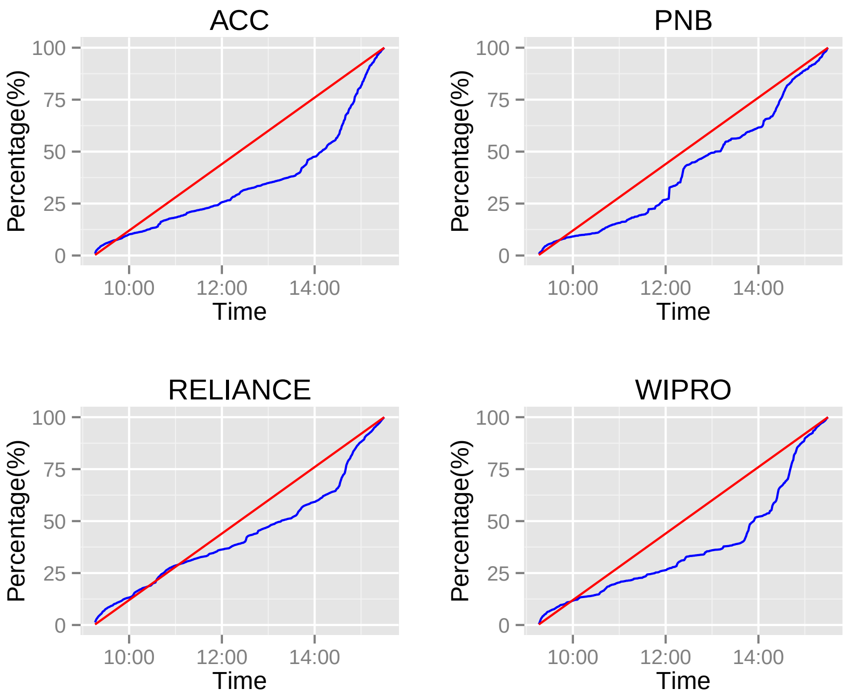


FIGURE 3.1: Volume Curve - Linear

In Figure (3.1), the x-axis represents time in market hours while the y-axis tells how much percent of the target order quantity has been traded for a given time. The blue curves show the real accumulative volume distribution. The red curves stand for the prediction of the accumulative volume distribution made by the forecasting strategy. In this case all red curves are straight lines, because the naive strategy simply distributes the order evenly.

It appears from Figure (3.1) that there tends to be a pattern that trades occur more around market open and close than lunch time. To verify this idea, I break the intra-day trading hours into three equal length periods (125 minutes for each period), and measure the average percentage of volume for each period over 10 days (2012/12/13 - 2012/12/27). Table (3.2) presents the measurement for the stocks listed in the NIFTY index. We can see that the period for lunch time has obviously less volume than the other two, about 7 percent less than the morning period and 20 percent less than the afternoon period.

	Morning (%)	Lunch Time (%)	Afternoon (%)
ACC	31.10	20.97	47.93
AMBUJACEM	30.89	24.16	44.95
ASIANPAINT	37.00	19.52	43.48
AXISBANK	32.82	24.14	43.05
BAJAJ-AUTO	33.45	23.84	42.71
BANKBARODA	29.18	24.42	46.40
BHEL	34.03	22.51	43.46
BPCL	21.63	25.12	53.25
BHARTIARTL	30.84	24.17	44.99

CAIRN	27.93	20.59	51.48
CIPLA	32.79	18.24	48.97
COALINDIA	31.89	23.19	44.92
DLF	32.83	22.26	44.91
DRREDDY	26.87	24.84	48.29
GAIL	24.83	23.84	51.33
GRASIM	30.84	25.04	44.12
HCLTECH	26.19	24.21	49.60
HDFCBANK	26.32	25.37	48.31
HEROMOTOCO	31.47	20.65	47.88
HINDALCO	30.61	25.54	43.84
HINDUNILVR	30.31	23.52	46.17
HDFC	32.46	22.69	44.85
ITC	27.85	27.15	45.01
ICICIBANK	30.98	23.86	45.17
IDFC	27.74	27.89	44.37
INFY	29.65	26.03	44.31
JPASSOCIAT	31.12	24.48	44.41
JINDALSTEL	33.94	21.22	44.83
KOTAKBANK	24.85	24.27	50.87
LT	35.55	26.62	37.82
LUPIN	26.93	26.07	47.00
M&M	30.45	24.57	44.98
MARUTI	36.31	22.20	41.49
NTPC	25.47	24.42	50.12
ONGC	30.42	24.68	44.90
POWERGRID	25.37	25.34	49.29
PNB	32.19	26.52	41.28
RANBAXY	35.51	18.82	45.67
RELIANCE	30.76	24.19	45.05
RELINFRA	33.53	23.30	43.17
SESAGOA	30.25	21.18	48.57
SIEMENS	32.80	21.83	45.37
SBIN	32.56	22.05	45.39
SUNPHARMA	29.84	24.19	45.97
TCS	27.22	23.41	49.37
TATAMOTORS	40.04	22.66	37.29
TATAPOWER	32.84	20.80	46.36
TATASTEEL	30.38	22.61	47.01
ULTRACEMCO	27.94	27.62	44.45
WIPRO	29.25	23.08	47.67

Average	30.56	23.60	45.84

Table 3.2: Volume Distribution (NSE)

	Morning (%)	Lunch Time (%)	Afternoon (%)
SHP.L	24.18	17.44	58.38
XTA.L	29.99	22.14	47.88
EXPN.L	22.89	21.51	55.60
HSBA.L	27.83	19.20	52.96
IMT.L	29.79	18.27	51.94
LSE.L	24.74	20.38	54.88
TSCO.L	30.71	18.99	50.29
AAL.L	31.90	20.54	47.56
STAN.L	31.59	18.86	49.55
RBS.L	35.80	22.98	41.22
TATE.L	26.05	19.57	54.38
LLOY.L	34.70	22.71	42.58
VOD.L	30.39	18.19	51.42
LGEN.L	28.04	27.50	44.46
NXT.L	29.05	18.76	52.19
UU.L	22.62	20.30	57.08
GSK.L	28.95	18.47	52.58
BAES.L	25.49	27.13	47.38
BATS.L	29.38	18.83	51.79
RIO.L	33.12	22.16	44.72
Average	28.86	20.70	50.44

Table 3.3: Volume Distribution (LSE)

Therefore, I take the extraction of this intra-day seasonality pattern as the first step for better volume distribution modeling. One natural and common approach used by the industry is to divide the intra-day trading hours to $\hat{N}$ equal length bins, as described above, and calculate the average volume percentage distributed to each bin b_i over the last L days. This gives a moving average profile with parameters $\hat{N}$ and L. This approach builds the profile in advance of market open using market data of historical days and does not change or update the profile during trading hours. It is simple to calculate and maintain the profile as it is just a plain fixed length array. This approach cannot leverage any new information arriving after market open, as the profile has been built in advance.

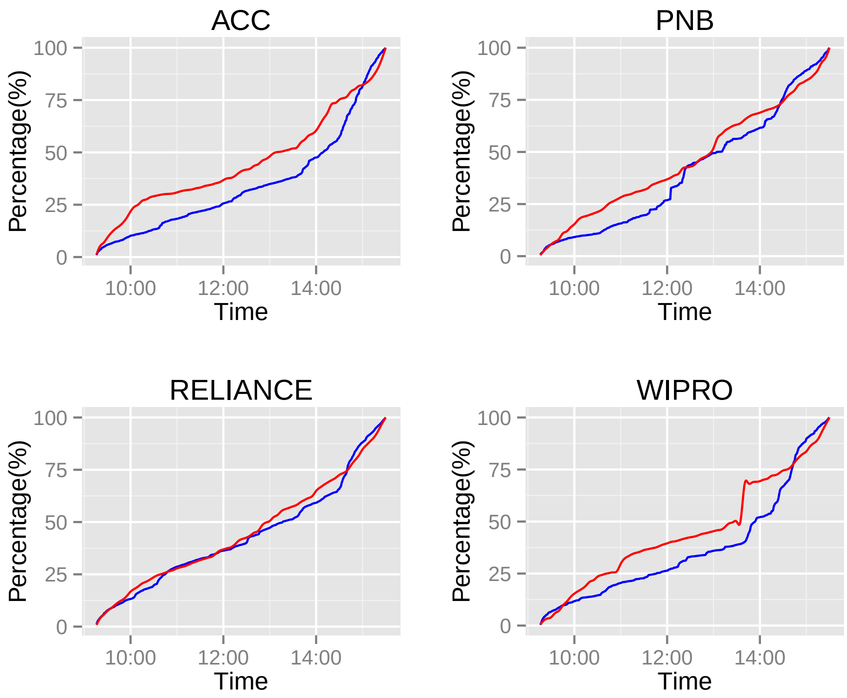


FIGURE 3.2: Volume Curve - Static Moving Average

Figure (3.2) presents examples for the static moving average distribution curves, with a setting of $N = 375$ (5-minute bin) and $L = 30$. The x-axis and y-axis are time in market hours and how much percent of the target order quantity has been traded for a given time. The blue curves represent the real accumulative volume distribution. The red curves stand for the prediction of the accumulative volume distribution made by the static moving average strategy.

Both the linear and static moving average approaches do not require knowledge of the total volume of the target day, they are essentially just relative curves. Then the question that comes to our mind is, what if we know the total volume that will happen in the market in advance? Can we leverage this information to make a prediction for the intra-day distribution? This question can be formulated as: knowing $\hat{V}$ as the total volume at market close time, can we predict the relative volume distribution series $r_{i \in 1,\ldots,\hat{N}}$?

Ideally, given the intra-day history $v_{i \in 1,\ldots,i-1}$, if we can predict the real volume v_i in each bin b_i, we have a perfect prediction of the distribution too, which is $r_i = \frac{v_i^{real}}{\hat{V}}$. We can use techniques for time series like ARMA or GARCH as the predictor of v_i^{real}. As what we really care about here is the relative distribution, the goodness of such predictors is not critical if we have a large N. We use a naive predictor, which uses v_{i-1}^{real} as the prediction of v_i^{real}. This gives us a 1-bin "lagged" prediction of the volume distribution, as shown in Figure (3.3).

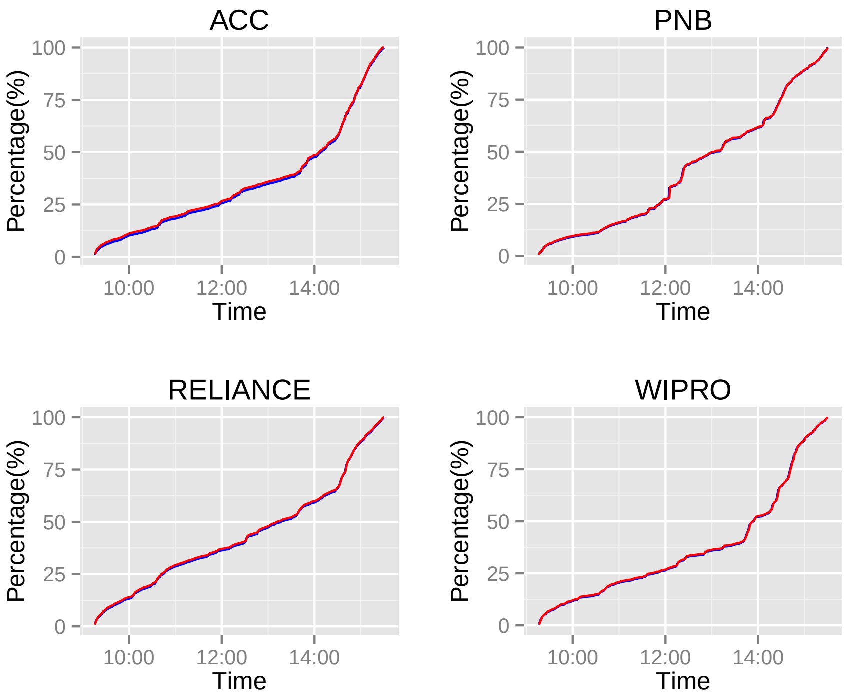


FIGURE 3.3: Volume Curve - Lagged with Known Total Volume

Figure (3.3) follows the same rule as the two Volume Curve figures shown before. The x-axis and y-axis are time in market hours and how much percent of the target order quantity has been traded for a given time. The blue curves represent the real accumulative volume distribution. The red curves stand for the prediction of the accumulative volume distribution made by the lagged strategy. We can see that by knowing the total volume, it is much easier to make prediction for the distribution even with a simple 1-bin lagged forecasting strategy.

Now we have several models for volume distribution. To understand how well of each of them performs, we need to be able to quantify their performance. As explained in the beginning of this section, we use daily VWAP as the benchmarking method. In our data set, we have full order book events for every historical day, this enables us to calculate the accurate real VWAP for each day.

Our benchmark price is the exact VWAP using real market data for every trade, P_{VWAP}. As described above, with each strategy we have a prediction series $\phi_{i \in 1,\ldots,\hat{N}}$ for bins $b_{i \in 1,\ldots,\hat{N}}$. To evaluate the performance of a strategy s, we assume a target order quantity Q. Our aim is to trade Q over the whole trading period as sub orders, and have the realised VWAP $\hat{P}_{VWAP}$ to catch P_{VWAP}. Given that with strategy s, we have the prediction series of volume proportion $\phi_{i \in 1,\ldots,\hat{N}}$. Having $\hat{p}_i$ as the reference price (which we will explain later), our realised VWAP is:

$$\hat{P}_{VWAP} = \frac{\sum_{i}^{\hat{N}} \hat{p}_i \times Q \times \phi_i}{Q} = \sum_{i}^{\hat{N}} \hat{p}_i \times \phi_i \tag{3.3}$$

Thus, we have a realised VWAP for our orders as $\hat{P}_{VWAP}$. The difference between P_{VWAP} and $\hat{P}_{VWAP}$, measured in ten thousandth, is the slippage of the tracking strategy. I use this slippage as the quantified performance indicator, the smaller the slippage the better the strategy is.

Definition 3.3. The performance of a VWAP tracking strategy is measured in slippage s, where $s = 10000 * |\frac{P_{VWAP} - \hat{P}_{VWAP}}{P_{VWAP}}|$

We run a VWAP tracking task for each model, and use the slippage between the realised traded VWAP and the real VWAP as the performance result. Due to the randomness in the nature of financial market data, we take the average slippage over T days of historical data as the overall performance indicator.

One challenge of this benchmarking process is that we cannot have accurate trading prices because we do not really trade in the real market. Whenever we issue an order, there is no exchange to match our order and decide the price. This means:

1. There is no difference whether we issue a market order or a limit order.
2. There is no guarantee that our orders can be executed at an exact time.
3. The price we use for our orders can be different to the one we could get in the real market.

All these have impact on the VWAP evaluation process and can result in errors. While the first two are beyond our control in our historical-data-based experiment setting, we do have some control over the last one.

The price process also plays an important role in VWAP calculation. One does not have to make a perfect volume distribution prediction to still have a good VWAP evaluation, if the exact price process is known. To mimic this impact, we have to be able to match the real price for the orders as close as we can. Given that we have a full track of every trade in the historical data, we can calculate a time weighted average price for each bin, and use that as our reference price.

Recall that in our performance evaluation settings, we have $\hat{N}$ bins series $b_{i \in 1,\ldots,\hat{N}}$, where b_i spans time period $[\hat{t}_i, \hat{t}_{i+1})$. We define the reference price $\hat{p}_i$ for bin b_i as the time weighted average price:

Definition 3.4. Select all trades $\sqcup_i$ happened in $[\hat{t}_i, \hat{t}_{i+1})$, i.e. $\hat{t}_i \leq t_i < \hat{t}_{i+1}$, $\hat{p}_i = \frac{\sum_{\hat{t}_i \leq t_i < \hat{t}_{i+1}} p_i \times (t_i - \hat{t}_i)}{\hat{t}_{i+1} - \hat{t}_i}$

In Definition 3.4, $\hat{p}_i$ represents the expected price for a trade falling randomly in the period $[\hat{t}_i, \hat{t}_{i+1})$. Thus we take $\hat{p}_i$ as the reference price for the order generated by a strategy for bin b_i.

There are other options for getting the reference price $\hat{p}_i$ for bin b_i, such as simple arithmetical average over all trades in the period, the price of the first or the last trade in the period, or volume weighted average price for the period. The first two are easy to calculate, but probably less realistic. VWAP is a better choice as it represents the interest of the market participants in the period, and gives more sense in economic over the other two because it takes liquidty into account. Therefore we pick up VWAP as the reference price in our setting.

3.4 VWAP Strategy Design

In this section, I develop VWAP strategies by using intraday volume distribution prediction. The goal of a VWAP strategy is that, given a target total order quantity, split the quantity into sub-orders and execute them in market, then try to minimize the difference of the executed VWAP of the sub-orders and the real VWAP of the whole trading period. Note that ideally, one wants to actually beat the real VWAP, i.e. have lower executed VWAP if it is a buy order, or have higher executed VWAP if it is a sell order. In this chapter, I focus on volume modeling, which can only help to catch the real VWAP in best case. I do not distinguish whether the order is buy or sell and just try to minimize the absolute difference of the executed VWAP and the real VWAP.

3.4.1 VWAP Strategies with Intraday Volume Distribution Prediction

With the intraday volume distribution models described in previous section, we can already make up VWAP strategy for each of the models, i.e. split the target order quantity exactly as the distribution curve produced by the model. In this section, we first examine how does these strategies perform under the evaluation approach we proposed in last section. This gives us a foundation to let us have a view on how much improvement we can expect to achieve from goot design of strategies. Then we do analysis of how intraday VWAP envoles when we observe volume shock, which reveals the difficulty of beating traditional VWAP strategy based on static moving average curve and shed light on further work in following sections.

Our data set is from the National Stock Exchange (NSE), India. Before testing, we build static moving average curves of intraday volume distribution for each stock using 3 months data from September 13, 2012 to December 12, 2012, which has 60 trading days in total. The test data begins on December 13, 2012 and ends at April 8, 2013, giving 78 days in total. As the liquidity of stocks can have an impact, we only test on stocks listed in the NIFTY index, which gives 50 stocks.

We run the VWAP algorithms for each testing day, and collect the slippage between the VWAP traded by the algorithm with the real VWAP. The slippage is measured in basis points (bps), which is $\frac{1}{10000}$. For each algorithm, we look at the distribution of the slippages over all 50 stocks and 78 days and take the mean as the final performance indicator.

First, we would like to evaluate a theoretical strategy, which uses exactly the same proportion series from the test data as the prediction:

$$\phi_i = \frac{v_i}{\sum_{i}^{\hat{N}} v_i} \tag{3.4}$$

That said, this strategy gives a perfect prediction of the intra-day volume distribution. The only difference between the realised VWAP from this strategy and the real VWAP comes from the reference price we use for bins $b_{i \in 1,\ldots,\hat{N}}$. The performance of this strategy tells us two things:

1. The goodness of the reference price approach. We expect the slippage to be minimal.
2. The best performance we can reach for this VWAP approach. We take the result as the upper bound of the performance test for all strategies.

Figure (3.4) shows the performance of this theoretical strategy on the NSE data. The the average slippage of this strategy on NSE data is 0.63 bps (i.e. basis points), which is presented in the title next to the strategy name Theoretical. The graph is a standard density plot that shows the shape of the distribution of slippages. The x-axis stands for different values of slippage, while the y-axis shows the density of each slippage value in the whole set, e.g. 0.10 corresponds to 10% of the whole data set. We can see that the distribution of the slippage of all samples is quite dense around zero. This result confirms that the reference price approach is a good one.

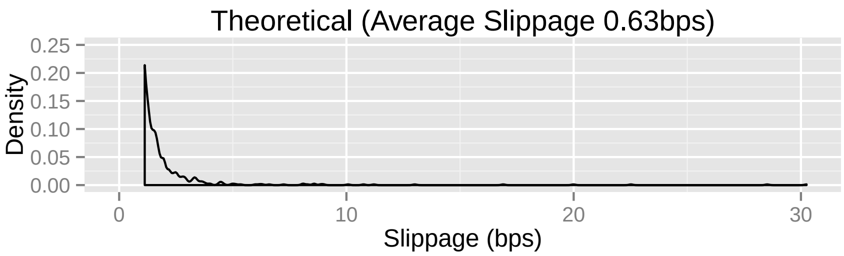


FIGURE 3.4: VWAP Slippage - Theoretical (NSE)

Figure (3.5) shows the performance of the Theoretical strategy on the LSE data. The average slippage is 4.82 bps.

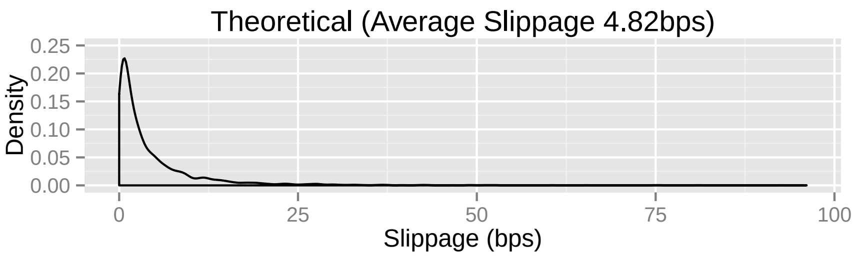


FIGURE 3.5: VWAP Slippage - Theoretical (LSE)

Next, let us look at the performance of the simplest strategy, the linear one. Figure (3.6) outlines the performance of this strategy on the NSE data. The average slippage is 13.06 bps. This is a much bigger number than the one (0.63 bps) we see in Figure (3.4).

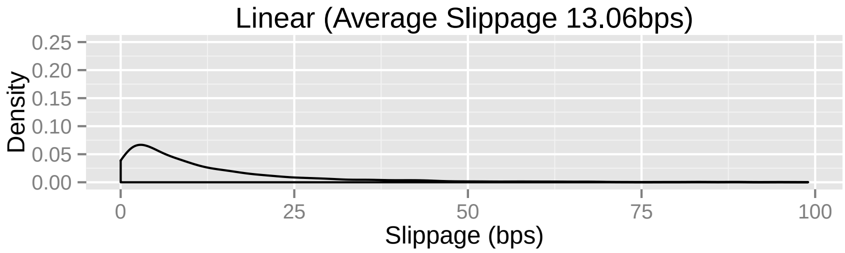


FIGURE 3.6: VWAP Slippage - Linear (NSE)

Figure (3.7) outlines the performance of this strategy on the NSE data. The average slippage is 17.21 bps. Again it is a much bigger number than the one (4.82 bps) we see in Figure (3.5).

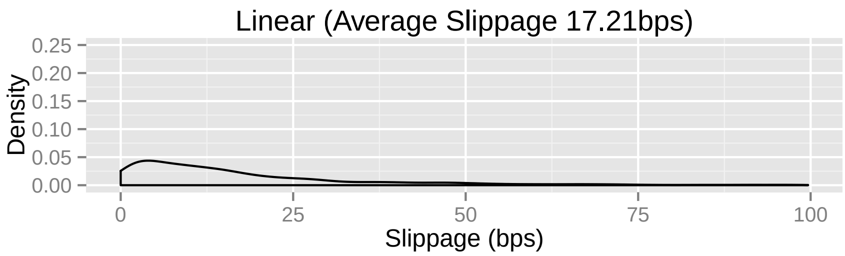


FIGURE 3.7: VWAP Slippage - Linear (LSE)

As explained above, this strategy does nothing more than simply distribute the order quantity uniformly over the trading period. We expect that this is the one that has the worst performance, as a lower bound of the performance that a strategy can have.

As shown in Section (3.3), there is a seasonal pattern in the intra-day volume distribution. It appears to have more volume traded in the morning and afternoon period than the lunch time period. We have a strategy that is based on a moving average profile using L historical days, which is commonly used in industry. As described in Section (3.2), this strategy allocates $\frac{\sum_{j \in 1,\ldots,L} \frac{v_{i,j}}{V_j}}{L}$ for bin b_i, where $\frac{v_{i,j}}{V_j}$ is the volume proportion of bin b_i for the jth day.

Obviously, this strategy calculates $\hat{N}$ proportion values for $\hat{N}$ bins in advance, thus we call it static moving average strategy. Depending on L, there can still be noise in the $\hat{N}$ values, e.g. a day with a big volume shock in a short time (several bins) can introduce noise for those bins in the output profile. Although we can enlarge L to minimize this impact, big L may have impact on the seasonality extraction too, as the seasonality can change over a long time. A further improvement for this strategy could be performing regression on the $\hat{N}$ bins to have a smooth curve.

Figure (3.8) demonstrates the differences between the static moving average strategy and the ones with regression. We try regression with polynomial curves of order 2, 4 and 8.

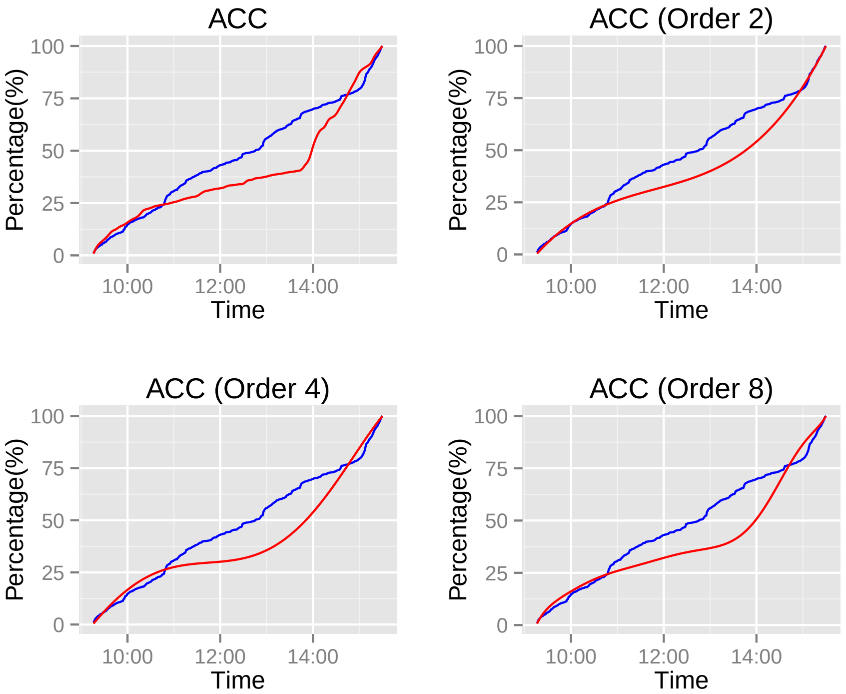


FIGURE 3.8: Volume Curve - Static Moving Average and Regressions

In Figure (3.9), we see that the static moving average strategy gives 9.54 bps average slippage, which is much better than the 13.06 bps from the linear strategy. Surprisingly, there is no gain from doing regression, it even hurts. Considering the computational cost of doing regression, this is truly not a good improvement.

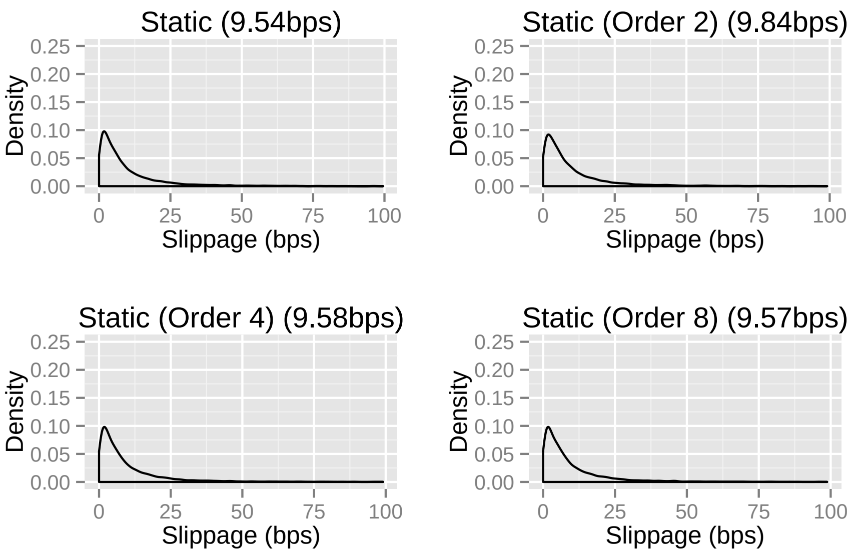


FIGURE 3.9: VWAP Slippage - Static Moving Average (NSE)

Figure (3.10) shows the performance on the LSE data set, there is no obvious improvement as well with the regressions.

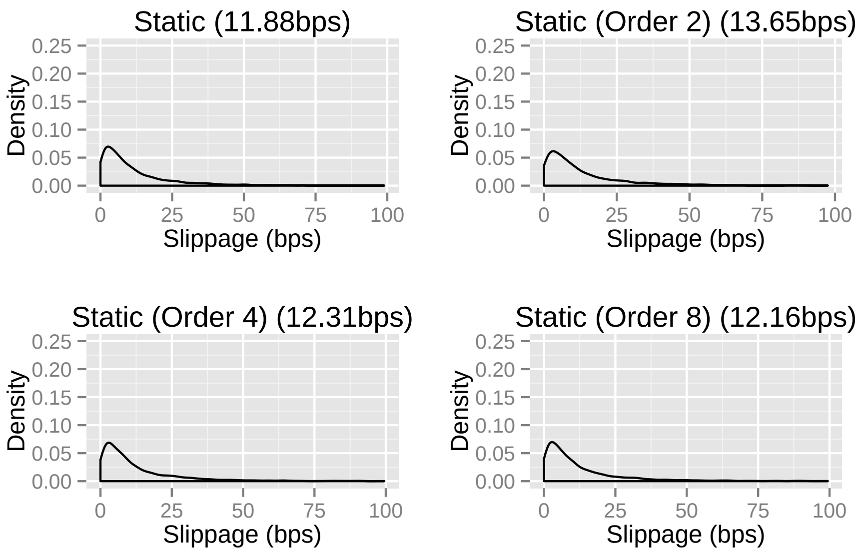


FIGURE 3.10: VWAP Slippage - Static Moving Average (LSE)

Our static moving average strategy builds the static bins curve in advance for each stock individually. While each stock has its intra-day volume distribution seasonality, the whole market can have seasonality too, this can be caused by macro economic news or the mutual impact of the stocks on each other. Can we extract and leverage this market level seasonality? Białkowski et al. (2008) provide an idea to separate the seasonality into two parts, one called common feature and another called stock specific feature. They group the proportion bins series for the targeted stock into a matrix, and compute the principal components as the common feature. We applied a similar approach to our data set to check the performance.

Having S stocks $s_{k\in 1,\ldots,S}$ in our data set, we have pre-calculated the proportion bin series $\{\phi_{k,i\in 1,\ldots,\hat{N}}\}$ for stock s_k. This gives a $\hat{N}\times S$ matrix X. We take the first principal component of this matrix as the predicted volume distribution for every stock.

Having the matrix X as:

$$X = \begin{pmatrix} \phi_{1,1}, \phi_{2,1}, \ldots, \phi_{S,1} \\ \phi_{1,2}, \phi_{2,2}, \ldots, \phi_{S,2} \\ \vdots \\ \phi_{1,\hat{N}}, \phi_{2,\hat{N}}, \ldots, \phi_{S,\hat{N}} \end{pmatrix} \tag{3.5}$$

We caculate vector w as the eigenvector of matrix $X^T X$. The first component $\dot{x}$ of X is thus $\{\dot{x}_{i\in 1,\ldots,\hat{N}}\} = X \cdot w$. We then normalize $\{\dot{x}_i\}$ to get the series $\{\hat{\phi}_i = \frac{\dot{x}_i}{\sum_i^{\hat{N}} \dot{x}}\}$. We then use $\{\hat{\phi}_i\}$ as the prediction of proportion series for every stocks.

We call this strategy as the principal component strategy, it is different from the static moving average one in that it extracts the market level seasonality instead of the stock specific one, and it uses a single prediction of volume distribution for all stocks.

Figure (3.11) shows the performance of the principal component strategy on the NSE data. The average slipppage is 10.23 bps, which is worse than the static moving average strategy. This indicates that the market level seasonality has less impact on liquidity than the individual stock specific seasonality.

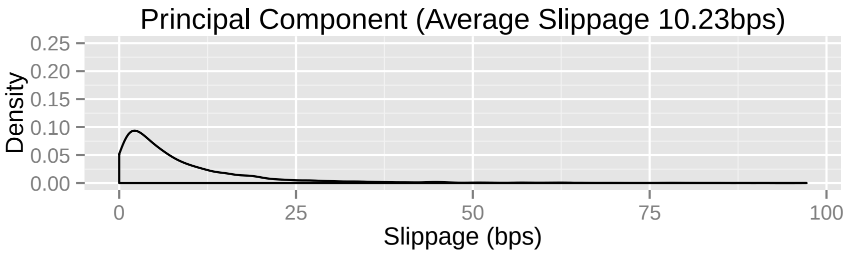


FIGURE 3.11: VWAP Slippage - Principal Component (NSE)

Figure (3.11) presents the performance of the principal component strategy on the LSE data. Again the average slippage (14.59 bps) is worse than the static moving average strategy.

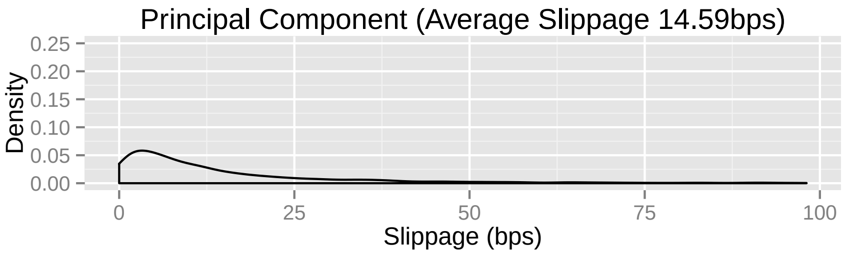


FIGURE 3.12: VWAP Slippage - Principal Component (LSE)

The way Białkowski et al. (2008) model the stock specific feature is to use ARMA or SETAR. They use a dynamic approach to update the prediction using intra-day information. The idea is to make the prediction ϕ_i one-step ahead. For each bin b_i, they use ARMA or SETAR to predict volume series for all remaining bins, $\hat{v}_{i,j \in i,\ldots,\hat{N}}$ where $\hat{v}_{i,j}$ means the prediction of volume for bin b_j at the starting time of bin b_i. Then the one-step ahead prediction for the proportion ϕ_i for bin b_i can be made as $\hat{\phi}_i = \frac{\hat{v}i,i}{\sum_{j \in i,\ldots,\hat{N}} \hat{v}_{i,j}}$. This proportion prediction $\hat{\phi}_i$ is only applied to the remaining volume to trade before the starting time of bin b_i. With $\hat{\phi}_i$ one can decide the volume to trade in bin b_i is $\hat{\phi}_i \times (Q - \sum_{j}^{i-1} \hat{\phi}_j \times Q)$.

However, we could not get good results with these two strategies on the data sets we have. Figure (3.13) shows the performance of the two strategies on the NSE data. The

ARMA strategy has an average slippage 11.88 bps, while the SETAR strategy has an average slippage 10.90 bps. Both are worse than the static moving average strategy.

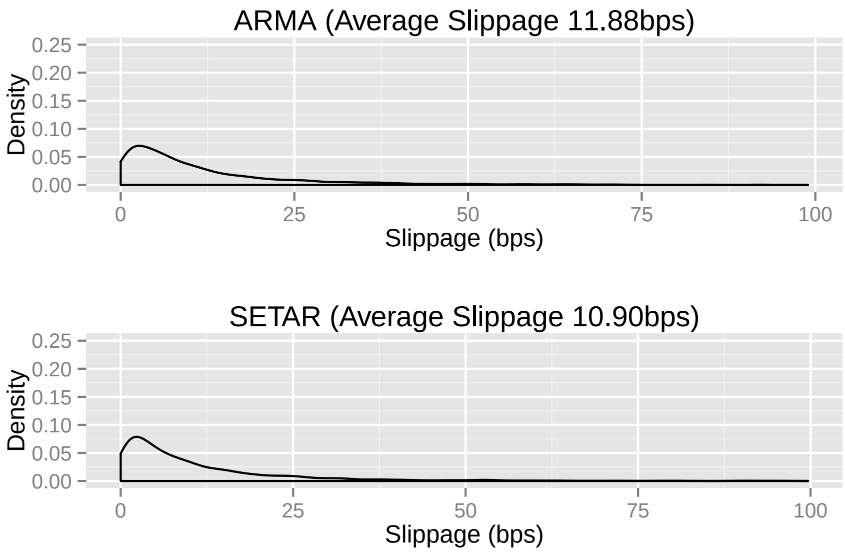


FIGURE 3.13: VWAP Slippage - ARMA and SETAR (NSE)

Figure (3.14) gives the results on the LSE data. The ARMA strategy has an average slippage 16.59 bps, while the SETAR strategy has an average slippage 19.96 bps. Again, both are worse than the static moving average strategy.

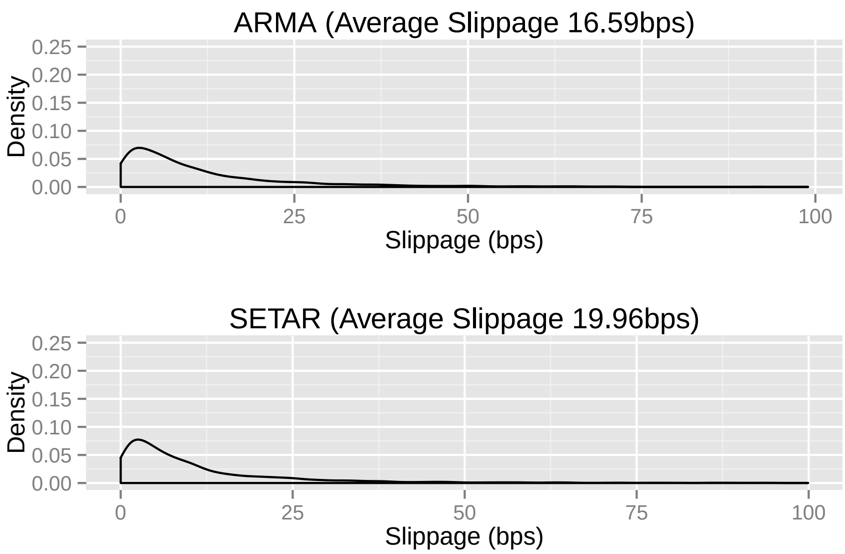


FIGURE 3.14: VWAP Slippage - ARMA and SETAR (LSE)

As we can see from the VWAP definition Equation (3.1), the trick of the VWAP tracking strategy is that up until the starting time of bin b_k, the part for traded order quantity in the final VWAP has been fixed. Let $\dot{P}^{[1,k)}_{VWAP}$ denotes the VWAP for executed part of the target order before the starting time of bin b_k, we have:

$$\dot{P}^{[1,k)}_{VWAP} = \frac{\sum_{i}^{k-1} \hat{p}_i \times v_i}{\sum_{i}^{\hat{N}} v_i} = \sum_{i}^{k-1} \hat{p}_i \times \phi_i \tag{3.6}$$

At this stage, no matter what we do for the remaining proportion prediction, we can not change the $\dot{P}^{[1,k)}_{VWAP}$. The larger k is, the smaller the impact we can have in the remaining prediction series. Let $\dot{P}_{VWAP}$ denote the final VWAP of the executed order, we have:

$$\dot{P}_{VWAP} = \frac{\sum_{i}^{\hat{N}} \hat{p}_i \times v_i}{\sum_{i}^{\hat{N}} v_i} = \dot{P}^{[1,k)}_{VWAP} + \sum_{i=k}^{\hat{N}} \hat{p}_i \times \phi_i \tag{3.7}$$

The question is, if we made a forecasting error in the early stage (before bin b_k) which let $\dot{P}^{[1,k)}_{VWAP}$ deviates from $P^{[1,k)}_{VWAP}$, what is the best thing we should do later? Naturally, we have two options:

1. Try to beat VWAP in the next bins $b_{i \in k,\ldots,\hat{N}}$, that said, to get a better price $\dot{P}^{[k,\hat{N}]}_{VWAP} < P^{[k,\hat{N}]}_{VWAP}$ for a buy order, and $\dot{P}^{[k,\hat{N}]}_{VWAP} > P^{[k,\hat{N}]}_{VWAP}$ for a sell order.
2. Forget about the mistake as it is sunk cost, still try to catch $P^{[k,\hat{N}]}_{VWAP}$ for the next bins.

Option 1 is undoubtedly a difficult task. The reason we choose VWAP as our benchmark for market impact, and the one for which it is usually used in industry, is that it decouples the price process and the volume distribution, which releases people from the complicated modeling exercise of price process. To have a better execution price, one has to take the price process into account. Recall that as per Equation (3.1), we can not beat VWAP without being able to predict the price series $p_{i \in 1,\ldots,\hat{N}}$ even if we know the exact volume proportion series $v_{i \in 1,\ldots,\hat{N}}$. Therefore, this option is outside of our interest.

Option 2 sounds more reasonable. It seems to imply that we do not get much from the intra-day information history. When we make decisions for bin b_i, the history we have observed at $b_{i \in 1,\ldots,i-1}$ seems to contribute to nothing more than $P^{[1,k)}_{VWAP}$. Is this true? Let's look at a simple case, shown in Figure (3.15).

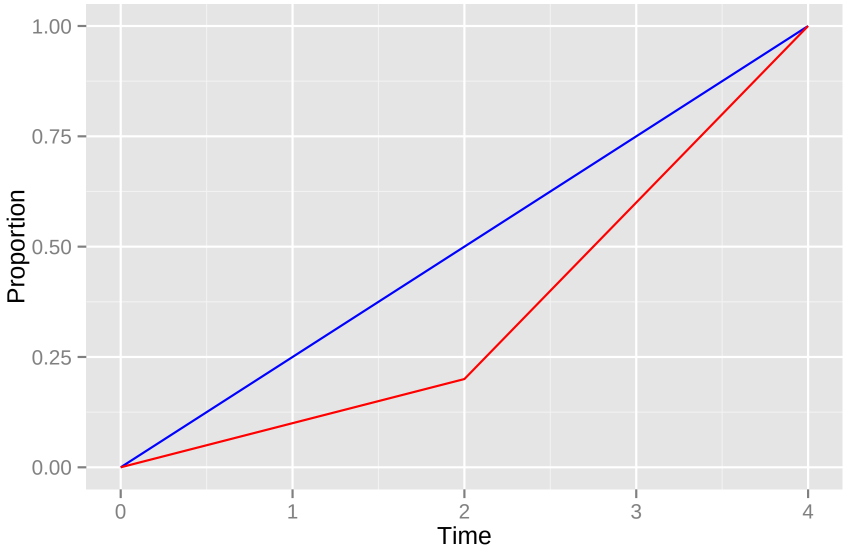


FIGURE 3.15: Manmade Volume Distribution

Let's assume we have a day where the volume is distributed as the red line in Figure (3.15). The blue line is the prediction made by our VWAP trading strategy (here it is simply the linear one). Having a bin length equal to 1, we have 4 bins in this example. Let the reference price series for the 4 bins are 100, 99, 100, 101. Now, we have the real VWAP (volume distributed as the red line) as:

$$P_{VWAP} = 0.1 \times 100 + 0.1 \times 99 + 0.4 \times 100 + 0.4 \times 101 = 100.3$$

And the traded VWAP for our strategy as:

$$\dot{P}_{VWAP} = 0.25 \times 100 + 0.25 \times 99 + 0.25 \times 100 + 0.25 \times 101 = 100$$

Which gives us a slippage as 30bp. However, if we split the bins into two periods, one for bin b_1 and b_2, and another for bin b_3 and b_4, what does the VWAP look like for each period? For period 1, we have the real VWAP as:

$$P^{b_1,b_2}_{VWAP} = \frac{0.1 \times 100 + 0.1 \times 99}{0.1 + 0.1} = 99.5$$

And the traded VWAP for our strategy as:

$$\dot{P}^{b_1,b_2}_{VWAP} = \frac{0.25 \times 100 + 0.25 \times 99}{0.25 + 0.25} = 99.5$$

Which means we have caught the exact VWAP by the end of period 1. For period 2, we have the real VWAP as:

$$P^{b_3,b_4}_{VWAP} = \frac{0.4 \times 100 + 0.4 \times 101}{0.4 + 0.4} = 100.5$$

And the traded VWAP for our strategy as:

$$\dot{P}^{b_3,b_4}_{VWAP} = \frac{0.25 \times 100 + 0.25 \times 101}{0.25 + 0.25} = 100.5$$

Again, we have caught the exact VWAP for period 2 (individually). The 30bp slippage comes from the fact that the volume traded in period 2 is 4 times as much as the volume traded in period 1. The conclusion from this simple example is that, at time t, simply make prediction based on the shape of the distribution curve observed in history $[0, t)$ is not enough to avoid slippage. In fact it could lead to bad slippage just like the example shows.

Next question is, can we do it better? For the same example, assume that we were at any time t where $2 < t < 4$, we had realised that the traded VWAP has slipped from the real one. What is the best decision we could make? Clearly, without being able to predict the price process, it is a very hard decision. Also, don't forget we have a VWAP strategy, which gives a prediction as the blue line represented in Figure (3.15), and we had been following it up until t. For example, say we were at time $t = 3$, where we have the real VWAP as:

$$P^{[0,3)}_{VWAP} = \frac{0.1 \times 100 + 0.1 \times 99 + 0.4 \times 100}{0.1 + 0.1 + 0.4} = 99.83$$

And the traded VWAP as:

$$\dot{P}^{[0,3)}_{VWAP} = \frac{0.25 \times 100 + 0.25 \times 99 + 0.25 \times 100}{0.25 + 0.25 + 0.25} = 99.67$$

We would see a slippage as 16bp. If we draw the partially observed volume distribution and compare it with the prediction our strategy made, we would see the curves shown in Figure (3.16).

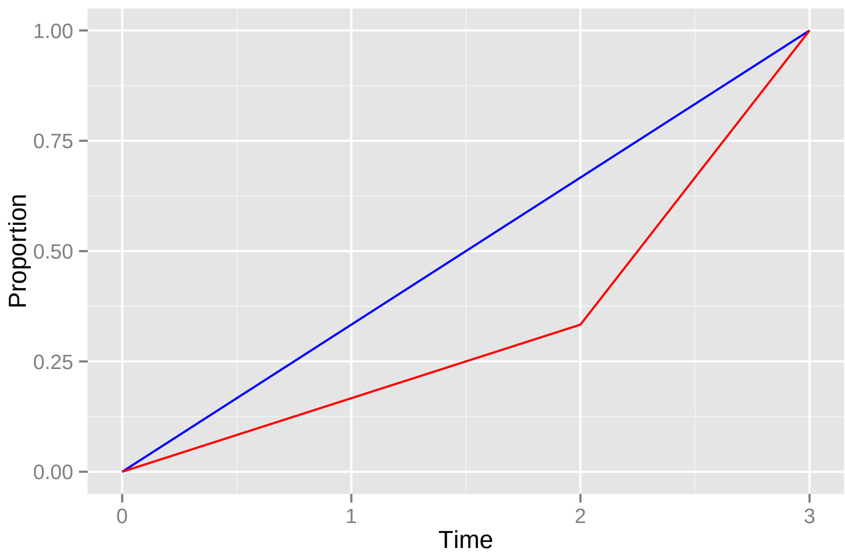


FIGURE 3.16: Manmade Volume Distribution (Partial)

We would have observed the volume traded bin b_3 is much bigger than predicted (given that we saw the volume traded in bin b_1 and b_2 was 0.1, and our prediction of the volume distribution is linear, we would have predicted the volume traded in bin b_3 is 0.1, which is much smaller than the real value 0.4). So, at time $t = 3$, we would know 3 points from the intra-day history:

1. We made a forecasting error in volume distribution prediction and got a 16 bp slippage as penalty.

2. The volume has been increasing faster than predicted since time $t = 2$.

3. The volume has been boosted linearly so far.

As stated before, we do not try to beat VWAP at any stage, so point 1 is not very useful for us. With points 2 and 3, do we expect the volume shock to keep going for the rest of the day? This sounds like a good question to investigate, but remember that what we want is to catch the real VWAP for the whole remaining period. That requires the ability to predict the distribution curve as a whole, in this case it is a linear curve. Thus, without knowing the remaining total volume, it is pointless to discuss how long the volume shock would last.

Coming back to the dynamic strategy used by Białkowski et al. (2008), the prediction it makes for bin b_i is $\hat{\phi}_i = \frac{\hat{v}_{i,i}}{\sum\limits_{j \in i,\ldots,\hat{N}} \hat{v}_{i,j}}$, where $\sum\limits_{j \in i,\ldots,\hat{N}} \hat{v}_{i,j}$ can translate into the remaining total volume $\hat{V}_i$. In this way, this strategy translates the prediction of proportion $\hat{\phi}_i$ into two predictors for bin b_i, one is a one-step ahead prediction for the volume in b_i as $\hat{v}_{i,i}$, and the other is the remaining total volume $\hat{V}_i = \sum\limits_{j \in i,\ldots,\hat{N}} \hat{v}_{i,j}$. Then it makes the prediction as $\hat{\phi}_i = \frac{\hat{v}i,i}{\hat{V}_i}$. While ARMA and SETAR are hopefully good at the short-time, one-step ahead prediction $\hat{v}_i$, it is doubtful that they can make good predictions for long-term value $\hat{V}_i$. Given that both the values are updated dynamically at each step, the resultant curve shape can have quite some noise. All these challenge the goodness of the strategy.

So far, it seems that at time t we have to focus on the whole curve shape of the rest of the day. The task does not seem to become easier given that we have known the intra-day history up until t. The next questions are, is that really helpful that if we can make good prediction of the volume distribution for the rest of day? How much can it contribute to the VWAP benchmark? Does t matter? We will address these questions in the next section.

3.4.2 Impact of Total Volume

The strategies we have developed in last section share two common issues:

1. They produce prediction of volume distribution in advance of market open, and do not adjust the prediction after that.

2. They do not require prediction of the total volume traded at market close.

For the first issue, one can imagine using the intra-day information to dynamically adjust the prediction. Indeed, Białkowski et al. (2008) uses ARMA and SETAR models to achieve that. However, we could not get good results with those models on our data set, as mentioned in Section (3.4.1).

The second issue does look like something one can improve. Intuitively, one should be able to make a prediction very close to the real distribution when knowing the total volume in advance. If this assumption holds, we can simplify the problem to be predicting the total volume instead of the whole curve. Predicting the whole curve would be a much more difficult task.

We start with two cheating algorithms to verify the idea. For both algorithms we presume that we can make perfect prediciton of the total volume before market open. Because our tests are based on historical data sets, we can use the real total volume of each historical day as the "predictor" of the total volume for that day. This is of course unrealistic in practise, but we just use this approach to verify the impact of being able to make good prediction of total volume to VWAP strategies.

By using the real total volume V as the predictor $\hat{V}$, we have a naive method to make prediction of the proportion ϕ_i for bin b_i. That is to use the volume in the last bin b_{i-1} as the prediction of the volume of bin b_i, then the proportion ϕ_i is $\phi_i = \frac{v_{i-1}}{V}$. Based on this method, we propose the two algorithms as:

- Partial Participation: At the beginning of the day, we use the static curve algorithm to trade up until a time $\hat{T}$. $\hat{T}$ is equivalent to a percentage P of the time it uses the real total volume in the whole trading time T: $\hat{T} = T * (1 - P)$. Then we switch to use the real total volume in the rest of the day and make a prediction based on that.

- Mix Participation: Make two predictions, one uses the static moving average curve algorithm and the other uses the real total volume. Mixing them up by having the part of using the real total volume with a given percentage P, and the part of using static moving average curve with percentage $1 - P$.

Each of the two algorithms has one parameter, P. We adjust them from a small value to big to see the impact. Figure (3.17) and Figure (3.18) highlight how good these two algorithms are and the effect of varying P. Clearly, it suggests the importance of being able to predict the total volume.

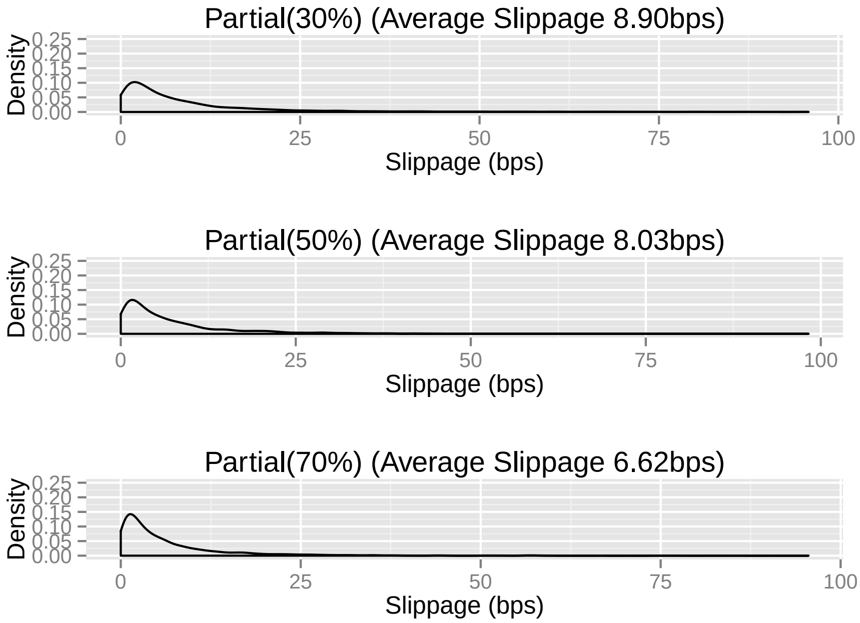


FIGURE 3.17: VWAP Slippage - Partial Participation (NSE)

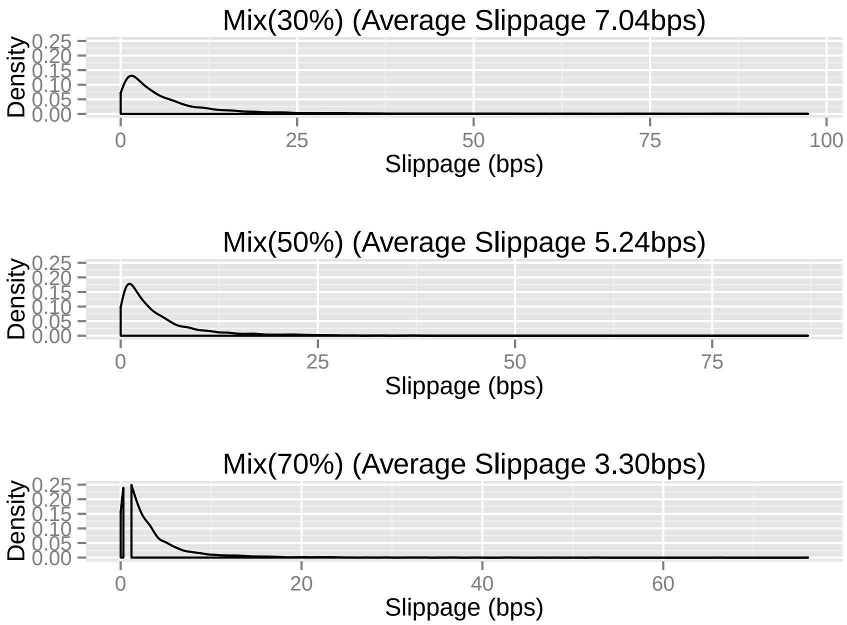


FIGURE 3.18: VWAP Slippage - Mix Participation (NSE)

Figure (3.19) and Figure (3.20) present the goodness for LSE data. They show the same pattern that being able to predict the total volume significantly improves VWAP slippage.

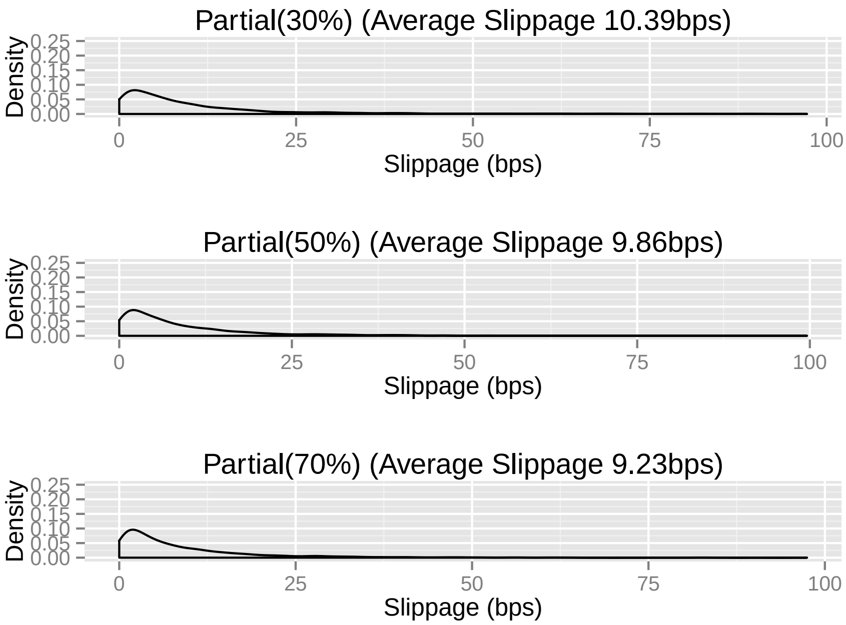


FIGURE 3.19: VWAP Slippage - Partial Participation (LSE)

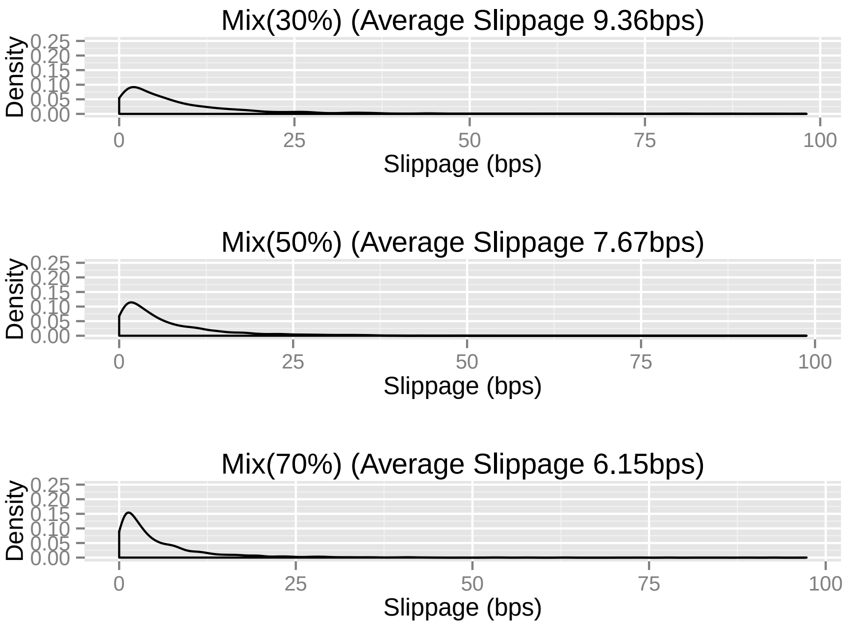


FIGURE 3.20: VWAP Slippage - Mix Participation (LSE)

Before market open, we do not know much about the real total volume at the end of the day. The only thing one can do is to use historical daily volume to predict the number. One can use models that we introduced and proposed in Chapter 2, but still they give just a rough estimation. During the day when we observe more and more trades happening in the market, we are getting more and more information about the intraday situation. One could imagine that this information can be used to predict the remaining total volume in the market. We do a simple test to verify this idea. We divide the trading time into K bins. For each bin b_k, we predict the remaining total volume with the realised accumulated volume and the static moving average curve built on historical data. Because the curve is a prediction of the intrday accumulative volume distribution, for a given bin b_k, we can find from the curve that the predicted percentage of the realised accumulated volume. Doing a simple calculation with both the percentage and the realised accumulated volume, we can get the prediction of the remaining total volume. We expect when it is getting close to the end of the day, this prediction would get close to the real value.

Figure (3.21) gives results for several sample stocks in a testing day. We can see the idea works very well for most of the stocks. This encourages us to develop an algorithm that can leverage this information.

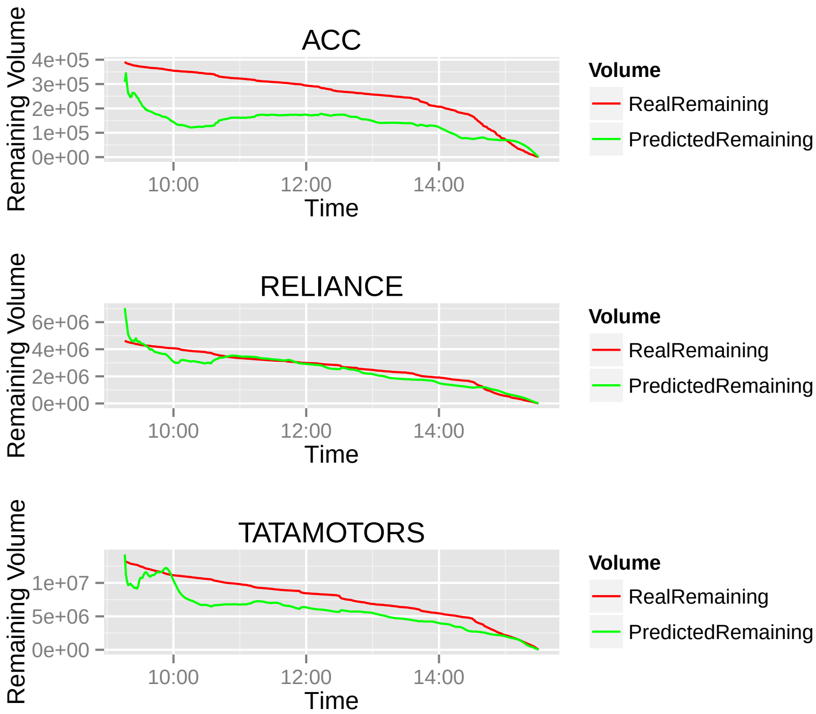


FIGURE 3.21: Remaining Total Volume Prediction (Sample Stocks in NSE)

3.4.3 Hybrid VWAP Strategy

Our main goal is to beat the strategy based on static moving average curve, which is the one normally used in industry. In the literature, there is a line of work that decouples the curve into two parts. One is a common shape shared by stocks, the other is a stock specific dynamic pattern. The stock specific part has a bigger role in the algorithm as this is what the algorithm can learn during the day using observed trades in market data. Białkowski et al. (2008) uses ARMA(1,1) and Self-Extracting Threshold Autoregressive (SETAR) to model this part. As mentioned in Section (3.4.1), we could not replicate the good result claimed by them on the Indian market data we have, which suggests us to find another approach. Based on the empirical study, we have developed a three-phase hybrid VWAP strategy:

1. At the beginning of the day, use the moving average of total volume of previous days as the predicted total volume. We call this the participation phase as it essentially participates in what happens in the market.

2. In the middle of the day, follow the static curve algorithm.

3. In the end of the day, use the realised cumulative volume to predict the final total volume, and apply the same algorithm as phase 1 (participation phase).

For the beginning and end of the day, we follow the method described in Section (3.4.2) to make prediction of the proportion ϕ_i, i.e. $\hat{\phi}_i$, for bin b_i. In these two phases, we have a predicton of the total volume V as $\hat{V}$. We also know the total traded volume, so we have the total remaining volume $\hat{V}_i$ for bin b_i, i.e. $\hat{V}_i = \hat{V} - \sum_{j}^{i-1} v_j$. We just need to predict the volume traded in bin b_i to get $\hat{\phi}_i$, i.e. $\hat{\phi}_i = \frac{\hat{v}_i}{\hat{V}_i}$. We use the volume in the last bin b_{i-1} which is v_{i-1} as the prediction of the volume in bin b_i, in this case we have $\hat{\phi}_i = \frac{v_{i-1}}{\hat{V}_i}$.

This hybrid strategy has two parameters. One represents the percentage of the total order quantity to trade in the first phase. The algorithm trades up until its cumulative traded volume reaches this level, then switches to the second phase. The other one decides when to switch back to the participation phase. For now, we try several groups of these two parameters and use brute force to find the best ones, which are 10% and 30 minutes correspondingly. In the future, we are going to do regression analysis on these two parameters for further tuning.

Figure (3.22) shows how the hybrid algorithm works in a sample day for the stock RELIANCE. First it starts with the participation phase, up until close to 9:30, where the cumulative traded volume reaches 10% of the total order quantity. After that it switches to the static curve algorithm, up until 15:30. In the last half hour, it switches to the participation algorithm using the predicted total volume in that time.

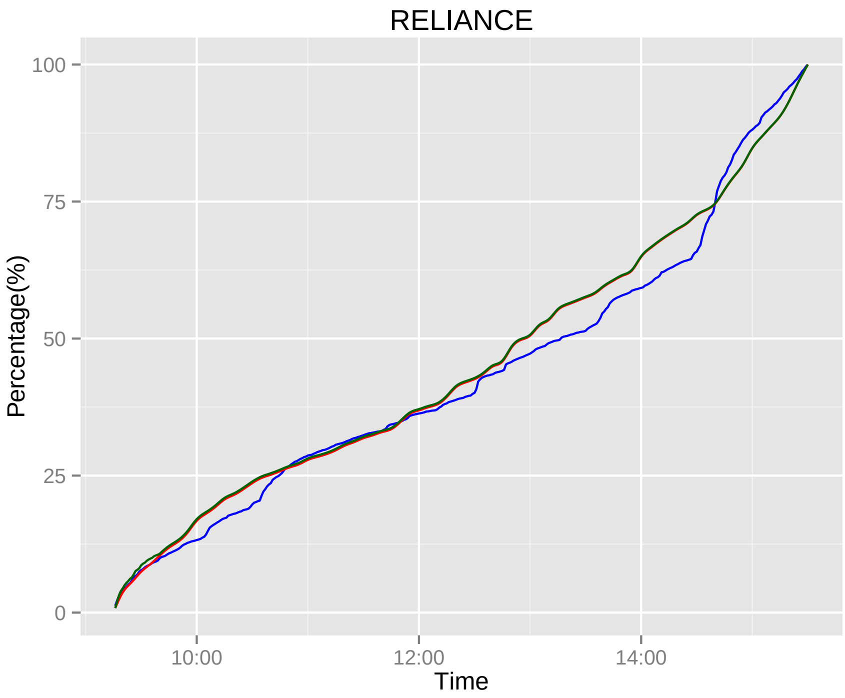


FIGURE 3.22: VWAP - Hybrid

Figure (3.23) shows the intraday trace of the strategy. The red lines are the series of TWAP for each bin b_i. The blue lines are realised VWAP series for each bin using the trades before it. The green lines are the series of VWAP got by the Hybrid strategy.

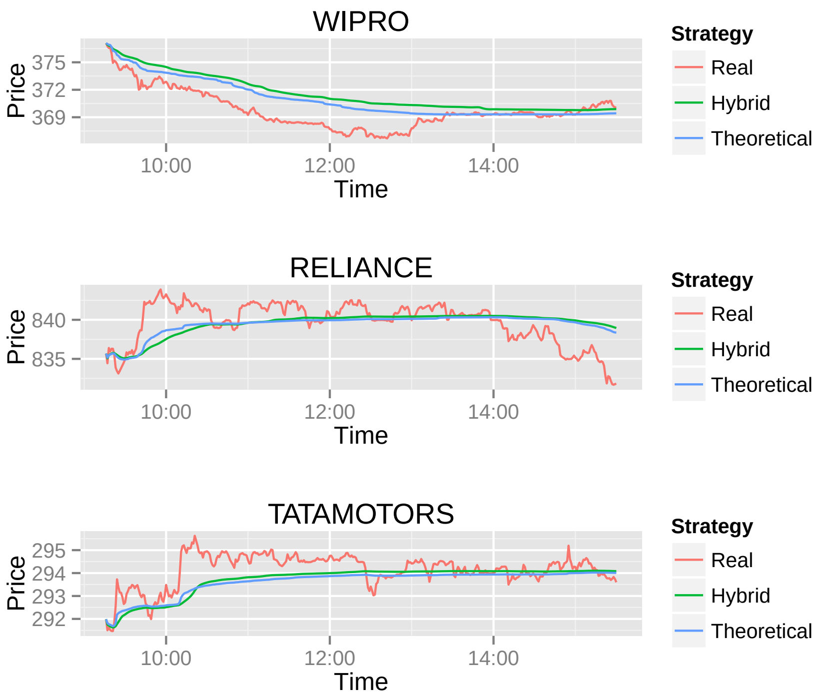


FIGURE 3.23: Hybrid Strategy Intra-Day Trace

Figure (3.24) shows the performance of this strategy. There are four realizations. All of them have the percentage parameter set to 10%, while the switch time is set to market close (no switch), half hour, two and half hours, four and half hours before market close. Surprisingly, the one with no switch performs the best, which indicates the use of total remaining volume prediction is actually harmful.

The best one gives a better slippage 8.65 bps in contrast with the static moving average strategy (9.54 bps). However the ones that do switching at the end do give worse performance than the one that only does participation at the beginning. This indicates that dynamically predicting total volume prediction at the end of the day and adjusting volume allocation is not good. This finding confirms the difficulty of improving VWAP trading with dynamic intraday adjustment.

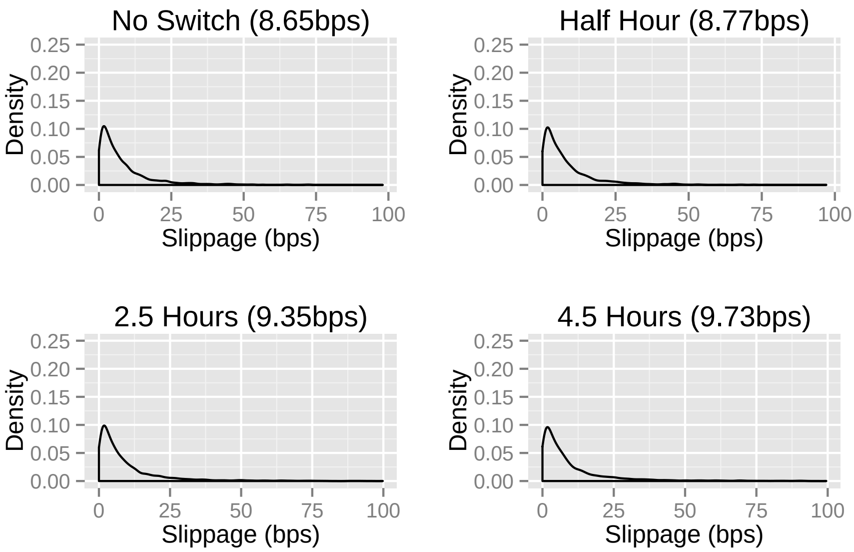


FIGURE 3.24: VWAP Slippage - Hybrid (NSE)

3.4.4 Looking into Limit Order Book Levels

So far, we have gone through strategies that only look at trades data. As described in Section (3.2), there are other types of events in a limit order book market. One could imagine the other types of events may provide information for the volume distribution too. We will look into this idea in this section.

As we know, in an order driven market, participants post both limit orders and market orders to trade. Generally, limit orders are preferred if there is no urgency, because they have lower cost and give the traders better control on price. Although there can be many levels in an order book for both bid and ask orders, the ones that are far away from the top of the order book are not likely to be executed, only the ones close to the top level reveal the real interest of the participants. As shown in Table (3.1), the total size of the orders in the top 5 levels matches the size of traded volume well. Naturally, we are interested in the relationship between the total size of top levels of the order book and the volume distribution. Can we find some clues there to help the prediction of volume innovation? Figure (3.25) outlines the intra-day dynamics of the total size of the top levels of the order book of the stock RELIANCE listed in NSE (We have observed the similar pattern in the whole data set, here we pick up 4 sample days to illustrate the pattern). Unfortunately, it is hard to see any clue indicating the volume distribution.

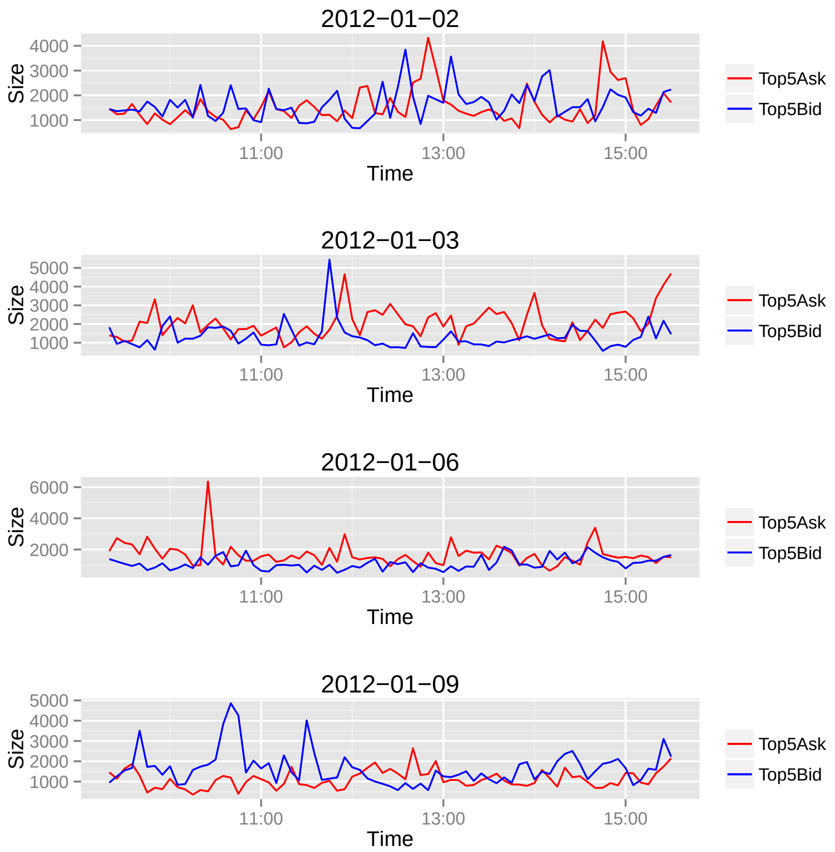


FIGURE 3.25: Top 5 Bid and Ask Sizes (NSE, Stock RELIANCE)

Figure (3.26) shows the difference between the size of bid and ask orders in the top 5 levels of the stock RELIANCE listed in NSE (Again we have observed the similar pattern in the whole data set, here we just pick up 4 days to illustrate). Still, it is unclear how this can help the prediction of volume distribution. Most of the time, the difference moves up and down around 0, showing the fight between the buy and sell side. We could have a hypothesis that the size or volatility of the difference may indicate more trading pressure or activities, but we can not see any evidence from the figures. What is more difficult is to associate this information to the intra-day relative volume distribution, it is hard to see any signal for that.

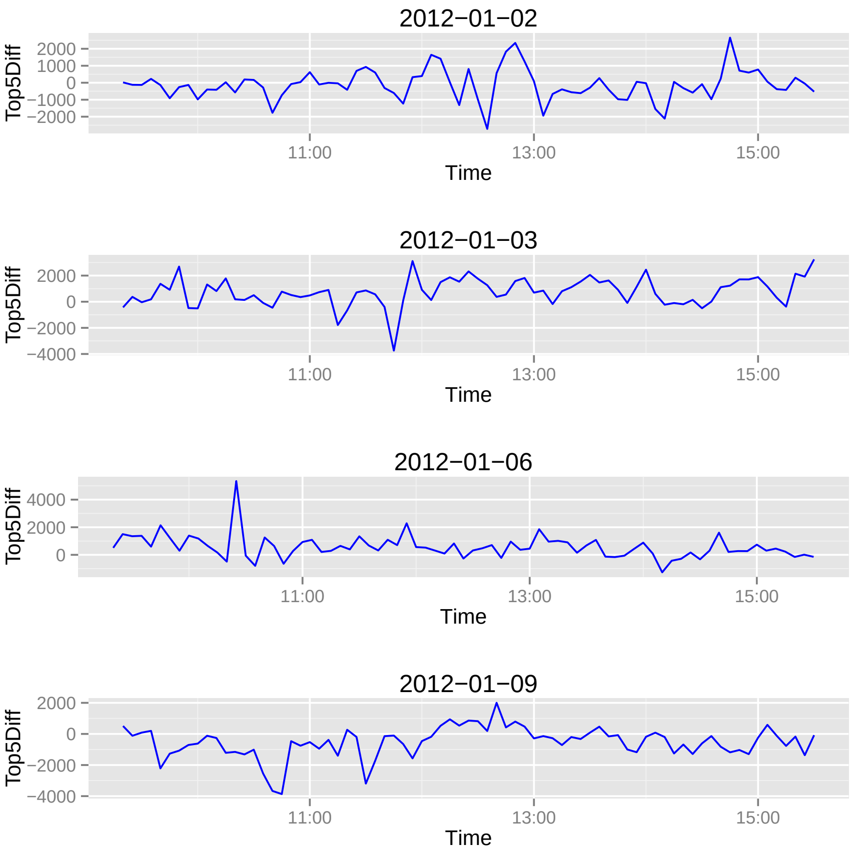


FIGURE 3.26: Top 5 Bid and Ask Size Difference (NSE, Stock RELIANCE)

Figure (3.27) presents the dynamics of the spread of the stock RELIANCE listed in NSE (We have observed the similar pattern in the whole data set, here we just pick up 4 days to illustrate). Spread is highly correlated with liquidity, small spread comes with high liquidity. We would expect to see similar intra-day seasonality patterns in the spread dynamics. Figure (3.27) fails this hypothesis. The figures look more like noise than meaningful indicators.

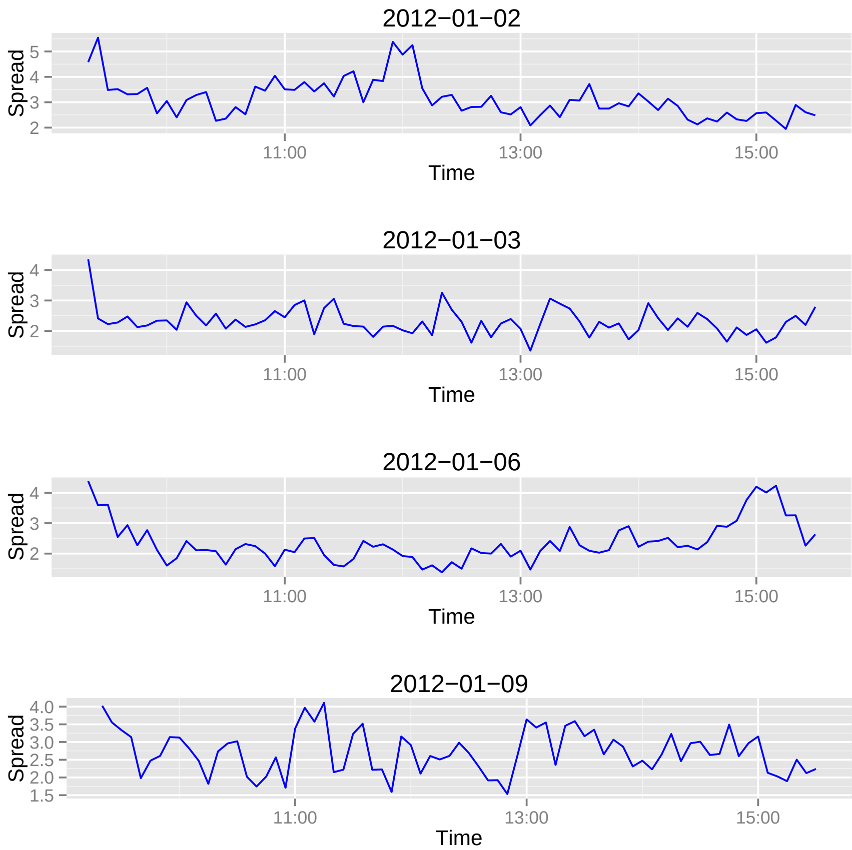


FIGURE 3.27: Limit Order Book Spread (NSE, Stock RELIANCE)

The last thing we would like to check is the ratio between trades and all market events. This suggests how easy it is to see a trade in the market. Because the market is order driven and limit orders are normally preferred to market orders, most of the events in the market are about limit orders. As modifying an existing order is much cheaper than creating a new order, most of the events are order modifications. Thus, if we knew that for a specific period, the ratio between trades and all market events suggest more volume, we would be able to capture that liquidity innovation. Figure (3.28) demonstrates the dynamic of this ratio of the stock RELIANCE listed in NSE (We have observed the similar pattern in the whole data set, here we just pick up 4 days to illustrate). Again, it appears our hypothesis does not hold.

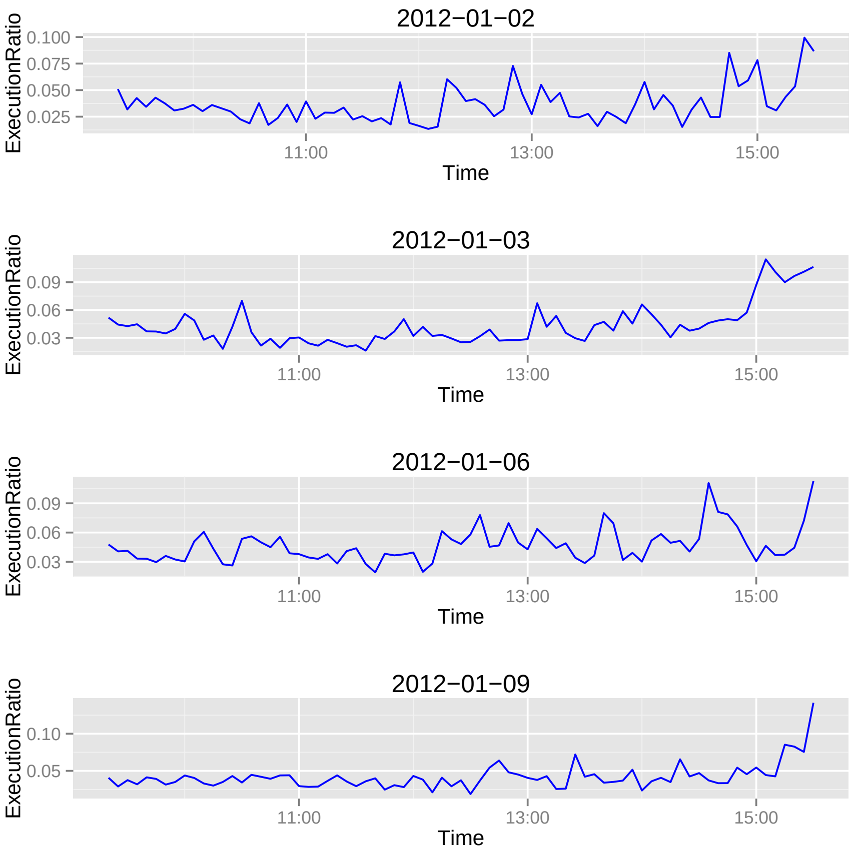


FIGURE 3.28: Execution Ratio (NSE, Stock RELIANCE)

Now we have a good understanding that this task is actually quite difficult. The fact that we cannot make use of the rich information in limit order books for volume distribution prediction is frustrating. But recall the motivation of the task, it is to help the trader to reduce market impact. If we think about it a bit differently, the reason we like the idea of volume distribution prediction is that it removes the worries for price process modeling, but we do not have to do that if we can find other alternatives to help reducing market impact. Following this idea, we will develop a new strategy in the next section which can actually contribute to this motivation.

3.4.5 Buy/Sell Strategy

Recall that in Section (3.3), a trader is motivated to hide the private trading incentive from the public, in the interest of getting better execution of the order. VWAP tracking is one of the common approaches for doing that, this quantifies the market impact of an order by the difference between the execution price and the market VWAP in the same period. In a typical senario, a trader issues a VWAP tracking order and assigns a target period to it, the executor splits the parent order into small chunks and spreads them across the period, and makes sure that the total order quantity has been traded by the end of the period. The parent order and the children orders have the same side, i.e. if the parent order is a buy order, all children orders are buy orders too. This is to simplify the execution, have lower transaction costs and make it easier to calculate the executed VWAP. Usually the trading systems also only support this type of operation, to avoid the complexity and cost of managing two sides for the order. In this way, the traded volume during the period is monotonically increasing.

Back to the motivation, one could question that what is the actual impact of executing the VWAP order in both sides. By doing so, we will not be restricted by the total order quantity, thus we can leverage the finding in the participation strategy to lock the volume distribution. This leads us to a way to develop a new strategy. Recall that in Section(3.4.3), the biggest obstruction for us to keep doing the participation phase is that we are not able to predict the exact quantity of the total daily volume. Assume we have the predicted total volume as $\hat{V}$, while the real total volume is V. We can have two cases when $\hat{V} \neq V$:

1. If $\hat{V} < V$ we will end up trading too fast to run out of order quantity at time $T' < T$, the traded VWAP would be $P^{T'}_{VWAP}$, while the real VWAP is P_{VWAP}. Depending on the distance of T' and T, the slippage can be large. We can see this from Figure (3.23).
2. If $\hat{V} > V$, we will have untraded order quantity at the market close time, which will force us to trade all of them at the market close price. Figure (3.23) shows the market close price can be very different than the final VWAP.

In these two cases, the second one leaves us very little space for adjustment, whenever we realize the predicted $\hat{V}$ is less than the real V, we will have to predict the price process in the rest of the day to catch up. Considering the difficulty of predicting price process, we would like to avoid falling into this case. Thus, we need to find a way to transfer the last case into the first one by reducing the quantity of the predicted total volume, which we will discuss later. Now, let's assume we always fall into the first case. At the time we run out of order quantity, if we relax the restriction of being at only one side, we can then go along the opposite side to keep trading, and then back and forth to make sure we end up at the required total order quantity.

This strategy will make the total traded volume nondeterministic before the market close. Depending on the prediction error of the total daily volume between $\hat{V}$ and V, the realised total traded quantity is $\frac{V}{\hat{V}} \times Q$. To judge the feasibility of this strategy, we would need to have a view on the the prediction error. As described before, we have two approaches for predicting the total daily volume, both of them takes the last L days historical daily total volume as input. One takes the moving average of the L days as the prediction for the next day. The other fits the L days into the ARMA-DECOMP model proposed in Chapter 2 and then uses the model to make the prediction.

We follow Section (2.2.4) to define the prediction error $e_{i,t}$ for day t of instrument s_i as the log of the percentage between the actual volume $v_{i,t}$ and the predicted volume $\hat{v}_{i,t}$.

$$e_{i,t} = ln\frac{v_{i,t}}{\hat{v}_{i,t}} \tag{3.8}$$

And for instrument s_i the prediction error across all M days is the standard deviation of the single day error:

$$\tilde{e}_i = \sqrt{\frac{\sum_{t}^{M}(e_{i,t} - \bar{e}_i)^2}{M}} \tag{3.9}$$

Table (3.4) presents the prediction errors for these two approaches for the NIFTY stocks. As we are focused on the case where the predicted $\hat{V}$ is less than the real V, we filter out the errors where $\hat{V} > V$. On average, the moving average approach gives 68.21% error while the ARMA-DECOMP approach gives a better result as 58.51%. That means we need to exceed a bit more than half of the required order quantity in that case. This is not an exciting result, but not bad either. Indeed, trading more than half of the desired volume may increase market impact, especially when the desired one is large. But recall the definition of VWAP, the larger the volume a trader put into the market, the more the greater the participation in market activity, thus the more VWAP represents itself. In that case, VWAP may not be an ideal benchmark for the market impact. Taking that into account, we focus on the cases where the desired order quantity is not a significant part of the whole volume in market. With this consideration, exceeding half of the quantity would be an acceptable trade-off.

	Symbol	Moving Average Error (%)	ARMA-DECOMP Error (%)
1	ACC	78.30	62.92
2	AMBUJACEM	66.97	62.13
3	ASIANPAINT	70.46	61.52
4	AXISBANK	47.84	36.96
5	BAJAJ-AUTO	67.90	67.90
6	BANKBARODA	83.56	70.25
7	BHEL	66.57	45.69
8	BPCL	80.23	68.39
9	BHARTIARTL	76.13	50.44
10	CAIRN	109.51	124.31
11	CIPLA	77.03	67.88
12	COALINDIA	79.14	68.66
13	DLF	49.65	33.24
14	DRREDDY	67.03	56.75
15	GAIL	71.85	64.46
16	GRASIM	76.96	75.78
17	HCLTECH	83.09	81.97
18	HDFCBANK	53.87	42.96
19	HEROMOTOCO	63.96	50.67
20	HINDALCO	51.16	41.55
21	HINDUNILVR	65.52	62.63
22	HDFC	105.53	109.36
23	ITC	65.46	58.98
24	ICICIBANK	47.79	42.49

25	IDFC	48.92	40.46
26	INFY	65.68	56.13
27	JPASSOCIAT	34.12	31.46
28	JINDALSTEL	73.73	53.49
29	KOTAKBANK	90.70	71.53
30	LT	51.06	35.12
31	LUPIN	72.50	64.24
32	M&M	74.02	74.02
33	MARUTI	75.63	64.41
34	NTPC	67.17	62.91
35	ONGC	64.33	52.83
36	POWERGRID	83.15	59.10
37	PNB	69.91	55.38
38	RANBAXY	67.44	52.15
39	RELIANCE	48.28	39.49
40	RELINFRA	38.91	26.92
41	SESAGOA	61.16	38.45
42	SIEMENS	87.99	68.84
43	SBIN	41.80	30.87
44	SUNPHARMA	71.43	62.25
45	TCS	79.34	68.28
46	TATAMOTORS	53.49	49.89
47	TATAPOWER	72.07	56.52
48	TATASTEEL	40.86	31.23
49	ULTRACEMCO	110.48	110.66
50	WIPRO	61.00	60.90
51	Average	68.21	58.51

Table 3.4: Prediction Error for Total Daily Volume (NSE)

Table (3.5) shows the prediction errors for the two approaches for the LSE stocks. Again ARMA-DECOMP gives a constant better result.

	Symbol	Moving Average Error (%)	ARMA-DECOMP Error (%)
1	SHP.L	65.47	56.50
2	XTA.L	41.40	31.60
3	EXPN.L	57.75	46.11
4	HSBA.L	47.16	39.00
5	IMT.L	54.03	45.74
6	LSE.L	83.22	72.50
7	TSCO.L	46.65	37.83
8	AAL.L	48.54	37.46
9	STAN.L	47.73	39.11

10	RBS.L	67.55	48.72
11	TATE.L	42.46	40.77
12	LLOY.L	65.38	49.71
13	VOD.L	44.50	40.70
14	LGEN.L	42.72	46.22
15	NXT.L	56.24	50.97
16	UU.L	66.08	46.25
17	GSK.L	43.46	42.12
18	BAES.L	45.24	40.00
19	BATS.L	45.36	39.96
20	RIO.L	51.75	34.42
21	Average	53.13	44.29

TABLE 3.5: Prediction Error for Total Daily Volume (LSE)

Now, let's think about the case where the predicted $\hat{V}$ is greater than the actual real volume V. In this case, we need to adjust the trading spread before market close to catch up the desired order quantity. There are two issues:

1. At time $t < T$, we need to check if the traded order percentage is behind the real volume distribution curve, while the curve is still unknown.
2. When we decide to adjust the trading speed, we then need to determine how much to scale it.

For issue 1, we have no way to compare the traded order percentage directly with the real curve - we would have followed the curve if we had a measurement for it. But recall that we can have a static curve made in advance based on a moving average of the historical days, which is a good candidate to use as an alternative of the real curve, because of the seasonality it reflects. So, having t as the start time of the i^{th} bin b_i, we have information for:

1. The desired order quantity to be traded, Q.
2. The total traded order quantity up until t, q_{i-1}.
3. The predicted total volume $\hat{V}$.
4. The observed volume traded in the previous bin b_{i-1}, v_{i-1}.
5. The observed total volume traded up until t, V_{i-1}^t.
6. The reference volume distribution function obtained as a moving average of the past L days, which is a function of time t, $f(t)$. The accumulated volume proportion can thus be defined as a function $F(t) = \int_{T_{start}}^{t} f(x)\mathrm{d}x$.

Our buy/sell strategy starts with doing exactly as the first phase of the hybrid strategy, that said, the order quantity to be traded in bin b_i would be $\frac{v_{i-1}}{\hat{V}} \times Q$. This will be fine if $\frac{q_{i-1}}{Q} \geq \frac{V^t_{i-1}}{V}$, because we can trade in both sides to make sure in the end we have bought or sold exactly as the desired quantity Q. As explained above, the tough thing is to determine when do we have $\frac{q_{i-1}}{Q} < \frac{V^t_{i-1}}{V}$, which is when we need to adjust the trading speed. With the reference accumulated volume proportion function $F(t)$ and the observed total volume V^t_{i-1}, we can have an approximation of the real daily total volume as $\frac{V^t_{i-1}}{F(t)}$. So, the strategy is to compare q_{i-1} and $\frac{V^t_{i-1}}{F(t)}$, if $q_{i-1} < \frac{V^t_{i-1}}{F(t)}$, adjust the speed.

The unadjusted trading speed for bin b_i is $\frac{v_{i-1}}{\hat{V}} \times Q$, here we can reduce the predicted total volume $\hat{V}$ to boost the speed. The aim is to catch up the realised accumulated volume curve, naturally we would use $\frac{v_{i-1}}{\hat{V}}$ as the scale ratio, i.e. set:

$$\hat{V} = \frac{\sum_{j=1}^{i-1} v_j}{Q \times F(t)} \times \hat{V}$$

However $\sum_{j=1}^{i-1} v_j$ represents what has been traded, what we care is really what has not been traded, so the scaling should be:

$$\hat{V} = \frac{Q - \sum_{j=1}^{i-1} v_j}{Q \times (1 - F(t))} \times \hat{V}$$

Taking the randomness that comes with the realised total volume for each bin b_i into account, we need to filter the adjustment to make it smooth, so we only shrink $\hat{V}$, never enlarge it.

Let Q_{buy} be the order quantity traded on buy side, and Q_{sell} be the order quantity traded on sell side, the following equation must hold:

$$Q = |Q_{buy} - Q_{sell}|$$

In production, one can decide when to switch to the other side by looking at the order book and the price, to get the best time for switching. This is not the focus of this chapter, so we will not discuss it. As long as the market is liquid, we will be able to safely operate to guarantee that we have just traded Q for the desired side.

One could have a concern regards to the transaction costs. The buy/sell strategy trades in two directions, which results that it trades more than the original total order quantity. The extra volume it generates is:

$$Q_{extra} = Q_{buy} + Q_{sell} - Q$$

This means that we may need to pay transaction costs and spreads the extra volume Q_{extra}. We can minimize such costs by using limit orders instead of market orders. Because posting limit orders is actually providing liquidity to the market, most exchanges give compensation for such orders rather than charging for them. Limit orders can also avoid the cost for spread, because it does not require to immediately cross the order book. Limit orders do not guarantee timely execution as market orders though, so we may have to use market orders to trade when we get close to the market close time. In that case, as long as the market is liquid, and the original total order quantity is large (which is always the case for a VWAP order), the transaction costs of the last orders will be trivial compared to the total cost of the entire execution. Taking these points into account, I do not focus on the transaction costs in this chapter.

We show the performance of the buy/sell strategy on NSE data set in Figure (3.29). On average, it gives 3.34 bps slippage, which is a huge improvement compared to the static moving average approach.

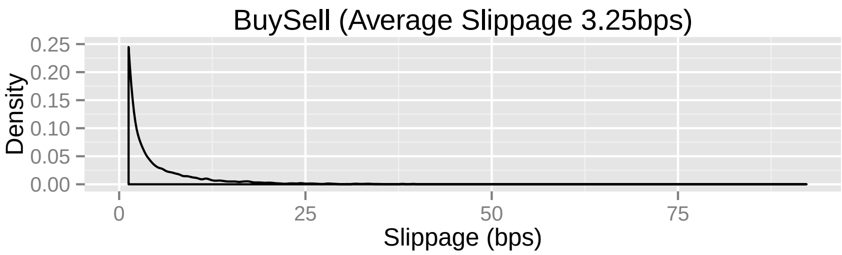


FIGURE 3.29: VWAP Slippage - Buy/Sell (NSE)

Figure (3.30) presents the performance on the LSE data set. On average, it gives 8.84 bps slippage. Still, it is a big improvement over the 13.59 bps slippage that the moving average static curve strategy gives.

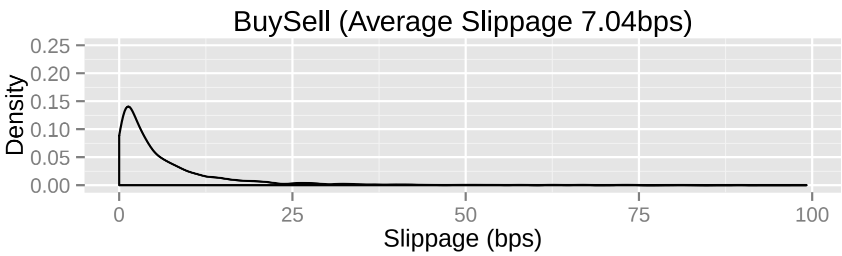


FIGURE 3.30: VWAP Slippage - Buy/Sell (LSE)

Figure (3.31) shows the distribution of the extra volume traded by the strategy on the NSE data, measured as a percentage. The average exceeding percentage is 44.09%, which is also better than the 58.51% we got for the case where the predicted $\hat{V}$ is less than the real V. This result indicates that the buy/sell strategy works well for catching up when $\hat{V} > V$.

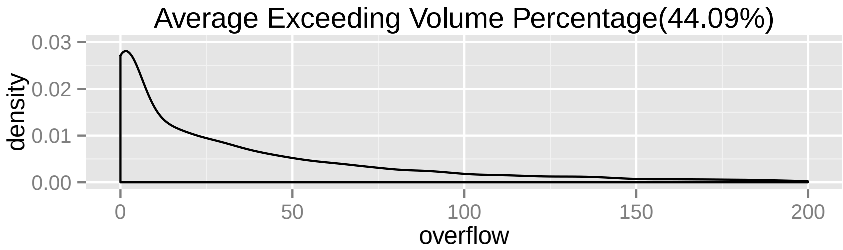


FIGURE 3.31: Exceeding Volume Percentage (NSE)

Figure (3.32) graphs the results for LSE data. The average exceeding percentage is worse than the one we got from the NSE data, which could largely because there are less data points in the LSE data.

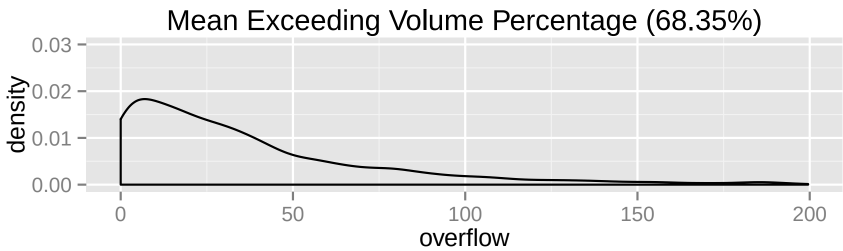


FIGURE 3.32: Exceeding Volume Percentage (LSE)

The results support that this buy/sell strategy generates much less slippage for VWAP trading with an acceptable exceeding volume.

3.5 Conclusion

In this chapter, I have addressed the research question about modeling the intraday volume distribution. Having a good prediction model for the intraday volume distribution helps trading strategies to better minimize their transaction costs and market impact. In particular, there is a class of trading strategies called VWAP strategies that aim at distributing large orders along with the intraday volume distribution to reduce the market impact of such orders. The VWAP strategies are important for institutional market participants because they often have needs to execute large orders in short time. I have done thorough analysis for VWAP strategies using various intraday volume distribution models on real market data from two exchanges (NSE and LSE). I have developed a new bidirectional VWAP trading strategy that gives significant improvement over traditional VWAP trading strategies.

I started by modeling the intraday volume distribution with simple models like theoretical or linear model. I then tried the traditional model based on static moving average curve built on historical data. I also tried to replicate models suggested by previous research work, such as principal component, ARMA and SETAR. I have shown that the traditional strategy based on static moving average curve performs best and it is very difficult to outperform it.

I have given analysis of the VWAP trading process with synthetic data to illustrate the difficulty of designing dynamic VWAP trading strategies by just using relative volume distribution. I have then shown the impact of being able to predict total volume on VWAP trading strategies. Based on the findings I have developed a hybrid strategy that gives a reasonable improvement over the traditional strategies.

Inspired by the analysis of the above VWAP strategies and the empirical findings in the market data, I have developed a bidirectional trading strategy called buy/sell strategy. This strategy significantly outperforms all the other strategies. The buy/sell strategy trades in both buy and sell side, and dynamically adjusts the traded volume in both sides to ensure the net volume matches the target total order quantity.

The buy/sell strategy however has to trade more than the original order quantity in total because of the bidirectional trading. I have shown that the exceeding volume it generates is in a reasonable range. Future work will be on optimizing the timing of changing direction to minimize the exceeding volume.

Chapter 4

Modeling Liquidity with The Hawkes Processes

4.1 Introduction

In this chapter, I try to model liquidity on the microstructure level of the market data. At this level we can observe that the timestamps of the individual trade events are irregularly spaced, and each event can have different traded price and size. I use the Hawkes processes to model such events. By using a non-parametric kernel estimation method, I verify that there is a self-exciting feature in the market data from National Stock Exchange (NSE).

An increasing number of financial transactions are executed by automatic and algorithmic trading strategies in order-driven markets. In such markets, a centralised limit order book is available to market participants to post buy or sell limit orders for a given price and quantity. If the limit orders do not cross the best offers in the limit order book, they will rest and wait to be executed. Executions happen in two cases. Firstly, when market orders hit the best available offers in the limit order book, the orders standing on the offers are executed. Secondly, if someone posts aggressive limit orders which cross the best available offers, these orders and the ones standing on best offers are executed directly. With the growth of electronic trading engines and algorithmic participants, markets now generate events at a very high frequency with several events occurring per millisecond. These events are irregularly time-distributed and rich in properties like price, quantity, side as well as the order book status at the occurring time. The availability of such data generates a lot of interest in statistical modeling.

With the rapid development of the algorithmic trading business, it is more and more important for algorithmic trading engines to optimize their intraday trading strategies. Businesses like market making or statistical arbitrage benefit from sophisticated algorithms that can seek profit opportunities from the market, but suffer from transaction cost and market impact. Thus it is very important for such businesses to have optimal execution strategies. For any trading strategy to make sense, it is critical that the trading system can execute orders with best timing and best price. As stressed by Lehalle (2013), in the sense of optimal liquidation, large orders need to be split through time, which requires proper modeling of market micro-structure like price formation process and volatility statistics. When there are multiple venues available for trade, optimal allocation of orders through venues can help minimise risk exposure and reduce transaction costs as well. Significant academic work has been dedicated to execution optimization. For example, Junca (2011) models the optimal execution strategy in the presence of price impact and fixed transaction cost as an impulse optimal control problem. Alfonsi et al. (2010) study the exponential recovery of the number of limit orders and the bid-ask spread in the context of block market orders placement optimization.

Efforts were made to study the irregular distribution features of financial time series, one well known model is the Autoregressive Conditional Duration (ACD) model (Engle and Russell, 1998). Time series models typically measure events at uniform time intervals. However, order book events are naturally a series of events rather than random numbers with uniform timestamps, which do not necessarily fit time series models well. Cont et al. (2010) propose a continuous-time stochastic model that uses Laplace transform methods to compute probabilities of order book events conditional on the state of the order book. Cont and Larrard (2012b) use a Markovian queueing system to model the arrivals of order book events. Cont and Larrard (2012a) model the intraday dynamics of the limit order book with a Markovian jump-diffusion process, and derives functional central limit theorem for the joint dynamics of the bid and ask queues.

Order book events tend to be self-exciting because traders would trade more when they have some critical news and this trading activity itself can be a signal to others. Thus trades would tend to happen in clusters. Recent contributions have emphasised that a well-studied self-exciting Point Process, known as Hawkes processes, introduced in Hawkes (1971a) and Hawkes (1971b).

Hawkes processes were first introduced in seismicity analysis in Ogata (1988) as Epidemic-Type After Shock (ETAS) models. Recently, they have also started to gather interest from financial researchers. Bowsher (2007) presents a comprehensive econometric framework. Cartea et al. (2011) use a multi-factor self-exciting process which allows for feedback effects in market buy and sell orders and the shape of the limit order book. Bacry et al. (2012b) explain the Epps effect in a microstructure toy model. Bacry et al. (2011) model the variations of asset prices at the tick-by-tick level using Marked Point Processes.

Hawkes processes have been employed as a modeling tool in algorithmic trading strategies. Toke and Fabrizio (2012) model trades-through, i.e. transactions that reach at least the second level of limit orders in an order book, with a bivariate Hawkes process. Hewlett (2006) fits a bivariate Hawkes process to arrival data for buy and sell trades in FX markets in order to optimize liquidation strategies. Similarly, Carlsson et al. (2007) and Shek (2010) study bivariate Hawkes processes to model bid/offer imbalance in limit order books. The intensity obtained from the model is used as a signal to update the trading strategy. Chavez-Demoulin and McGill (2012) apply the Epidemic-Type After Shock (EATS) model Ogata (1988) to predict intraday Value-at-Risk. Bayraktar and Ludkovski (2014) present a point process model whose intensity depends on the liquidation price for maximising the expected revenue from liquidating the entire position held.

There are techniques developed for non-parametric kernel estimation for Hawkes process. Lewis and Mohler (2011) and Marsan and Lengliné (2008) propose methods based on Expected Maximization (EM) procedure of a penalised likelihood function. Bacry et al. (2012a) present a numerical spectral method that estimates the kernel from the covariance matrix of the Hawkes process but only fits for symmetric kernel. Bacry and Muzy (2014) and Bacry and Muzy (2013) show that the correlation matrix of the Hawkes process is related to the kernel by a system of Wiener-Hopf integral equations, and thus can be used to solve the kernel by the Nyström method introduced by Nyström (1925).

The structure of this chapter is as follows. In Section (4.2) I present the modeling of the Hawkes process including both parameters estimation and non-parametric kernel estimation, goodness of fit and the way to calculate conditional durations. Section (4.3) presents market data from the National Stock Exchange (NSE) in India and discuss the need for data pre-processing, then shows how to fit market data with Hawkes processes and uses the resulting model to improve strategies tracking the Volume Weighted Average Price (VWAP). Section (4.4) concludes the chapter and indicates further directions.

4.2 Modeling

4.2.1 Hawkes Processes

Let us first recall standard definitions and properties of Hawkes processes. These processes were introduced in Hawkes (1971b) as a special case of linear self-exciting processes with an exponentially decaying kernel.

Let $\{t_i\}_{i\in\{0,1,2,\dots T\}}$ be a sequence of non-negative random variables such that $\forall i \in \mathbb{N}, t_i < t_i + 1$. We denote the number of events, $\mathrm{N}_t = \sum_i 1_{t_i<T}$, the associated counting process. We define $\mu : \mathbb{R} \mapsto \mathbb{R}_+$ as the deterministic base intensity of arrivals, and ϕ as the propagator of past events u onto the intensity of arrivals in the future. The arrival intensity λ_t of the process is defined by (4.1).

$$\lambda(t) = \mu(t) + \int_{-\infty}^{t} \phi(t-s)\,\mathrm{dN}_s \tag{4.1}$$

Hawkes (1971b) proposes an exponential kernel $\phi(t) = \alpha\, e^{-\beta t}$, so the intensity of the model becomes:

$$\begin{aligned} \lambda(t) &= \mu(t) + \int_{0}^{t} \alpha\, e^{-\beta(t-s)}\,\mathrm{dN}_s \\ &= \mu(t) + \sum_{t_i<t} \alpha\, e^{-\beta(t-t_i)} \end{aligned} \tag{4.2}$$

Parameters α and β express the influence (scale and decay) of the past events t_i on the intensity of the process.

Assuming stationarity gives $E[\lambda(t)] = \Lambda$ where Λ is a constant. We have:

$$\begin{aligned}\Lambda = E[\lambda(t)] &= E[\mu(t) + \int_{-\infty}^{t} h(t-s)\,\mathrm{dN}_s] \\ &= \mu(t) + E[\int_{-\infty}^{t} h(t-s)\lambda(s)\,\mathrm{d}s] \\ &= \mu(t) + \int_{-\infty}^{t} h(t-s)\,\Lambda\,\mathrm{d}s \\ &= \mu(t) + \Lambda\int_{0}^{\infty} h(v)\,\mathrm{d}v\end{aligned} \tag{4.3}$$

Which gives:

$$\Lambda = \frac{\mu(t)}{1-\int_0^\infty h(v)\,\mathrm{d}v} \tag{4.4}$$

Equation (4.4) immediately gives for the univariate Hawkes process the unconditional expected value of the intensity process:

$$E[\lambda(t)] = \Lambda = \frac{E[\mu(t)]}{1-\alpha/\beta} \tag{4.5}$$

The integral of the kernel function $\phi(t)$ is called the branching ratio n, which models the endogeneity (or reflexivity) present in the process, is:

$$n := \int_0^\infty \phi(t)\mathrm{d}t > 0 \tag{4.6}$$

Because the arrival time of the events in the Hawkes process is a series of discrete timestamps $t_i, i \in \mathbb{N}$, we can rewrite the conditional intensity defined in (4.1) as:

$$\lambda(t) = \mu(t) + \sum_{t_i<t}\phi(t-t_i) \tag{4.7}$$

As explained in Filimonov and Sornette (2013), this linear structure of the conditional intensity allows one to consider the Hawkes process as a Poisson process $t_i^c, i \in \mathbb{N}$ with rate $\mu(t)$, where t_i^c is the center of the i^{th} cluster. In this context, each event t_i is either an immigrant or a descendant. The branching ratio n is the most crucial parameter of the Hawkes process in the sense that it quantifies how many events are triggered by a parent event (when $n = 1$ every events are immigrants). Though the focus is not on this aspect of the Hawkes process here, I will later follow the work in Filimonov and Sornette (2013) for checking the calibration issues that can hurt the application of the Hawkes process.

One can simulate this self-excited intensity process by the usual thinning method Ogata (1988). Figure (4.1) gives one sample realisation of the simulated process. The x-axis and y-axis are time and intensity respectively. The blue triangles represent the time series of the simulated events. The red curve is the conditinal intensity $\lambda(t)$. We can see when there are clusters of events (blue triangles) the intensity goes high (the red curve) and then gets decay when the rate of events decreases.

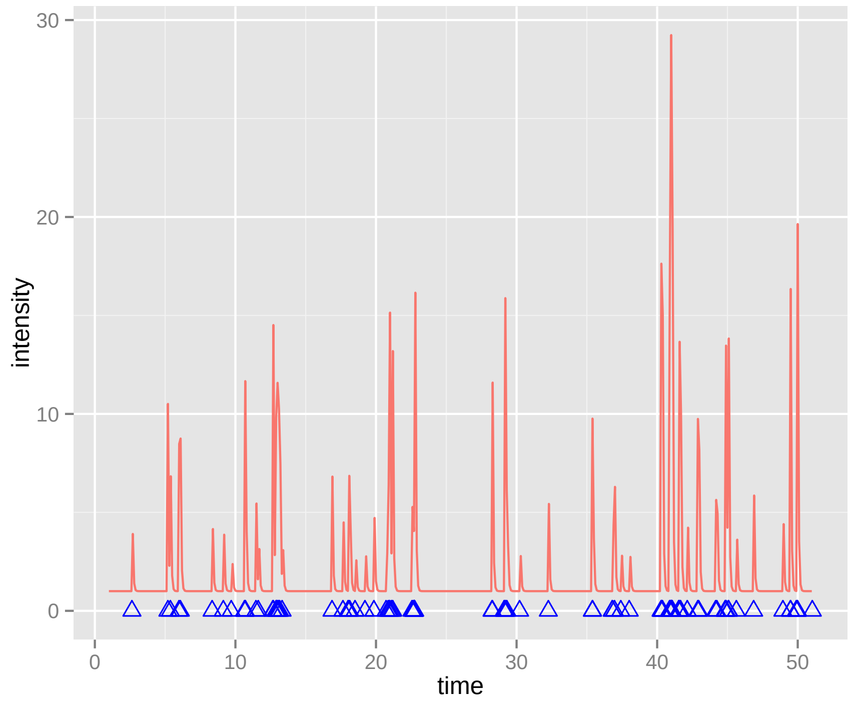


FIGURE 4.1: Simulation of a univariate Hawkes process with $\mu = 1, \alpha = 10, \beta = 20$

The log-likelihood of a simple point process N_s can be computed as Ogata (1981):

$$ln\ \mathcal{L}((N_t)_{t\in[0,T]}) = \int_0^T (1 - \lambda(s))\mathrm{d}s + \int_0^T ln\ \lambda(s)\mathrm{dN}_s \tag{4.8}$$

which in the case of a Hawkes model can be explicitly computed as:

$$ln\ \mathcal{L}(\{t_i\}_{i=1,\dots,N}) = t_N - \int_0^{t_N} \mu\,\mathrm{d}s + \sum_{i=1}^{N} ln\ [\mu + \sum_{k=1}^{i-1} \alpha e^{-\beta(t_i - t_k)}] \tag{4.9}$$

Ogata (1978) shows that for a stationary univariate Hawkes process, the maximum-likelihood estimator $\hat{\theta}^T = (\hat{\mu}, \hat{\alpha}, \hat{\beta})$ is:

- consistent, i.e. converges in probability to the true values $\theta = (\mu, \alpha, \beta)$ as $T \to \infty$:

$$\forall \epsilon > 0, \lim_{T\to\infty} \mathrm{P}[|\hat{\theta}^T - \theta| > \epsilon] = 0 \tag{4.10}$$

- asymptotically normal, i.e.

$$\sqrt{T}(\hat{\theta}^T - \theta) \to \mathcal{N}(0, \mathrm{I}^{-1}(\theta)) \tag{4.11}$$

where $\mathrm{I}^{-1}(\theta) = (E[\frac{1}{\lambda}\frac{\partial\lambda}{\partial\theta_i}\frac{\partial\lambda}{\partial\theta_j}])_{i,j}$

- asymptotically efficient, i.e. asymptotically reaches the lower bound of the variance.

Therefore the more data points we can use as the input of the estimator, the more likely that we can get close to the true values of the model parameters. The fact that the amount of data in trading space is always big makes it a good application of doing such estimation.

The likelihood function for the exponential kernel specification can be computed in O(N) steps, whereas for the more general kernel $h(t)$, O(N^2) steps will be required as per Hewlett (2006). In this chapter, I focus on the exponential kernel so we just use the O(N) method.

I follow Toke and Fabrizio (2012) to calculate the log-likelihood estimator in a recursive way. For the univariate Hawkes process, we have:

$$\begin{aligned} R(l) &= \sum_{k=1}^{l} \alpha e^{-\beta(t_i - t_k)} \\ &= e^{-\beta(t_l - t_{l-1})}(1 + R(l-1)) \end{aligned} \tag{4.12}$$

Then we can write the log-likelihood estimator as:

$$\begin{aligned} ln\ \mathcal{L}(\{t_i\}_{i=1,\ldots,N}) = t_N - \int_0^{t_N} \mu \,\mathrm{d}s - \sum_{i=1}^{N} [\frac{\alpha}{\beta}(1 - e^{-\beta(t_N - t_i)})] + ln\ \int_0^{t_N} \mu \,\mathrm{d}s \\ + \sum_{i=2}^{N} ln\ [\int_0^{t_N} \mu \,\mathrm{d}s + \alpha e^{-\beta(t_i - t_{i-1})}(1 + R(i-1))] \end{aligned} \tag{4.13}$$

where $R(i-1)$ is defined as Equation (4.12) and we have $R(1) = 0$.

4.2.2 Spectrum Density

The spectral density of a time-series $\{t_i\}_{i=1,\ldots,N}$ is that how the variance of the data $\{t_i\}_{i=1,\ldots,N}$ is distributed over the frequency components into which $\{t_i\}_{i=1,\ldots,N}$ may be decomposed. The nature of the spectral density gives useful information about the nature of $\{t_i\}_{i=1,\ldots,N}$, for example, whether it is periodic or not. In this section we apply this technique to the Hawkes process to have further understanding of the process.

The spectral density $f(w)$ of an univariate Hawkes process is defined as Hawkes (1971b):

$$f(\omega) = \frac{\mu\beta}{2\pi(\beta - \alpha)}(1 + \frac{\alpha(2\beta - \alpha)}{(\beta - \alpha)^2 + \omega^2}) \tag{4.14}$$

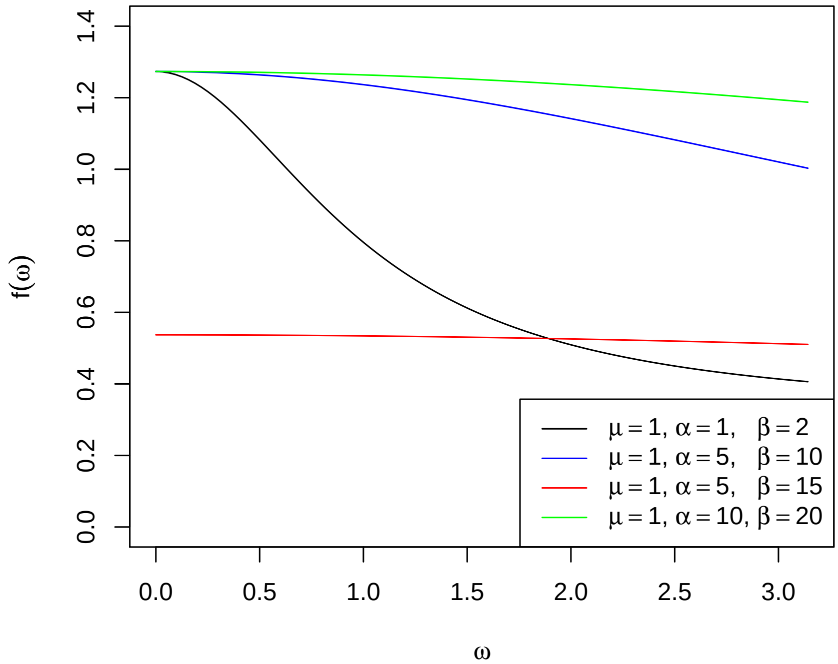


FIGURE 4.2: Spectrum of Hawkes Process

Figure (4.2) gives several samples with different parameter values. Observe that for each intensity function the spectral density is larger at low frequencies and gets smaller at high frequencies. With bigger α the density tends to be bigger (red VS green), which indicates historical events contribute more to the current intensity. The process then is acting as a low-pass filter. This matches what we have seen in the model as α represents the self-exciting effect.

With bigger β the density drops more slowly at the high-frequency end, which indicates most recent events contribute less than older ones for the current intensity. This again matches what is seen in the model as β represents the decaying speed.

Small α and large β result in low density for all frequencies, which suggests little self-exciting effect.

4.2.3 Non-parametric Kernel Estimation

In this section I introduce a non-parametric kernel estimation method for the Hawkes process. The method was proposed by Bacry and Muzy (2014). I will explain how does the method work, and verify the goodness of the method by applying it on synthetic data.

Let $*$ denote the convolution operator, where $A * B_t = \int_{\mathbb{R}} A_{t-s} B_s \mathrm{d}s$, Equation (4.1) can be written as:

$$\lambda(t) = \mu(t) + \phi(t) * \mathrm{dN}_s \tag{4.15}$$

Let Λ denote the unconditional intensity, i.e. $\Lambda = E(\lambda(t))$. And let $g(t)$ denote the density of the measure $E(\mathrm{d}N_t|\mathrm{d}N_0)$, we have:

$$g(t)\mathrm{d}t = E(\mathrm{d}N_t|\mathrm{d}N_0 = 1) - \Lambda\mathrm{d}t \tag{4.16}$$

Hawkes (1971a) proved that:

$$g(t) = \phi(t) + \phi(t) * g(t), \forall t > 0 \tag{4.17}$$

Hawkes (1971a) also proved that in Equation (4.17) $g(t)$ has a unique solution. Bacry and Muzy (2014) proposed an non-parametric kernel estimation method based on this result and the fact that Equation (4.17) is a Wiener-Hopf equation. This type of equations can be solved by the Nyström method introduced by Nyström (1925). Bacry and Muzy (2014) showed that $g(t)$ can be estimated on real data using empirical means. Therefore, one can solve $\phi(t)$ from Equation (4.17) with a measured $g(t)$.

Let's verify this method by estimating the kernels on synthetic data. First, we simulate a couple of the Hawkes process with the exponential kernels:

$$\phi(t) = \alpha e^{-\beta t} \tag{4.18}$$

where the approximate simulation time $T \approx 10^5$ seconds long. We have 2 groups of realisations as the following settings:

- $\mu = 1, \alpha = 1, \beta = 2$
- $\mu = 1, \alpha = 2, \beta = 4$

To measure $g(t)$ on a realized data with length N, we break down the time period T that is expected to largely capture the decay into discrete series $\tau_{i=0,1,\dots,L}$ with step Δ, where $L = \frac{T}{\Delta}$. Then for each τ_i, we count the number of jumps up until $\tau_i + \Delta$, summing this up and dividing it by N to get the estimation of $E(\mathrm{d}N_t|\mathrm{d}N_0 = 1)$. We estimate Λ, which is the expected number of jumps in unit time, by counting the total number of jumps in T as C and then have $\Lambda = \frac{C}{T}$.

Figure (4.3) presents the measurement of $g(t)$ on the simulated data. It can be seen that the correlation itself presents a decaying trend similar as the kernel.

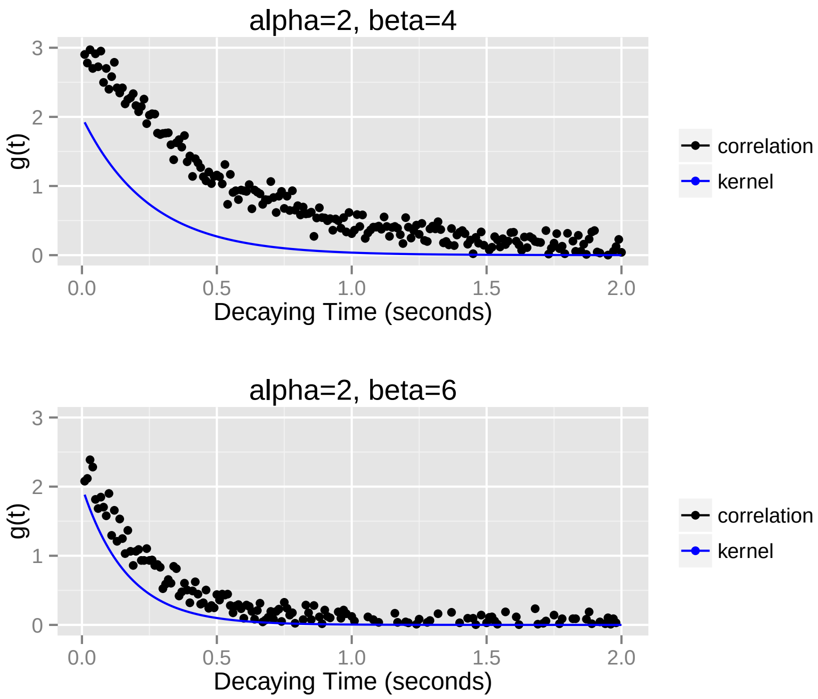


FIGURE 4.3: Correlations for univariate Hawkes processes

Once we have measured $g(t)$ from the data, we can solve Equation (4.17) to get $\phi(t)$. Because Equation (4.17) is a Wiener-Hopf equation, it can be solved using any of the standard techniques for solving this type of equation. We follow Bacry and Muzy (2014) to use the Nyström method introduced by Nyström (1925) using quadrature. We refer the reader who is interested in the implementation details to Press et al. (2007).

Figure (4.4) presents the estimation of the shapes of the kernel for the two realized Hawkes processes described above.

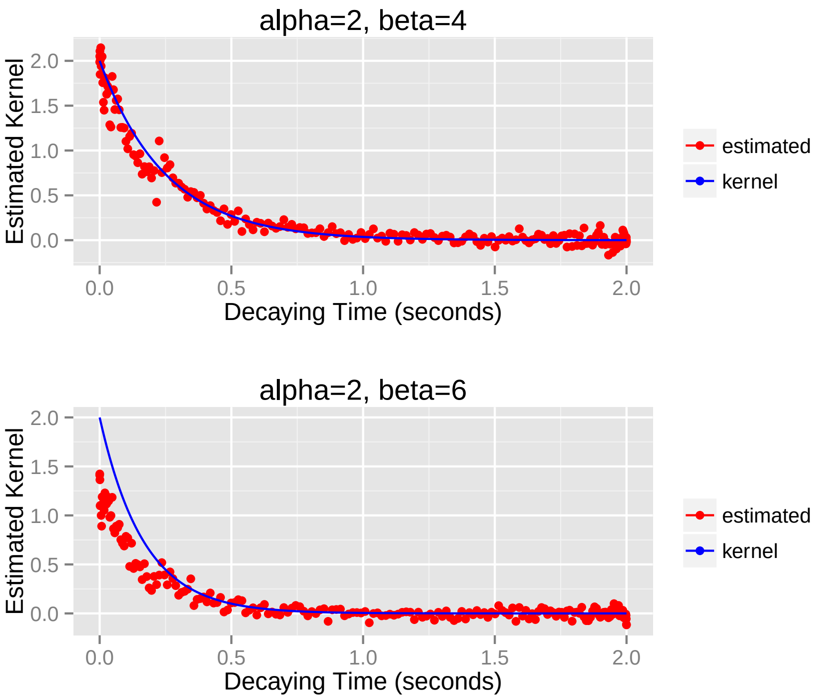


FIGURE 4.4: Kernel Estimation for univariate Hawkes processes

The goodness of this non-parametric approach is that it can reveal more information about the shape of the kernel function without pre-assuming a kernel type, i.e. no matter whether the kernel is an exponential function or a power law function, this approach can give the outline of the function shape from which we can get inspiration about the actual type of the function. As we will see later, this approach does not require the kernel functions to be symmetric when expanding to the multivariate version of the Hawkes process.

Filimonov and Sornette (2013) reflected that when calibrating the Hawkes process model with data, there are several biases that can lead to errors for the branching ratio n.

The importance of measuring the criticality of the branching ratio n is that, from an econometric perspective it quantifies the degree of market efficiency in the sense that the more it gets close to the critical value 1, the more past events can trigger future events, which means less randomness in the market and thus higher efficiency. Although the focus here is on the trading algorithms perspective rather than this econometric one, it is still valuable to justify the way the model is calibrated taking the arguments in Filimonov and Sornette (2013) into account.

With the power of the non-parametric kernel estimation, we can look into the shape of the kernel function rather than the simplified branching ratio. In contrast, the parametric approaches require finding the numerical solution of the minimization of the negative of the log-likelihood function, this could easily have issues where the optimal solution one finds is actually a local optimal rather than a global one. The non-parametric approach does not suffer from this issue. On examining the shape of $g(t)$ estimated from the data, if there is any self-exciting feature in the data, this correlation-like estimation is expected to present some decaying shape.

Filimonov and Sornette (2013) spotted 4 model biases for calibrating the branching ratio n:

- a few outliers can cause strong upward biases on the estimation of n when using power law memory kernels.
- strong effects on n resulting from the form of the regularization part of the power law kernel.
- strong edge effects on the estimated n when using power law kernels.
- the maximum log-likelihood estimation algorithm can result in ending up at a local optima rather than the global one.

The first three are all linked to the power law memory kernels, which we do not necessarily need to choose if we just care about the rough shape of the decaying function. In the context of a trading strategy, it is more important to know if there is a self-exciting feature presented in the data than how fast the self-exciting feature decays. The last one for the maximum log-likelihood estimation algorithm is avoidable if using the non-parametric kernel estimation approach.

Filimonov and Sornette (2013) also demonstrated calibration of the Hawkes process on a mixture of pure Poisson processes can lead to serious bias for n. This could be a serious issue because it means the calibration can be confused to give self-exciting features from the data which is completely random. Thus we specifically review this bias by examining a synthetic time series as Filimonov and Sornette (2013), i.e. construct the series by concatenating a group of independent Poisson processes with different intensity λ_i randomly picking up from range $[1, 10]$. Instead of calibrating the data for n, we first look at the empirical correlation $g(t)$ estimated on the data. We expect the estimated $g(t)$ to not present a decaying feature at all because the underlying data is essentially random.

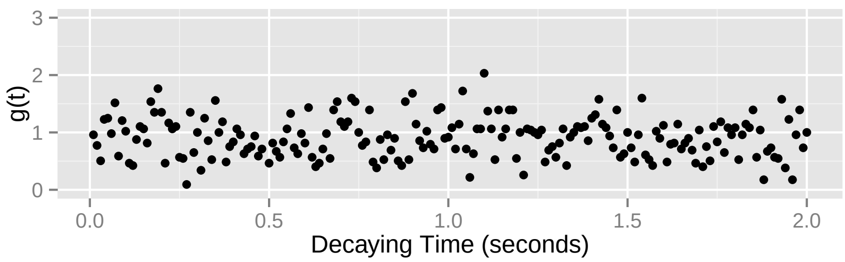


FIGURE 4.5: g(t) Estimated from Synthetic Poisson Series

Figure (4.5) graphs the $g(t)$ estimated from the synthetic Poisson process. From the graph it can be seen that it features lots of noise and is very different than the graph shown in Figure (4.3).

4.2.4 Goodness of Fit

Let $\lambda(t)$ be the intensity function of a point process, the integral of the intensity function

$$\Lambda(t_i) = \int_{t_i}^{t_{i+1}} \lambda(t)\mathrm{d}t \tag{4.19}$$

is a monotonically increasing function because $\lambda(t)$ is non-negative. If we consider the random time change $\tau = \Lambda(t)$ from t to τ, then $\{t_i\}$ is transformed one-to-one into $\{\tau_i\}$, where τ_i represents the expected number of events in $[t_i, t_{i+1}]$. Papangelou (1972) shows that $\{\tau_i\}$ has the distribution of a stationary Poisson process of intensity 1. Ogata (1988) defines series $\{\tau_i\}$ as residuals of the underlying point process and uses it to evaluate goodness of fit of the Hawkes process to the data.

For any consecutive events t_i and t_{i+1}:

$$\begin{aligned}\Lambda(t_i) &= \int_{t_i}^{t_{i+1}} \lambda(t)\mathrm{d}t \\ &= \int_{t_i}^{t_{i+1}} \mu\mathrm{d}t + \int_{t_i}^{t_{i+1}} \sum_{t_k<t} \alpha e^{-\beta(t-t_k)}\mathrm{d}t \\ &= \mu(t_{i+1} - t_i) + \int_{t_i}^{t_{i+1}} \sum_{t_k<t_i} \alpha e^{-\beta(t-t_k)}\mathrm{d}t \\ &= \mu(t_{i+1} - t_i) + \sum_{t_k<t_i} \alpha(e^{-\beta(t_i-t_k)} - e^{-\beta(t_{i+1}-t_k)})\end{aligned} \tag{4.20}$$

This computation can be simplified with a recursive element. Let us denote:

$$A_i = \sum_{t_k<t_i} \alpha e^{-\beta(t_i-t_k)} \tag{4.21}$$

We observe that:

$$\begin{aligned}A_i &= \sum_{t_k<t_i} \alpha e^{-\beta(t_i-t_k)} \\ &= 1 + e^{-\beta(t_i-t_{i-1})} \sum_{t_k<t_{i-1}} \alpha(e^{-\beta(t_{i-1}-t_k)} \\ &= 1 + e^{-\beta(t_i-t_{i-1})} A_{i-1}\end{aligned} \tag{4.22}$$

Let $A_0 = 0$, finally we have:

$$\Lambda(t_i) = \mu(t_{i+1} - t_i) + \frac{\alpha}{\beta}(1 - e^{-\beta(t_{i+1}-t_i)})A_i \tag{4.23}$$

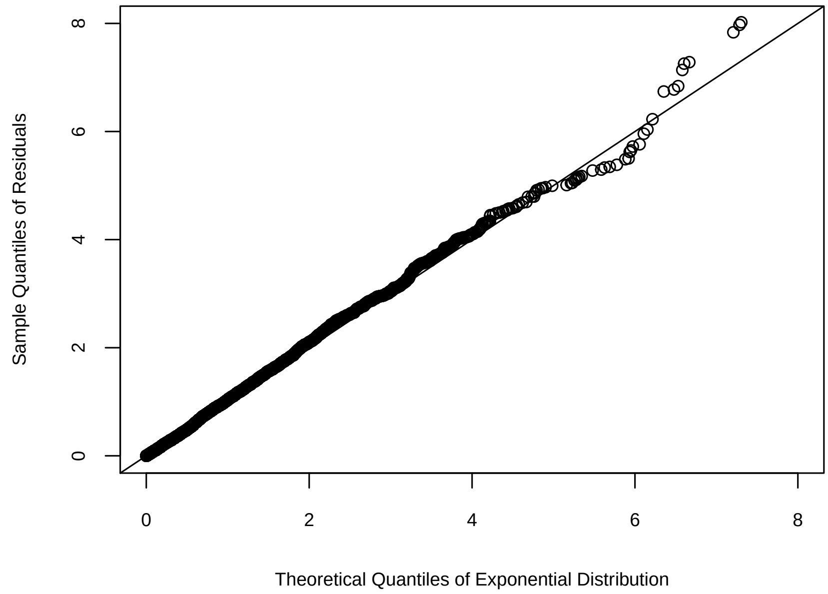


FIGURE 4.6: Q-Q plot of residuals of a simulated univariate Hawkes process (with $\alpha = 2$ and $\beta = 4$) against an exponential distribution

Figure (4.6) shows that the residuals of the simulated process follow the theoretical distribution (an exponential distribution with rate 1) well.

4.2.5 Conditional Intensity

With historical events up to time t_i being observed, we can vary the expected value of t_{i+1} as $\hat{t}_{i+1}$ from 0 to ∞ to calculate the value of τ_1. We have a function of $\hat{t}_{i+1}$ as $f(\hat{t}_{i+1})$:

$$f(\hat{t}_{i+1}) = \tau_i = \int_{t_i}^{\hat{t}_{i+1}} \lambda(t)\mathrm{d}t \tag{4.24}$$

We know $\{\tau_i\}$ follows a stationary Poisson process with intensity 1. Thus the delta series $\{\tau_{i+1} - \tau_i\}$ follows an exponential distribution. The Cumulative Distribution Function (CDF) of the exponential distribution is a monotonically increasing function. We can map the points on the curve of CDF function of exponential distribution one-to-one to the potential values of $\hat{t}_{i+1}$. Each point (x, y) on the curve of CDF represents that with probability y of a real-valued random variable X with exponential distribution will be found at value less than or equal to x. This suggests that given $\hat{t}$ we know the probability that $\hat{t}_{i+1}$ will be found less than or equal to $CDF(f(\hat{t}_{i+1}))$. Since this requires the history $\{t_0, ..., t_i\}$, we have a conditional expectation of the probability distribution of $\hat{t}_{i+1}$.

Given that the CDF of exponential distribution is:

$$CDF(x) = 1 - e^{-x} \tag{4.25}$$

The probability of x being less than value $\hat{x}$ is given by $CDF(\hat{x}) = 1 - e^{\hat{x}}$. So with a given probability p, we can solve the value of $\hat{x}$ as:

$$\begin{aligned} 1 - e^{-\hat{x}} &= p \\ \hat{x} &= -Log(1-p) \end{aligned} \tag{4.26}$$

Thus given a probability p and the history series $\{t_1, ..., t_i\}$. Denote the upper bound of t_{i+1} as $\hat{t}_{i+1}$, we have

$$\int_{t_i}^{\hat{t}_{i+1}} \lambda(t)\mathrm{d}t = -Log(1-p) \tag{4.27}$$

Applying Equation (4.23):

$$\begin{aligned} \mu(\hat{t}_{i+1} - t_i) + \frac{\alpha}{\beta}(1 - e^{-\beta(\hat{t}_{i+1}-t_i)})A_i &= -Log(1-p) \\ A_i &= 1 + e^{-\beta(t_i - t_{i-1})}A_{i-1} \\ A_0 &= 0 \end{aligned} \tag{4.28}$$

Because A_i has a recursive form with exponential expression, we do not have an analytical solution to Equation (4.28). But for a given history $\{t_{0,...,i}\}$, A_i is fixed, consequently we can numerically solve Equation (4.28) to get the value of $\hat{t}_{i+1}$. In this way, we are able to predict $\hat{t}_{i+1}$ conditional on a given probability p, with knowing the history $\{t_{0,...,i}\}$ and the parameters of the underlying Hawkes process. The link between p and $\hat{t}_{i+1}$ is the nice property of the Hawkes process that the transformed time series $\{\tau_i\}$ follows a stationary Poisson process of intensity 1.

In a limit order book market, one important question is to estimate how long will it take for the stimulating effect to fully decay? For a liquid stock, the effect should decay very fast as the participants observe the innovations quickly and react in real time, while for an illiquid stock the effect can decay more slowly as there are less participants and less trading activities.

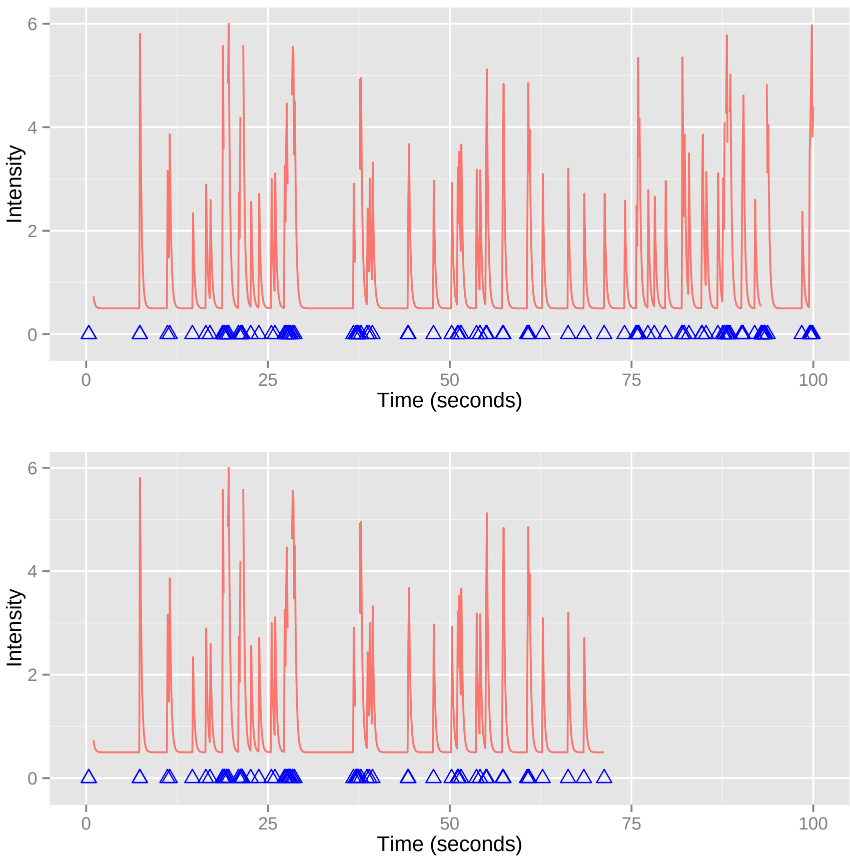


FIGURE 4.7: Two Realisations of a Hawkes process ($\mu = 0.5, \alpha = 3, \beta = 5$)

Figure (4.7) shows two realisations of a simulated Hawkes process with the same parameters and random seed but different stop time. The event history (blue triangles) shown in the bottom chart is exact the same as the part in the top chart up until time 72. After time 72, the realisation shown in the bottom chart stopped simulation, while the one shown in the top chart continues until time 105. The charts show that the conditional intensity at the time of the last event of the realisation shown in the top chart is larger than the one at the time of last event of the realisation shown in the bottom chart. This suggests the shape of the function $g(\hat{t}_{i+1})$ (4.24) will be different for the two time.

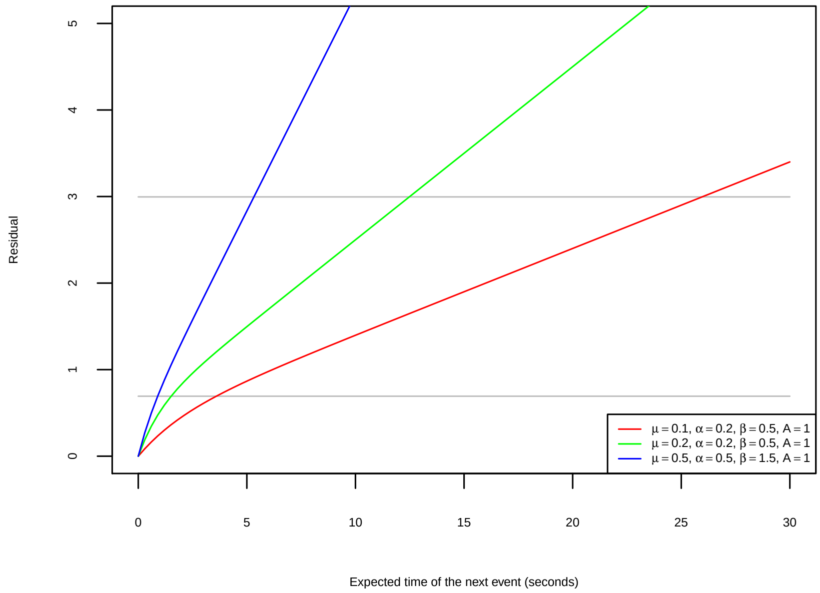


FIGURE 4.8: Conditional Residuals

Figure (4.8) graphs the shape of the conditional residual function $\Lambda(t_i)$ for three realisations of the Hawkes process with different set of parameters at a specified time t_i. The red curve is for a Hawkes process with parameters $\mu = 0.1, \alpha = 0.2, \beta = 0.5$, the green curve is for a Hawkes process with parameters $\mu = 0.2, \alpha = 0.2, \beta = 0.5$, the blue curve is for a Hawkes process with parameters $\mu = 0.5, \alpha = 0.5, \beta = 1.5$. I set the time t_i for the three realisations in a way to make they have the same value for A_i. This makes the shapes of their residual function $\Lambda(t_i)$ comparable. From the figure, we can see that the blue curve increases faster than the green one, while the green one increases faster than the read one. This is because their Hawkes parameters result different conditional residuals for the same A_i according to Equation (4.23).

The two gray lines in Figure (4.8) correspond to two values of the probability p in Equation (4.27). The top gray line represents probability 0.95, the bottom line is for probability 0.5. The intersection points of the gray lines and the coloured lines tell the expected time of the next event for the three realisations. We can see that for the same probability value (e.g. 0.95 which maps the top gray line), the realisation represented by the blue curve expects the next event much earlier than the ones represented by the green and red curves. This confirms the intuition that we can use the residual function $\Lambda(t_i)$ to predict the time of the next event for the Hawkes process.

With a given probability P, we can have a simple dynamic prediction algorithm for the univariate Hawkes process. For the observed events $\{t_{i=0,1,2,\ldots}\}$ we solve the Equation (4.28) with $p = P$ to get the predicted time of the next event $\hat{t}_{i+1}$. Depending on the different parameters of the underlying Hawkes process one can choose different confidence level to optimize the prediction algorithm.

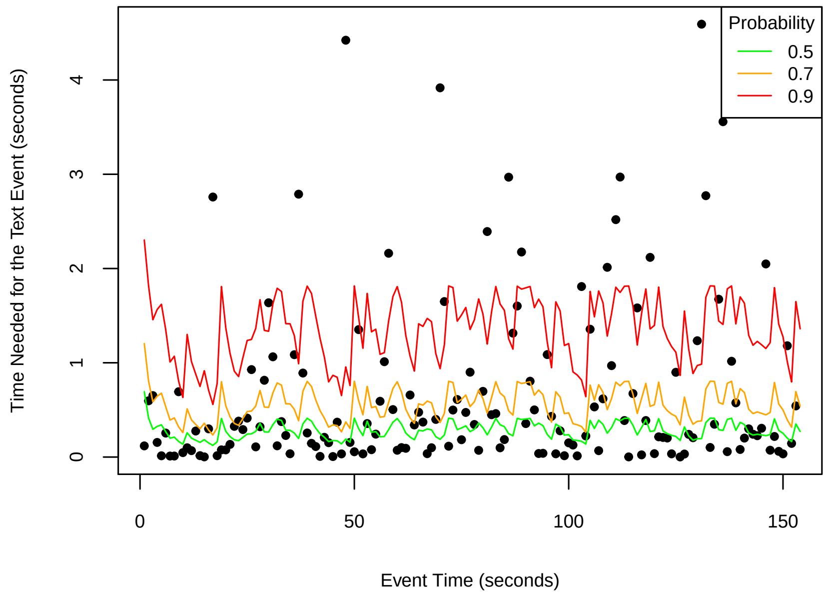


FIGURE 4.9: Prediction of The Next Event Time ($\mu = 1, \alpha = 1, \beta = 2$)

Figure (4.9) outlines the prediction results of the next event time for a realisation of the Hawkes process with parameters $\mu = 0.1, \alpha = 0.2, \beta = 0.5$. The black dots are the real time needed for getting the next event. The red, yellow and green curves represent the prediction made with probability 0.9, 0.7 and 0.5 respectively.

From Figure (4.9), we observe that the prediction made with a smaller probability captures the lower bound of the next event time well, while the prediction made with a bigger probability captures the upper bound of the next event time better. This matches the intuition that a smaller probability represents an aggressive strategy that tends to predict that the next event is likely to occur in shorter time, while a bigger probability represents a conservative strategy that tends to wait more to be sure to see the next event.

We gain a slight predictability from the Hawkes process with this approach. It essentially maps the conditional intensity back to the event occurring time, and thus enables us to make a one step ahead prediction given an observed events history. Although the residual process itself is a Poisson process with intensity 1, which is an unpredictable process meaning the time of each event is independently distributed and purely random, the mapping does link the conditional history to this known distribution and thus makes the time of the next event in the original Hawkes process dependant on the history. In this sense we get a way to model the conditional events time with a given confidence level.

4.3 Empirical Study

In this section, I apply the Hawkes process to real market data to verify the effectiveness of this model. We have shown that the Hawkes process has a good theoretical foundation, it captures the self-exciting feature nicely. But how does it work with live market data? We can only find out by actually using the real data to verify.

I have two goals to achieve for applying the Hawkes process. The first goal is to verify that whether the data has a self-exciting feature or not. We can do this by using the non-parametric kernel estimation method. Because the non-parametric method can visualize the shape of the kernel function, we can easily find if there is a decaying trend in the function. Only when we can confirm that the data has a self-exciting feature, we can then try to capture that with models.

The next goal is to use the Hawkes process to capture the self-exciting feature in the data of it is confirmed. I do this by calibrating the Hawkes process with the data, and check the goodness of fit with standard statistical technique. I use the parametric version of kernel estimation method for this purpose, to quantify the self-exciting feature.

This section is organized as following. Section (4.3.1) describes the data we use in our analysis. Section (4.3.2) presents some simple statistics of the data, to demonstrate the rationality of seeking self-exciting feature in the data. Section (4.3.3) tells how do we pre-process the raw data to make it fit for the application of the Hawkes process. Section (4.3.4) gives results of applying non-parametric kernel estimation method on the preprocessed data. Section (4.3.5) shows the goodness of fit of the Hawkes process on the data.

4.3.1 The Data

The empirical results are based on the analysis of market data from the National Stock Exchange (NSE) at the beginning of August 2012. The NSE is an order driven market. Market data corresponds to a stream of events affecting the limit order book. There are four types of events:

- *New Order* events that correspond to limit or market orders indicating the participant's intent to buy or sell.
- *Cancel Order* events which are instructions to cancel an existing order.
- *Order Modification* events which are instructions to modify an existing order, e.g. amending the price and/or the quantity.
- *Execution* events which are representative of a trade occurring between two counter-parties at a specific price for a given quantity.

Order modifications are the most frequent events among the four types. This suggests that algorithmic trading strategies regularly adjust their position to react to new information.

In 2011, the NSE was the third largest exchange by traded number of both single stock futures and stock index futures World Federation of Exchanges (2012). It had 161 and 156 million contracts traded respectively for single stock futures and stock index futures. On average there are 3GB uncompressed daily market data for stock and 7GB for futures. Many cash future arbitrage strategies are running on the NSE where participants try to statistically exploit the price relationship mismatch between the future instrument and its underlying stock.

Most of the executions happened for a small quantity, which introduces a lot of noise into the duration process of execution events as traders may post small orders on the top of book to game other algorithmic participants. Moreover, a Hawkes process tracks all the past events when calculating intensity, so the number of events increases the computational complexity.

4.3.2 Simple Statistics

Before we start fitting data into the model, let's examine some basic statistics of the data as shown in Table (4.1). The table presents the statistics of executions for all 50 stocks that composites the NIFTY index from January 1 2012 to January 24 2012 which, in total, comprise 18 effective days, ordered by the average daily volume $\bar{V}$. The third and fourth column, $\bar{q}_{Q10\%}$ and $\bar{q}_{Q30\%}$, represent the 10% and 30% quantile of the execution size of all the executions in the 18 trading days. It tells us that obviously a large proportion of the executions are very small ones. The fifth column $\bar{q}$ is the average execution size, the numbers suggest that the range of the sizes is not insignificant, lots of big executions do happen in the market, which pull the average size far away from the small values. The sixth column, Daily $\bar{q}$ was taken by averaging the average execution size per day over the 18 trading days, the numbers largely agree with the fifth column, which means that it can be a good proxy for the average of all in a daily perspective. The seventh column represents the average number of executions per day for each symbol, an interesting finding is that it does not necessarily need to match $\bar{V}$.

	Symbol	$\bar{V}$	$\bar{q}_{Q10\%}$	$\bar{q}_{Q30\%}$	$\bar{q}$	Daily $\bar{q}$	$\bar{N}$	$\bar{\rho}$
1	GRASIM	53474	1	2	11	12	4243	1.47
2	ASIANPAINT	63396	1	1	13	10	6553	1.60
3	ULTRACEMCO	141624	1	2	18	24	6799	1.66
4	SIEMENS	167466	1	3	21	22	7418	1.98
5	DRREDDY	266200	1	3	17	16	17234	1.21
6	ACC	275880	1	3	21	24	11619	1.65
7	PNB	381392	1	5	25	26	14941	1.67
8	BANKBARODA	407940	1	5	34	36	11456	1.92
9	BPCL	478123	1	4	28	29	16660	1.77
10	SUNPHARMA	572691	1	3	30	33	19504	2.27
11	HEROMOTOCO	607747	1	3	18	19	32494	1.38
12	RANBAXY	708197	1	3	41	40	17680	1.64
13	LUPIN	848054	1	2	22	24	34909	1.42
14	BAJAJ-AUTO	929920	1	3	19	19	46354	1.71

15	CIPLA	980136	1	6	55	58	17674	2.02
16	KOTAKBANK	981561	1	6	37	39	25101	1.76
17	HCLTECH	1277683	1	6	49	46	29133	0.96
18	MARUTI	1436500	1	5	23	22	59861	1.44
19	GAIL	1498023	1	5	47	52	30915	1.40
20	AMBUJACEM	1614063	1	8	95	102	16877	0.86
21	WIPRO	1666046	1	6	57	61	27036	1.22
22	INFY	1766523	1	3	17	15	109202	0.82
23	HDFC	1966373	1	5	42	42	47692	1.25
24	CAIRN	1988142	1	6	53	56	36978	1.41
25	JINDALSTEL	2069116	1	5	47	46	46072	1.32
26	ONGC	2156508	1	5	53	51	42032	1.26
27	HINDUNILVR	2185634	1	6	51	54	41331	1.00
28	M&M	2449734	1	5	37	38	64452	1.41
29	TCS	2452096	1	5	29	28	83745	0.91
30	SBIN	2468781	1	4	23	23	109563	1.08
31	HDFCBANK	2735305	1	7	53	55	50639	1.45
32	RELINFRA	2810405	1	9	50	49	56542	1.30
33	LT	3002595	1	5	27	27	111237	1.38
34	AXISBANK	3279246	1	5	33	32	98051	1.55
35	NTPC	3295282	1	7	137	128	25675	1.19
36	SESAGOA	3303821	1	10	79	79	42386	1.13
37	BHARTIARTL	4193785	1	8	71	65	64154	1.02
38	POWERGRID	4483786	1	11	179	210	23897	2.16
39	BHEL	4562677	2	7	60	58	77189	1.52
40	TATAPOWER	4581647	1	14	174	192	23484	1.52
41	COALINDIA	4628728	1	9	67	67	68411	1.13
42	RELIANCE	5543188	1	5	42	44	124912	1.63
43	ICICIBANK	5615540	1	6	46	47	119201	1.32
44	TATASTEEL	6874603	2	10	56	57	120263	0.94
45	ITC	7562008	1	10	111	117	64444	1.45
46	HINDALCO	10454941	2	19	148	155	68022	1.02
47	DLF	11498489	1	14	129	114	97415	1.60
48	IDFC	11797954	2	20	170	180	66335	1.32
49	TATAMOTORS	16369694	2	11	107	109	151402	1.16
50	JPASSOCIAT	19059954	1	29	312	322	59508	1.05

Table 4.1: Statistics of Executions

The last column of Table (4.1) which we name as $\bar{\rho}$ is, however, a measurement to give a feel for the key factor of total volume dynamics. As we can imagine, there are two factors that contribute to the daily total volume dynamic, one is the number of executions, and the other is the average size of execution. The question is, when the total volume of a day is observed bigger/smaller than the "normal" size (which can be a moving average over historical days), which factor contributes more to it? Do we expect to see more/less number of executions, or a bigger/smaller average execution quantity? This question is important because it can cause different modeling strategies. The raw market data that one receives from an exchange like the NSE is composited from individual messages as discrete time series. That said, each execution is represented by a message, which contains a timestamp and other information like price and quantity. If it is just the number of executions that makes the difference to total volume, we can then model the raw data as a point process where each message itself is a point and the time associated with that point will be the raw timestamp. If it is the average size of execution, we will need to model the data as a marked point process taking execution size as the mark of each point, or, conflate the messages with events whose size equal to the average size and transform the time accordingly. To answer this question, we need to analyse the data to find clues.

We define ρ as following. For symbol j, first take r_1 as the ratio between the total volume $V_{i,j}$ for day i and the moving average total volume $\hat{V}_{i,j}$ of L days before day i. Then take r_2 as the ratio between the number of executions $N_{i,j}$ for day i and the moving average number of executions $\hat{N}_{i,j}$ of the same L days before. So we have ρ as:

$$\rho = \begin{cases} \frac{r_1}{r_2} = \frac{\frac{V_{i,j}}{\hat{V}_{i,j}}}{\frac{N_{i,j}}{\hat{N}_{i,j}}} & \frac{V_{i,j}}{\hat{V}_{i,j}} \geq 1 \\ \frac{r_2}{r_1} = \frac{\frac{N_{i,j}}{\hat{N}_{i,j}}}{\frac{V_{i,j}}{\hat{V}_{i,j}}} & \frac{V_{i,j}}{\hat{V}_{i,j}} < 1 \end{cases}$$

If it is the number of executions that mostly decides the total volume of a day, we would expect to see $\rho < 1$.

The last column of Table (4.1) shows the average of ρ across all sample days as $\bar{\rho}$. The data presented in that column rejects this hypothesis. Only 2 out of 50 stocks have $\bar{\rho}$ slightly less than 1, while the others are greater than 1. This leads us to a way of conflating the messages into events that gives more weight to the average size of the intra-day executions.

4.3.3 Data Preprocessing

Because of the descriptive nature of Hawkes processes, we focus on how well the data fits the model to answer to what extent we can describe the data as "self excited". Particularly, we try to examine the volume data in this chapter.

Intuitively, we have raw timestamps for each trade in the data, so that we can fit it directly into the Hawkes model, and this is the common strategy seen in the literature. The shortcoming of this strategy is that the size of each trade can vary dramatically, which makes the impact of each trade very different. We examine this simple strategy first in this chapter. A further extension can be the introduction of a marked version of the Hawkes process, with the size an associated "mark" of the trade. This of course increases the calculation complexity of the model which may be not necessarily be needed. To simplify the calculation process without losing the purpose of taking trade sizes into account, another strategy simply consolidates the trades into fixed size chunks, and fits the transformed timestamps into the model.

To decide the size of the consolidated chunk, we need to find a strategy to find the proper quantity for each stock. The fact that the total volume for each stock at day varies randomly, and the intra-day volume distribution generally follows a U-shaped curve which implies that the volume process between days is not a continuous one, suggests that calibrating the data over many days as a single process is very difficult. We do not have a good strategy to normalize the volume per day to give a unified metric for each day. Therefore we decide to group the data by stock, fit the data at each day individually to a Hawkes process, and take an average over the fitted model to get an overview model for each stock.

For each day, each stock k, we have raw time series data as $\{t_i, p_i, q_i\}_{i \in 1,\ldots,N_k}$ where t_i, p_i, q_i represent the timestamp, price and quantity of the i^{th} execution, and N_k is the total number of executions. We first consolidate the series by merging executions to size $\bar{q}_i = \frac{\sum_i^{N_k} q_i}{N_k}$, to do this we fit a linear curve to the accumulated quantity series $t_i, \sum_j^i q_j$, then find the timestamps for the fixed interval series $\{n \times \bar{q}_n\}_{n \in 1,\ldots,N_k}$.

The following example illustrates this process. Taking a sample execution series as shown in Table (4.2), we compute the averaged executions size as 2. So the consolidated executions will have timestamps at 2, 4, 6, ..., 20.

	Time.	Quantity
1	1	1
2	2	1
3	5	1
4	8	4
5	9	3
6	10	1
7	15	1
8	17	3
9	19	3
10	20	2

TABLE 4.2: Sample Executions

The accumulated quantity is outlined in the upper part of Figure (4.10), while each dot represents the accumulated quantity at the corresponding time, the solid line is a piecewise linear fit of the data points. The lower part of Figure (4.10) presents how we find the timestamp for the consolidated series, the intersect points of the red dotted lines and x axis give the timestamps of the consolidated events. Compare these timestamps with the intersect points of the blue dotted lines and the axis in the upper part which are the original timestamps, one can see the difference introduced by this approach.

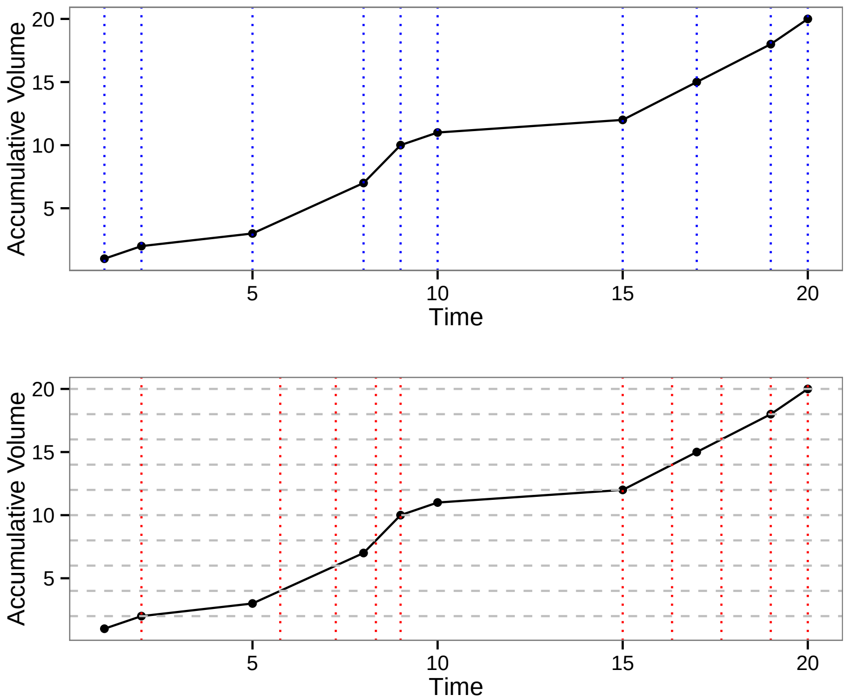


FIGURE 4.10: Consolidate Executions

To eliminate the U-shape seasonality in intra-day volume distribution, we use a moving average cumulative curve $f(t)$ extracted from L days of historical data and detrend the data for the testing day using this curve with the following algorithm:

1. Break the trading period into equal-length windows. Each window w_i has length B, starts at w_i^{start} and ends at w_i^{end}. Use the cumulative curve $f(t)$ to calculate the relative volume distribution proportion of window w_i as $p_i = f(w_i^{end}) - f(w_i^{start})$.

2. Convert the monotonic increasing time series of trades t_i into a difference series $\Delta_i = t_i - t_{i-1}$, where $t_0 = 0$.

3. For each Δ_i, if the corresponding time t_i falls into window w_i: $w_i^{start} < t_i \leq w_i^{end}$, scale Δ_i to $\hat{\Delta}_i = \Delta_i * p_i$.

4. Convert the scaled series $\hat{\Delta}_i$ to time space $\hat{t}_i$ such as $\hat{t}_i = \hat{\Delta}_{i-1} + \hat{\Delta}_i$.

Thus the converted time series $\hat{t}_i$ is "flattened" by the curve $f(t)$, such that the trades which happened in busy periods, like near market open and close, take more time between each other, and the ones that happened in quiet periods like lunch time take less time.

This U-shape intra-day seasonality can vary randomly for each day, which means the moving average curve based on historical data may not necessarily reflect the true imbalance per day. This inaccuracy may affect the calibration of the Hawkes process in the sense that the error may be taken as a false interpretation of the self-exciting effect. So we also try to use a true intra-day curve, which is a smooth version of the real accumulative curve by fitting a polynomial curve, as an alternative to eliminating the intra-day seasonality solution.

Combining the ways we select the timestamps and eliminating intra-day seasonality, we have 4 different approaches for calibrating the data into the Hawkes process:

1. Detrend the raw time series of trade events directly with the moving average curve built on historical data, without consolidation. Fit the resulted time series into the Hawkes process.

2. Detrend the raw time series of trade events directly with the true intraday curve of the testing day, without consolidation. Fit the resulted time series into the Hawkes process.

3. Consolidate the raw time series of trade events by merging trades into fixed size, and interpolating time with linear curve. Detrend the U-shape seasonality with a moving average curve built on historical data. Fit the resulted time series into the Hawkes process.

4. Consolidate the raw time series of trade events by merging trades into fixed size, and interpolating time with linear curve. Detrend the U-shape seasonality with the true intraday curve of the testing day. Fit the resulted time series into the Hawkes process.

4.3.4 Non-parametric Kernel Estimation with Real Data

I first try to apply the non-parametric kernel estimation method to the data. The benefit of using the non-parametric method is that it can visualize the shape of the kernel function. By looking at the visualized kernel function, we can verify that whether the data has the self-exciting feature or not. Recall that in Section (4.2.3), both the measured $g(t)$ and the estimated kernel of the synthetic Hawkes process data present a clear decaying trend. If the real data does have a self-exciting feature, we would expect to see a decaying trend in the measured $g(t)$ and the estimated kernel obtained on the real data too.

In this section I use the 4 approaches described in Section (4.3.3) to the NSE data. For each result of the 4 approaches, we apply the non-parametric kernel estimation method to check both $g(t)$ and the estimated kernel. I then look at the shape of both $g(t)$ and the estimated kernel to verify if there is a self-exciting feature. I have repeated this process to all stocks listed in NIFTY, and with all trading days in the data set and found the results are consistent. To illustrate the findings, I pick up the data on day January 3 2012 for 4 stocks, i.e. LT, RELIANCE, WIPRO, TATAMOTORS, and show the graphs in the following part of this section.

I start by measuring $g(t)$ on the data processed with approach 1. This approach uses the original raw data directly. Can we observe a decaying trend in the result? Figure (4.11) graphs the $g(t)$ measured with this approach. We do observe a slight decaying trend in each of the graph.

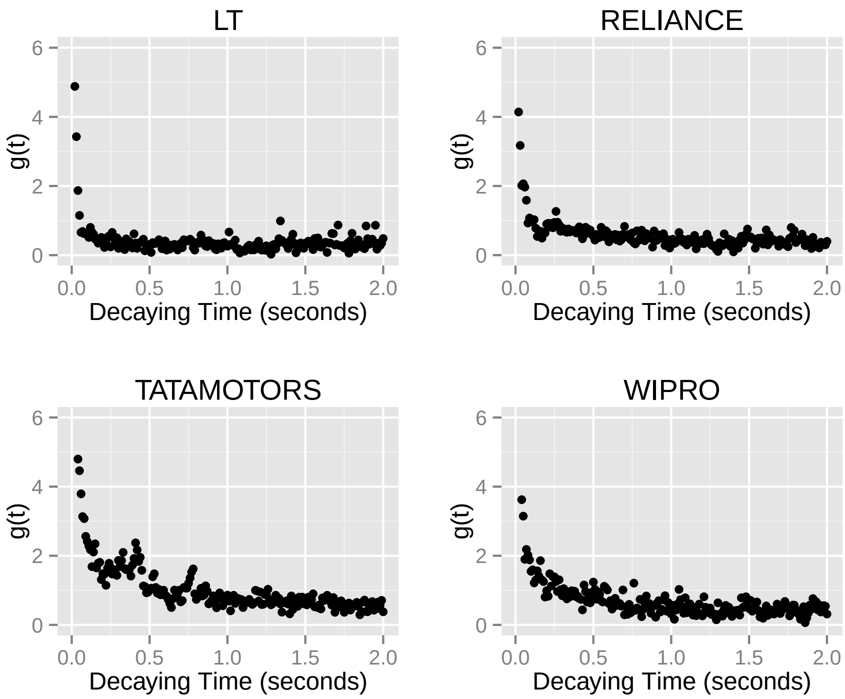


FIGURE 4.11: Measured $g(t)$ on Data Processed with Approach 1

Figure (4.12) show the estimated kernel. All the 4 graphs do not show any decaying effect. This suggests that the raw data itself does not present self-exciting feature, which is reasonable because we know that the quantity of each trade event can be different and there is a U-shape seasonality in the intraday volume distribution.

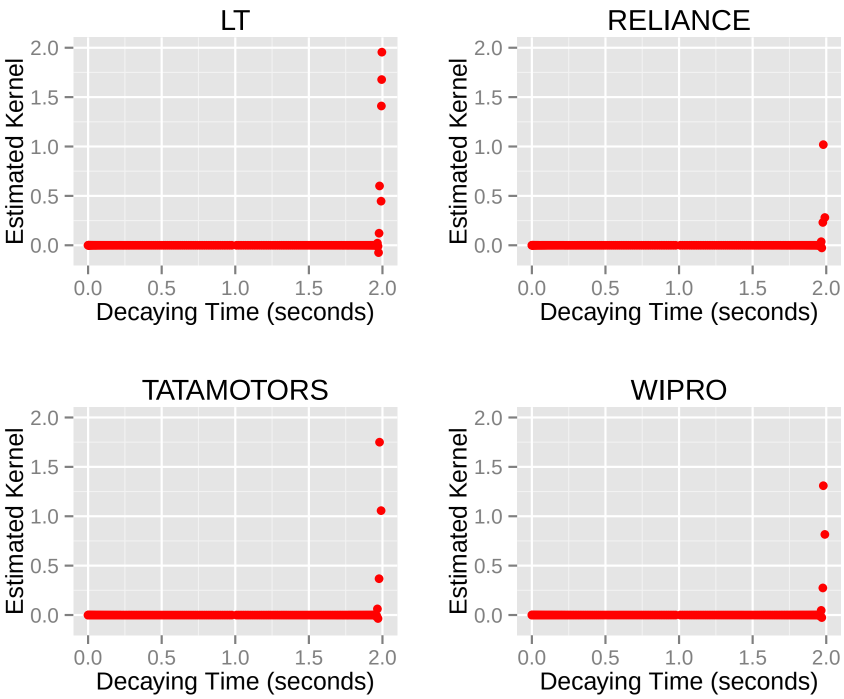


FIGURE 4.12: Estimated Kernel on Data Processed with Approach 1

Next I try to eliminate the U-shape seasonality. Figure (4.13) gives the shape of the measured $g(t)$ with the data processed with approach 2. We can see a decaying trend in each of the graph.

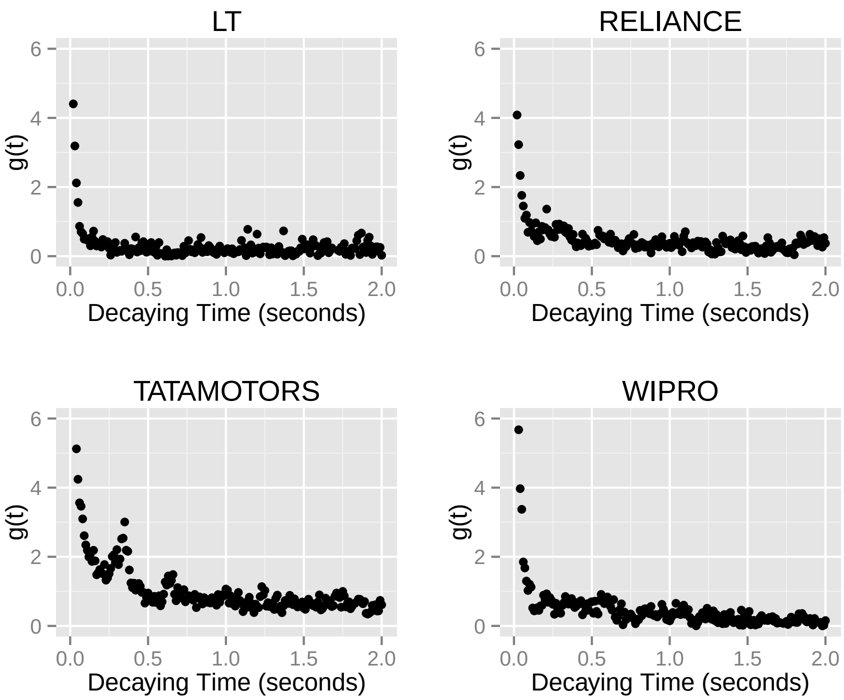


FIGURE 4.13: Measured $g(t)$ on Data Processed with Approach 2

But once again we do not see any decaying feature in the estimated kernel, in any of the 4 graphs shown in Figure (4.14). It seems that simply removing the U-shape seasonality is not enough.

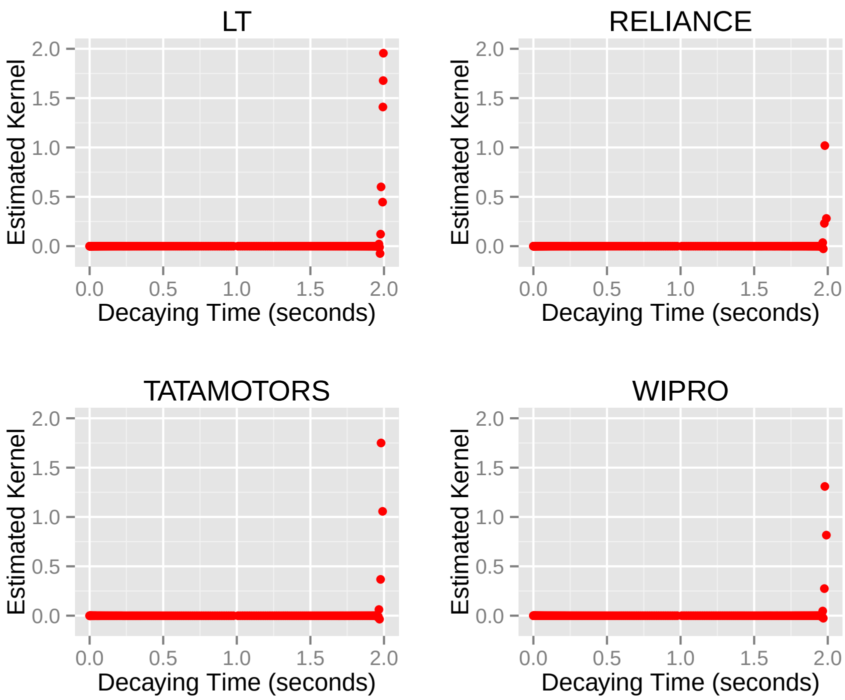


FIGURE 4.14: Estimated Kernel on Data Processed with Approach 2

Next, I use approach 3, which does both consolidation and interplating timestamps, and removing the U-shape seasonality with a moving average curve built on historical data. Figure (4.15) shows the shape of $g(t)$ measured on the data processed with this approach. We see a decaying trend in each of the graphs.

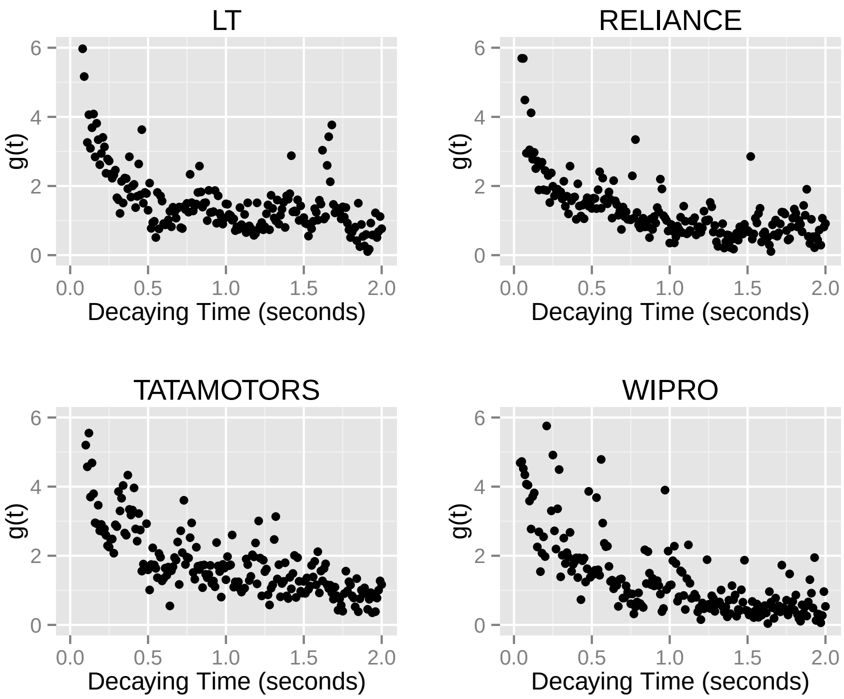


FIGURE 4.15: Measured $g(t)$ on Data Processed with Approach 3

Figure (4.16) outlines the shape of the estimated kernels. We can see that all the 4 graphs present clear decaying trend. This confirms that with proper data preprocessing, i.e. consolidating trades into fixed-size quantity and removing the intraday seasonality, we do see self-exciting feature in the market data.

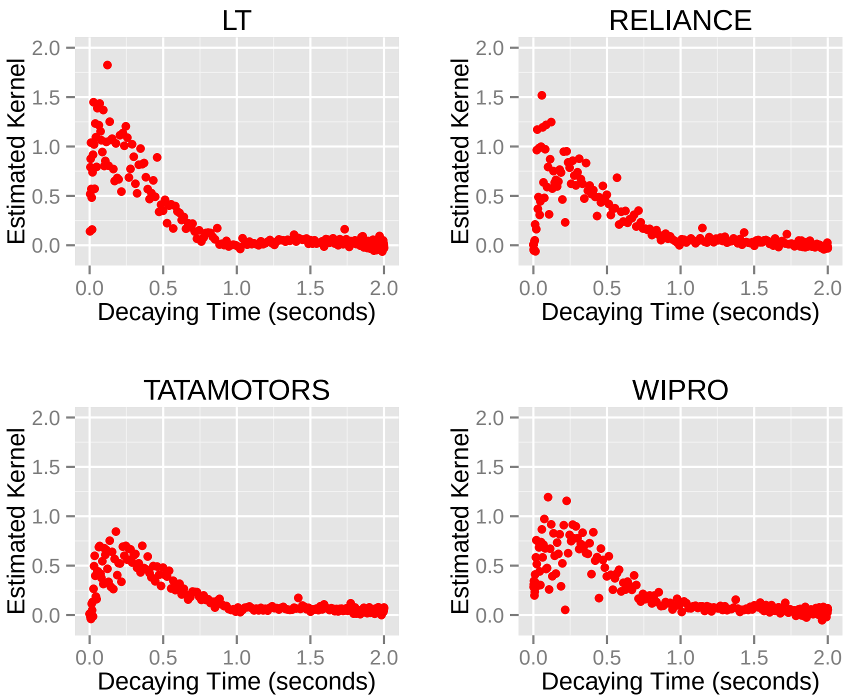


FIGURE 4.16: Estimated Kernel on Data Processed with Approach 3

Figure (4.17) and Figure (4.18) are results for the data processed with approach 4. This approach does both consolidation and removing seasonality. The only difference between this approach and approach 3 is that this one uses the true intraday volume distribution curve rather than the historical moving average curve. Figure (4.17) and Figure (4.18) present similar decaying feature, which again confirms that with proper data preprocessing, we can indeed find self-exciting feature in the market data.

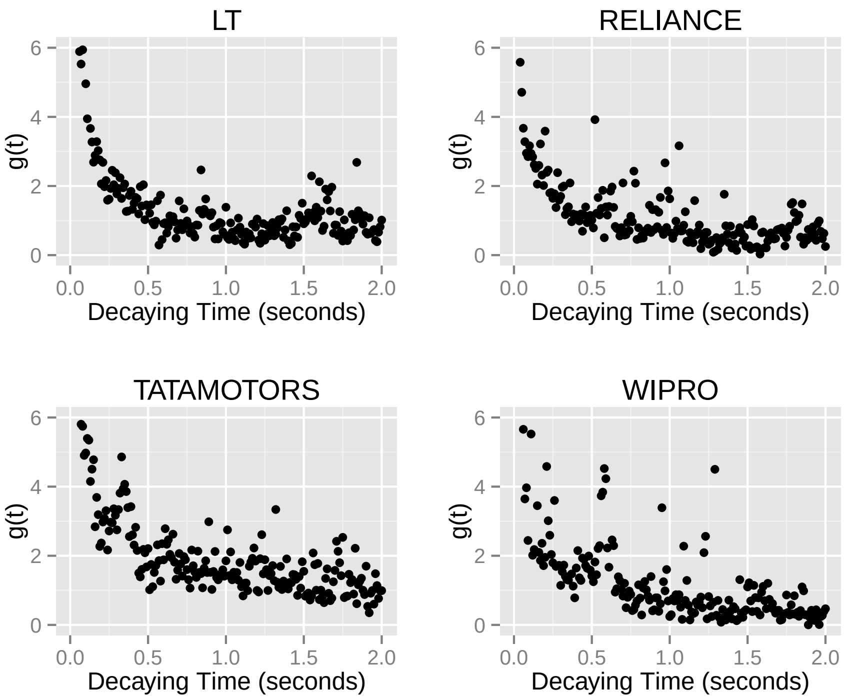


FIGURE 4.17: Measured $g(t)$ on Data Processed with Approach 4

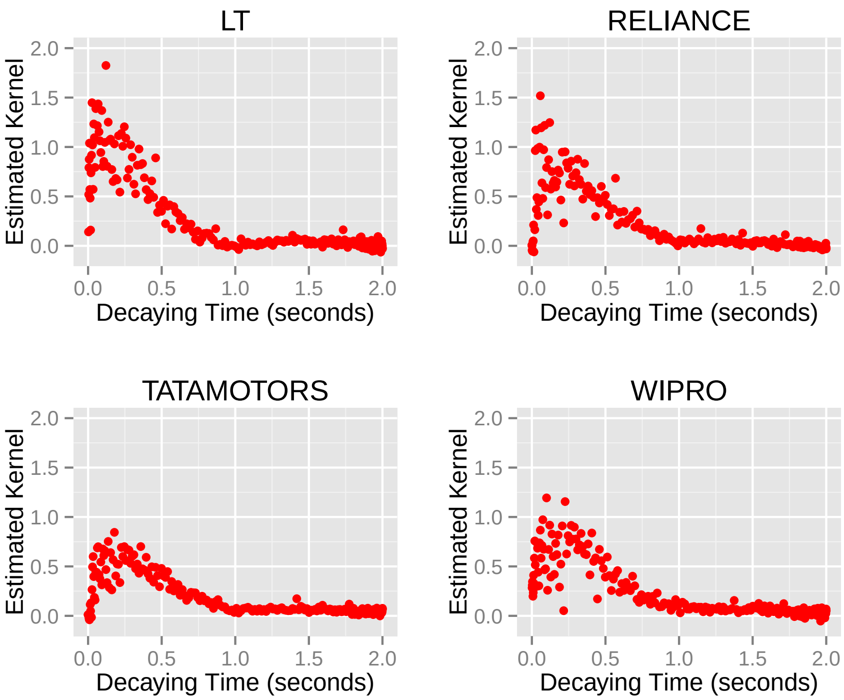


FIGURE 4.18: Estimated Kernel on Data Processed with Approach 4

4.3.5 Goodness of Fit

As mentioned in Section (4.2.4), the residuals of a Hawkes process should have an exponential distribution with rate 1. Thus we can use the Kolmogorov–Smirnov test to compare the residuals of the Hawkes model fitted by the data and the probability distribution of the exponential distribution. To apply this test, we have a null hypothesis as that the market data fits the Hawkes process nicely, which means the residuals follows an exponential distribution with rate 1. So our goal is to use the Kolmogorov–Smirnov test to see if we can reject this hypothesis.

I get two critical values from the Kolmogorov–Smirnov test to quantify the goodness of fit, and to decide whether we can reject the hypothesis. One is the maximum distance D between the empirical distribution function of the residuals data and the cumulative distribution function of the reference distribution, the smaller it is the more likely the null hypothesis is true. The other is P-value P, which is the probability of obtaining a test statistic result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true.

I run the test on 301 days of market data recorded from the NSE from January 2, 2012 to April 8, 2013. Because we need to build a moving average intra-day volume distribution curve based on couple of days, we take the first 30 days to build the initial curve, so the actual test starts from February 17, 2012. That gives 271 effective days for the test. For each day, I fit the data for each stock listed in the NIFTY index to a univariate Hawkes process using the 4 approaches described in Section (4.2.4), apply the Kolmogorov–Smirnov test to the residuals and get the two values D and P. So for the whole test, I have 4 groups of results corresponding to the 4 approaches, each has 271 pairs of D and P for each stock, I take an average over the 271 values for both D and P as the final indicator for goodness of fit. Table (4.3) outlines the numerical results. I denote D_0 and P_0 as the values for approach 1, D_1 and P_1 as the values for approach 2, $\tilde{D}_0$ and $\tilde{P}_0$ as the values for approach 3 and $\tilde{D}_1$ and $\tilde{P}_1$ as the values for approach 4.

	Symbol	D_0	P_0	D_1	P_1	$\tilde{D}_0$	$\tilde{P}_0$	$\tilde{D}_1$	$\tilde{P}_1$
1	HDFC	0.10	0.00	0.19	0.00	0.11	0.00	0.19	0.00
2	ONGC	0.09	0.00	0.20	0.00	0.13	0.00	0.17	0.00
3	TATAMOTORS	0.10	0.00	0.12	0.00	0.07	0.00	0.19	0.00
4	GAIL	0.15	0.00	0.22	0.00	0.10	0.00	0.22	0.00
5	BHARTIARTL	0.08	0.00	0.23	0.00	0.11	0.00	0.22	0.00
6	DLF	0.07	0.00	0.18	0.00	0.32	0.00	0.25	0.00
7	NTPC	0.12	0.00	0.17	0.00	0.33	0.00	0.48	0.00
8	HINDUNILVR	0.12	0.00	0.16	0.00	0.13	0.00	0.16	0.00
9	BAJAJ-AUTO	0.10	0.00	0.12	0.00	0.12	0.00	0.13	0.00
10	JPASSOCIAT	0.09	0.00	0.26	0.00	0.09	0.00	0.26	0.00
11	CAIRN	0.09	0.00	0.17	0.00	0.11	0.00	0.17	0.00
12	HDFCBANK	0.13	0.00	0.19	0.00	0.10	0.00	0.18	0.00
13	LT	0.08	0.00	0.21	0.00	0.07	0.00	0.17	0.00

14	ULTRACEMCO	0.11	0.00	0.13	0.00	0.10	0.00	0.12	0.00
15	GRASIM	0.16	0.00	0.17	0.00	0.37	0.00	0.48	0.00
16	HINDALCO	0.10	0.00	0.18	0.00	0.09	0.00	0.20	0.00
17	PNB	0.09	0.00	0.11	0.00	0.07	0.00	0.12	0.00
18	SESAGOA	0.07	0.00	0.12	0.00	0.08	0.00	0.11	0.00
19	BHEL	0.10	0.00	0.21	0.00	0.09	0.00	0.18	0.00
20	ICICIBANK	0.07	0.00	0.15	0.00	0.08	0.00	0.11	0.00
21	KOTAKBANK	0.13	0.00	0.10	0.00	0.10	0.00	0.10	0.00
22	RELIANCE	0.11	0.00	0.20	0.00	0.13	0.00	0.20	0.00
23	SIEMENS	0.14	0.00	0.15	0.00	0.14	0.00	0.15	0.00
24	JINDALSTEL	0.17	0.00	0.22	0.00	0.10	0.00	0.16	0.00
25	AMBUJACEM	0.10	0.00	0.14	0.00	0.09	0.00	0.14	0.00
26	LUPIN	0.18	0.00	0.20	0.00	0.14	0.00	0.19	0.00
27	DRREDDY	0.12	0.00	0.16	0.00	0.42	0.00	0.39	0.00
28	ACC	0.09	0.00	0.11	0.00	0.12	0.00	0.11	0.00
29	BPCL	0.11	0.00	0.17	0.00	0.13	0.00	0.18	0.00
30	HCLTECH	0.13	0.00	0.33	0.00	0.12	0.00	0.14	0.00
31	IDFC	0.11	0.00	0.20	0.00	0.10	0.00	0.16	0.00
32	COALINDIA	0.09	0.00	0.26	0.00	0.11	0.00	0.21	0.00
33	TCS	0.09	0.00	0.17	0.00	0.11	0.00	0.11	0.00
34	RELINFRA	0.12	0.00	0.23	0.00	0.46	0.00	0.56	0.00
35	HEROMOTOCO	0.08	0.00	0.18	0.00	0.08	0.00	0.17	0.00
36	BANKBARODA	0.11	0.00	0.12	0.00	0.11	0.00	0.13	0.00
37	INFY	0.07	0.00	0.21	0.00	0.09	0.00	0.21	0.00
38	TATASTEEL	0.06	0.00	0.12	0.00	0.06	0.00	0.09	0.00
39	MARUTI	0.11	0.00	0.15	0.00	0.09	0.00	0.14	0.00
40	WIPRO	0.09	0.00	0.13	0.00	0.08	0.00	0.12	0.00
41	M&M	0.08	0.00	0.17	0.00	0.08	0.00	0.19	0.00
42	ITC	0.18	0.00	0.33	0.00	0.12	0.00	0.33	0.00
43	CIPLA	0.10	0.00	0.19	0.00	0.11	0.00	0.19	0.00
44	RANBAXY	0.09	0.00	0.35	0.00	0.45	0.00	0.51	0.00
45	SUNPHARMA	0.09	0.00	0.41	0.00	0.10	0.00	0.41	0.00
46	AXISBANK	0.06	0.00	0.20	0.00	0.06	0.00	0.15	0.00
47	SBIN	0.06	0.00	0.11	0.00	0.07	0.00	0.10	0.00
48	ASIANPAINT	0.16	0.00	0.44	0.00	0.26	0.00	0.41	0.00
49	POWERGRID	0.14	0.00	0.21	0.00	0.12	0.00	0.21	0.00
50	TATAPOWER	0.08	0.00	0.16	0.00	0.08	0.00	0.16	0.00

Table 4.3: Goodness of Fit

From Table (4.3) we can see that none of the 4 approaches results in a good fit. And the P-values shows that we are far away from anything that is possible to optimize the process to have a better fit. This indicates the difficulty of applying the Hawkes process directly to raw market data.

Figure (4.19) shows sample Q-Q plots of residuals of the estimated Hawkes Process for stocks ACC, RELIANCE, WIPRO and TATAMOTORS listed in NSE on day December 17, 2012. From the plots we can see that for each stock, the distribution of the residuals of the estimated Hawkes process does not match the theoretical distribution (an exponential distribution with rate 1) well.

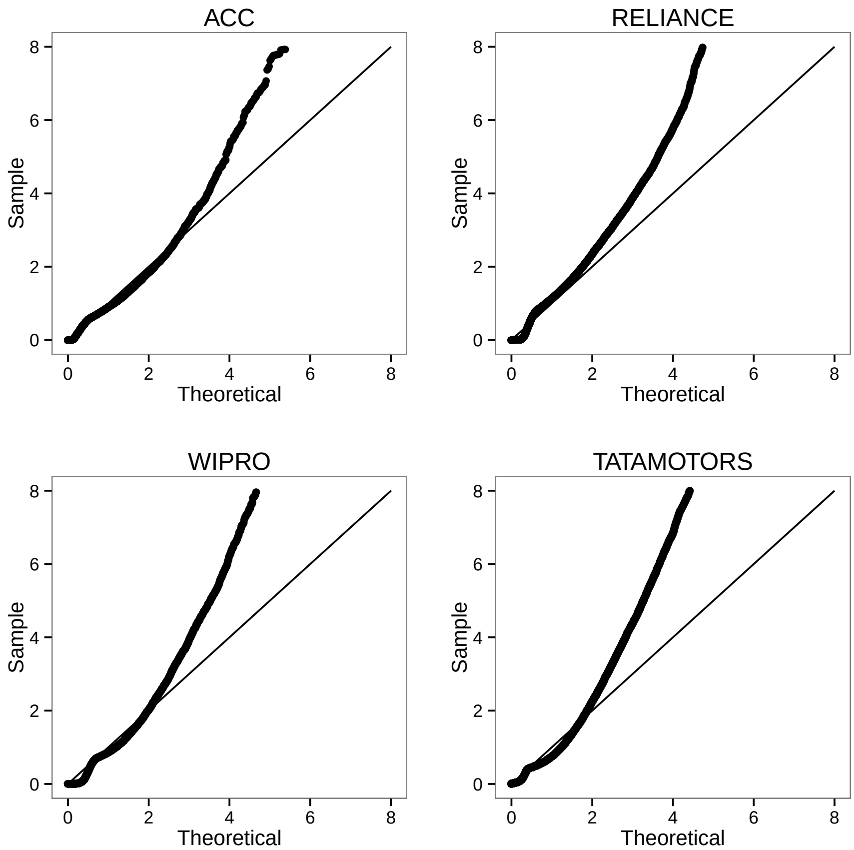


FIGURE 4.19: Q-Q plots of Sample Stocks Residuals

4.4 Conclusion

In this chapter I have modeled the liquidity market data in a microstructure level. I have applied the Hawkes process to model the self-exciting feature in the liquidity market data. I have verified with the non-parametric kernel estimation method that the NSE data does present a self-exciting feature.

I started by introducing the theoretical ground of the Hawkes process. I gave exploration of how to calibrate data with both parametric approach using maximum likelihood estimation, and non-parametric approach by solving a Wiener-Hopf equation system with the Nyström method.

I have tried to fit the Hawkes process to the market data from the NSE. I have shown that the shape of the estimated kernel from the data with the non-parametric kernel estimation method does have a decaying trend. This indicates that the data does have a self-exciting feature.

However I have found that fitting the Hawkes process to the data with parametric kernel estimation method does not give a good result. The results from the Kolmogorov–Smirnov test reject the hypothesis that the model fits the data. Future works it to investigate the cause of the failure and develop a better model to capture the features in the data.

Chapter 5

Predicting Market Impact for Big Trades

5.1 Introduction

In this chapter, I address the research question about predicting the liquidity impact of big trade events. At the microstructure level where we can observe individual trade events, understanding the relationship between these events is very important for making trading strategies. The events have different properties against each other, they have different timestamps, different traded prices and sizes, etc. Such details can reveal information to the market participants and thus have impact to the market. In this chapter, I try to study the liquidity impact of big trade events and develop algorithms to predict the impact.

From Chapter (4) I have shown that the Hawkes process has already been used in microstructure modeling in trading area. However most of previous work focused on the qualitative aspects. For example, Hardiman et al. (2013) used the branching ratio to measure the reflexivity (endogeneity) of the market, Carlsson et al. (2007) used absolute intensity as trading signals to make up trading strategies. However, it is not straightforward how can one use such models as a quantitative tool in terms of predicting the occurrence time of future events. Indeed, as we have seen in Chapter 4, it is also very difficult to fit real market data to a Hawkes model.

Even if we could fit the data to the model well, we find it is not easy to leverage the model to make predictions. I have developed a prediction algorithm in Chapter (4) using the fact that the residuals process of the Hawkes process is a Poisson process with rate 1, which enables us to make a conditional prediction of the occurrence time of the next event given the events history. But this algorithm can only make one step ahead prediction, which is just a Poisson process that is unpredictable in nature, so that prediction cannot be an exact number for the occurrence time but a range with a confidence level. In short, we will not be using the Hawkes model to make up a prediction strategy for a trader's sake.

However, applying the non-parametric kernel estimation on the data shows that there is a clear self-exciting feature in it. The difficulty of fitting the Hawkes process to the data nicely may come from the following points:

- Too much noise in the trades data - There are a lot of trades with different sizes co-existing in the data. The sizes can vary from 1 to 1000. One can imagine that some such trades may come from pure randomness and not necessarily contribute to the excitement, simply consolidating the trades into fixed sizes like we did in Chapter 4 may not eliminate such randomness.

- The timestamps of the trades were published by the exchange, which may not be the exact time of the actual events, because the exchange bundles the events when publishing. The exchange protocol may bundle multiple updates of multiple instruments within a single message Filimonov and Sornette (2013). To cope with this situation one has to decide some method to randomize the timestamps so that they are different when fitting to the model. This process may increase the fitting error.

- External excitations may also contribute to the volatility, these are not included in the simple univariate Hawkes process model.

There are possibilities that one can have systematic fitting failure for Hawkes processes. Filimonov and Sornette (2013) outlined a couple of such scenarios. For example, a few outliers in the data can cause strong upward biases on the estimation of branching ratio n which is based on the raw model parameters. The complicated calculation needed for maximizing the likelihood function can end up at local optima rather the global one. And in some cases one can confuse the estimation process by manually making up synthetic data by mixing up pure Poisson process with changes of regime. All suggest that the model has weaknesses in practice.

However, the idea that the financial market data could be self-exciting at some stage is intuitively attractive. Therefore, we try to study this feature from a statistical view in this chapter. In particular, we try to model the impact of a big trade event in the near future.

In this chapter, we propose four algorithms to predict the liquidity impact by big trade events. We give definitions for big trade events and measurement of their impact. We present empirical analysis with respect to the big trade events on real market data. Then we apply the four algorithms on the data to verify the effectiveness and evaluate their performance. We show that all the four algorithms give reasonable results while the one with a spread-based classification outperforms the others.

This chapter is organized as following. Section (5.2) provides the definitions and describes the four algorithms we propose. Section (5.3) shows empirical analysis and findings in the market data, which confirms that the big trade events do have impact on liquidity. Section (5.4) outlines the performance of the four algorithms. Section (5.5) concludes and discusses future directions of this research.

5.2 Modeling

Before we start to build the model, we need to illustrate a general background of the study that we are going to perform in this chapter. In a limit order book market, the intra-day executions of an instrument are a series of events $(t_i, p_i, q_i)_{i\in 1,2,...,N}$, where t_i is the timestamp measured with high precision (often at the milliseconds level or below), p_i is the price of the execution and q_i represents the quantity, N is the total number of executions in the target day. This simple context, however, has several issues lying in the technical details that need to be checked carefully.

In reality, $(t_i)_{i\in 1,2,...,N}$ is irregularly spaced in a semi-random manner, it could have some autoregressive or self-exciting nature but is likely to not stay in a simple state, as we have analyzed in Chapter 4. More noteworthy, depending on the method that the exchange uses to generate the timestamp, it is not necessary that $(t_i)_{i\in 1,2,...,N}$ is monotonically increasing. That said, it is possible to observe that $t_i = t_{i+1} = t_{i+2} = ...$ in the raw market data, which could suggest that they are all parts of a block trade that crosses through multiple levels of an order book. This type of event by the nature of having multiple events with the same timestamp can make the modeling work unnecessarily complicated, thus we consolidate such executions into one that sums up the quantities when preprocessing the data. By doing this we end up having series that have monotonically increasing timestamps.

Intuitively the price p_i must have some impact on the other two attributes. The series $(p_i)_{i\in 1,2,...,N}$ itself is a random process that has been the subject of enormous study in the literature, while it is still open ended in terms of determinism. To simplify the problem and focus on the liquidity side, it was decided not to use the price information in the analysis.

The quantity q_i tends to have an exponential-like intra-day distribution. The distributions between days could be different, in terms of both the mean and the standard deviation. So the first thing we would like to check empirically is the distribution shape, which we will see in Section 5.3. This random nature raises the question of how do we define big trades in terms of execution quantity. To measure the impact of such big trades on liquidity, we need to give definitions for what is the "impact" and over what time range do we expect to measure it.

5.2.1 Definitions

In a consolidated, monotonically increasing timestamped series $(t_i, q_i)_{i\in 1,2,...,N}$, q_i is a random variable that follows some exponential-like distribution - the smaller the size is, the more likely it is observed in the data. Because of this randomness, there is no single number that naturally defines a big quantity that distinguishes the trade from others. Toke and Fabrizio (2012) studies a trade-through event, which corresponds to a trade that crosses several levels of the order book. Although such events have an impact on liquidity in the sense that the gap between the top of the order book is expected to be filled soon by following trades, they are not necessarily big and the time needed to fill the gap is typically short. We are more interested in events that have a bigger impact that can last at the level of minutes. Therefore, we introduce the definition of big trades as:

Definition 5.2.1. Big Trade: an execution event $(t_k, \dot{q}_k)$ that $\dot{q}_k \geq Q(\{q_i\})$, where Q denotes a function that gives a specified quantile of the input data.

In practice, we need to determine the value of $Q(\{q_i\})$ before market open, we use the executions that happened in rolling historical days to build $\{q_i\}$.

With a big trade $(t_k, \dot{q}_k)$ observed, we are interested in the impact it makes on the liquidity in the following time range T. Since the total volume of each day for an instrument is a random variable, we do not have an absolute liquidity measure as a baseline. Thus we define the response to a big trade as relative:

Definition 5.2.2. Big Trade Response for T: the ratio r of the liquidity rate s_T of T to the rate S_W of a time window W just before the big trade.

We define the rate as the traded volume per unit time, i.e. $s_T = \frac{\sum\limits_{t \in T} q_t}{T}$, so it provides an overview of the liquidity in the specified time. Similarly, we define the rate of a time window W as $s_W = \frac{\sum\limits_{t \in T} q_t}{W}$.

If there is no big trade occur during time T, r is expected to be close to 1 when T is relatively small compared to the time window W, while the difference is expected to come from the random nature of the execution event series. When there is a big trade, we expect a stimulation to liquidity. The goodness of this definition of r is that it is a relative measure that does not depend on the absolute value of the liquidity level, which can be random and different across days.

With a single T it is difficult to see how the impact would evolve over time, so we extend the response definition further as a function:

Definition 5.2.3. Big Trade Response function: $R(t) = \frac{s(t)}{S_W}$, where t is a time point after the time of the big trade t_k, and $s(t)$ is a function that gives the liquidity rate of the time range $(t_k, t]$, i.e. $s(t) = \frac{\sum\limits_{t=t_k}^{t} q_t}{t - t_k}$.

The target of the prediction of the market impact of big trades is to have an estimation of the shape of $R(t)$, so that one trading strategy can make adjustments accordingly when such events are observed in real time.

5.2.2 Prediction Algorithm

We propose four algorithms for predicting the response function $R(t)$, one based on statistical average, one using the non-parametric kernel estimation of Hawkes process, one using a mixture of ARMA models, and one using a spread based classification. We will explain the details of each of the algorithms in this section, and review the performance in Section 5.4.

Static Algorithm

We call the first algorithm Static, meaning it simply builds the response function using an empirical estimation based on the historical data. We start building this estimation from one day data.

The first issue in estimating the response function $R(t)$ is that we only have discrete data. To solve this we sample the time range for the response function to have n points, estimate the function value at each points, then make a polynomial interpolation of the values to approximate $R(t)$. For a big trade $(t_k, \dot{q}_k)$, we sample the function values within the range $(t_k, t_k + T]$ to have m points. The estimation algorithm is as following:

1. Having trades data for an historical day as $(t_i, q_i)_{i \in 1,\dots,N}$, we first select the big trades $(t_k, \dot{q}_k)$ that $\dot{q}_k \geq Q(\{q_i\})$.

2. For each big trade $(t_k, \dot{q}_k)$, look back at the events before t_k, take n windows $\{w_i\}_{i \in 1,\dots,n}$ whose length equals to L. For each window w_i, sum up the total quantity of the executions as Q_i^w. Make an average of the n Q_i^w and divide by L to get the rate $S_w = \frac{\sum_i Q_i^w}{L}$. Note for each historical day, we only start counting big trade events after time $n \times L$ so that we always have enough time to build the n windows. This forces us to ignore the big trade events happened early in the day before time $n \times L$, this is fine as long as $n \times L$ is sufficiently less than the total trading time in a day and the historical data set is large. We ensure both in our empirical analysis.

3. Starting from t_k, collect the executions up until each time out of a series $\{t_i\}$, where each t_i represents a sampling point within the response time T. Summing up the executions for each t_i and dividing by $t_i - t_k$ gives the rate series $\{s_i\}$, which is a sample from the function $s(t)$.

4. Dividing $\{s_i\}$ by S_w results in a sample of the response function $R(t)$ as $\{R_i^{t_i}\}$.

5. Finally we build the response function $R(t)$ as a polynomial interpolation of $\{R_i^{t_i}\}$.

Hawkes Algorithm

The second algorithm, which we call Hawkes, is based on the the Hawkes process that we describe in Chapter 4. Note although we have shown in Chapter 4 that the Hawkes process with an exponential kernel does not fit the data well, it does not prevent us from using the Hawkes process as part of a predictor for the response function here. Because the Hawkes process models the self-exciting feature, we can use it as a simulator and make prediction based on the simulated data. Recall that the arrival intensity λ_t of the Hawkes process is conditional on a filtration $\mathcal{F}_t$ as defined by 5.1:

$$\lambda(t) = \mu(t) + \int_0^t \phi(t-s)\, \mathrm{dN}_s \tag{5.1}$$

where $\mu(t)$ represents the base intensity, $\phi(t-s)$ is the kernel function that models how the past events affect the intensity of arrival in the future. Having the kernel function $\phi(t-s)$ in a specific form such as exponential or power law, one can estimate the parameters of the kernel function through approaches like maximizing likelihood function Ogata (1978). One can also use the non-parametric kernel estimation method proposed by Bacry and Muzy (2014) to get the shape of the kernel function without set a specific form.

In Chapter 4, we confirm that the NSE data does have self-exciting feature by using the non-parametric kernel estimation method, which is inline with the idea that the big trade events can have impact on liquidity. We also find that the parametric method does not fit the data well. However, to measure and predict the impact of the big trade events, we have to use a parametric way to quantify the impact. And the goal of this the research question in this chapter is not to use the model to qualify features in the data, but to build a predictor that output the response time for the events. Taking these two points into account, we choose the parametric kernel estimation technique to use in this algorithm.

With the estimated kernel function one can then calculate the conditional intensity through Equation 5.1. As per definition the intensity is the expected number of events in unit time, what we really care about is to be able to predict the occurrence time of the future events, so we need to transform the conditional intensity to time space. As proposed in the previous chapter, this can be done by converting the process from time space to the residual space $\{\tau_i\}$:

$$\tau_i = \int_{t_i}^{t_{i+1}} \lambda(t)\mathrm{d}t \tag{5.2}$$

Also, knowing $\{\tau_i\}$ has the distribution of a stationary Poisson process of intensity 1 Papangelou (1972), we are able to predict the probability distribution of the one-step ahead time of the next event conditional on the past events. However, this is still very limited in the sense that the duration between two events is generally short, being able to forecast only one-step ahead does not help a trader much to adjust strategy.

We thus propose a method that is based on conditional average intensity over a given period. When we observe a big trade, we measure the intensity around the time it happened, and use the Hawkes model to calculate how long the intensity would take to fall back to the normal level. We do this via a Monte Carlo approach, i.e. simulating a Hawkes process with the same parameter and calculating the decay time needed for the specified intensity level statistically. To make it computationally efficient, we do this simulation exercise in advance of the testing process, and store a map between several discrete high intensity levels and the corresponding decay time.

We describe the algorithm in two parts. The initialization phase is for building a model which fits the Hawkes model with the training data set and build the intensity decay map. The testing phase is for running the algorithm on the test data set.

For the initialization phase, we do the following for each stock:

1. Transform the data from event time to volume time. The original data we have has every trade execution in a linear timestamped series form. Each execution has its price, quantity and timestamp. We consolidate the executions into a series of events with fixed quantity size Q.

2. Detrend the volume time with the intraday seasonality curve. We build a moving average volume distribution curve with L look-back days of historical data. We use this curve to detrend the transformed volume time we get from the last step, to remove the typical U shape seasonality of intraday volume distribution.

3. Fit a univariate Hawkes model with the preprocessed data. We use maximum-likelihood method to find the best parameters of the model. We get a set of parameters for each training day.

4. By the end of the training period, we take a simple average of all the daily parameter sets as the estimated parameters of the Hawkes model.

5. Build the intensity decay map. We first simulate the estimated Hawkes model for a time period with length T, find the maximum intensity M during the simulation period, then make a sequence $\{m_i\}$ ranging from the unconditional intensity $\mu(t)$ to M, with step s. For each number m_i in the sequence, we find all the cases in the data that the intensities reach m_i, for each case we measure how long does it take to get back to the unconditional intensity $\mu(t)$, then take the average time needed for all the cases as $\bar{t}_i$. Finally we make up the map with key/value pairs $\{(m_i, \bar{t}_i)\}$.

For the testing phase, we do the following for each stock:

1. When we observe a big trade event $(t_k, \dot{q}_k)$, we measure the intensity m_{t_k} at time t_k as $m_{t_k} = \frac{\dot{q}_k}{t_k - t_{k-1}}$. Then we find the closest intensity number in the pre-built map $\{(m_i, \bar{t}_i)\}$, the corresponding decaying time $\bar{t}_i$ is then the predicted response time for this big trade event.

2. By the end of each testing day, we update the Hawkes model and intensity map with the new data.

This Hawkes based algorithm provides a way to model the response time with intensity measure, which would make even more sense when there are more algorithmic trading activities, so that the trade events can present more reflexivity. However from what we see in the goodness of fit work from the last chapter, we do not have a good fit of the model on the data. This would of course limit the performance of this algorithm, so we only see it as a reference solution.

Mixture of ARMA Algorithm

Finally, we propose an algorithm mixing several ARMA (Autoregressive moving average) models. ARMA is a commonly used time series model, with which one can predict the future from historical time series data.

Given a time series of data $\{t_i, x_i\}$ where t_i is an integer index, and x_i is a real number, an $ARMA(p, q)$ model Mills (1991) is given by :

$$x_i = c + \epsilon_i + \sum_{k=1}^{p} \varphi_k x_{i-k} + \sum_{k=1}^{q} \theta_k \epsilon_{i-k} \tag{5.3}$$

where p is the number of autoregressive terms and q is the number of moving-average terms, c is a constant, φ_k and θ_k are the parameters for autoregressive and moving-average model respectively, ϵ_i are the error terms.

In our case, we first need to convert the executions data which by its origin has irregularly spaced timestamps to a time series. We do this by downsampling the data into M fixed-length bins, so that the executions series $(t_i, q_i)_{i \in 1,\dots,N}$ is converted to a series $(t_j, Q_j)_{j \in 1,\dots,M}$, where t_j is a regularly spaced timestamp, Q_j is the accumulated quantity of executions happened in range $(t_{j-1}, t_j]$ (with $t_0 = 0$).

Considering the fact that the ARMA model is weak at making long term predictions, we only forecast one-step ahead of the time that we observe the big trade. The next question is how do we forecast the shape of the function $R(t)$? The answer is to do it in a similar manner to the Static algorithm. We use K ARMA models $\{a_{i \in \{1,\dots,K\}}\}$ that a_i has sampling interval i seconds. We forecast with each of them and then make a polynomial interpolation of the prediction points.

Denote the variables of the K models as a vector of points X_i:

$$X_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,K} \end{pmatrix} \tag{5.4}$$

So we have the prediction in a matrix form:

$$X_i = C + E_i + \sum_{k=1}^{p} \Phi_k X_{i-k} + \sum_{k=1}^{q} \Theta_k E_{i-k} \tag{5.5}$$

with C, E_i, Φ_k and Θ_k as the vector replacements of c, ϵ_i, φ_k and θ_k, respectively.

Spread Algorithm

If we look into the market microstructure, liquidity depends on some properties such as spread, order book depth, volatility, imbalance at top of order book, etc. One can come up with a model that takes all these properties into account, and fit historical data to estimate the model. With a concern for computational efficiency, we propose a model that only involves spread.

The rationale of our proposal is that spread is a good proxy of order book resilience, which indicates how fast the order book will recover from a big trade that wipes through several levels of the order book. The spread right before a big trade thus is a good indicator of how long will it take for the order book to react to the big trade and may imply the execution rate within the time.

For the sake of implementation, we take this as a simple add-on to the Static algorithm. We estimate a list of response functions for G spread levels from historical data. Then, when we observe a big trade event, we look at the current spread value, and use the corresponding pre-estimated response function to forecast the response.

The estimation process is based on the one for the Static algorithm, but in addition we have a map with spread level as key and response function as value. The algorithm process is as follows:

1. Setup a map $\mathcal{M}$ to store the spread for response function pair.

2. Having trades data for an historical day as $(t_i, q_i)_{i\in 1,\dots,N}$, we first select the big trades $(t_k, \dot{q}_k, \tau_k)$ that $\dot{q}_k \geq Q(\{q_i\})$, where τ_k denotes the spread value.

3. For each big trade $(t_k, \dot{q}_k, \tau_k)$, look back at the events before t_k, take n windows $\{w_i\}_{i\in 1,\dots,n}$ whose length equals to L. For each window w_i, sum up the total quantity of the executions as Q_i^w. Make an average of the n Q_i^w and divide by L to get the rate $S_w = \frac{\sum_i Q_i^w}{L}$.

4. Starting from t_k, collect the executions up until each time from a series $\{t_i\}$, where each t_i represents a sampling point within the response time T. Summing up the executions for each t_i and dividing by $t_i - t_k$ gives the rate series $\{s_i\}$, which is a sample from the function $s(t)$.

5. Dividing $\{s_i\}$ by S_w results in a sample of the response function $R(t)$ as $\{R_i^{t_i}\}$.

6. Build the response function $R(t)$ as a polynomial interpolation of $\{R_i^{t_i}\}$.

7. Keep $R(t)$ in $\mathcal{M}$ with key τ_k.

In the testing phase, when we observe a big trade event $(t_k, \dot{q}_k, \tau_k)$, and query $\mathcal{M}$ with τ_k to get response function $R(t)_{\tau_k}$. One thing to note is that in real data, spread can be arbitrarily valued. Sometimes this results in some values as outliers that have only few data points to build $R(t)$. Or one could see spread values that were not observed in historical data, which leads to no information for $R(t)$. To solve these issues, we set G at a level that for any $\tau_k \leq G$, there are enough data points to build $R(t)$ in historical data. For data points with $\tau_k > G$, we merge them with points $\tau_k = G$.

5.2.3 Evaluation Method

The response function by itself is a continuous function. To evaluate the prediction of such a function, we need a way to quantify the goodness of the prediction. For that purpose, I define a performance indicator for a predicted response function in this section.

From a trading strategy perspective, the most important output of the response function is the time needed for the liquidity rate falls back to the level before the big trade event arrived. I make the performance indicator based on the comparison between the real time needed for the fall and the value output by the predicted function.

Denote $\dot{t}_{real}$ as the real time when the liquidity rate falls back for a big trade $(t_k, \dot{q}_k)$, we have:

$$s(\dot{t}_{real}) = S_W \tag{5.6}$$

In contrast, the predicted time $\dot{t}_{pred}$ must meet the equation:

$$R(\dot{t}_{pred}) = 1 \tag{5.7}$$

Applying the predicted function R to Equation (5.7) and solving the resulted equation give the value of the predicted time $\dot{t}_{pred}$.

With the measured real time $\dot{t}_{real}$ and the solved predicted time $\dot{t}_{pred}$, I define the performance indicator P at time (t_k) as the prediction error:

$$P = log(\frac{\dot{t}_{pred} - t_k}{\dot{t}_{real} - t_k}) \tag{5.8}$$

For a specific stock, there are a bunch of big trades for any trading day, which map to the same number of performance results for that day. We take the mean absolute percentage error, 95% quantile and the standard deviation of all the performance results over all the trading days as the benchmarking set for the algorithms.

5.3 Empirical Analysis

Our analysis is based on the market data recorded from the NSE with range from January 2, 2012 to April 8, 2013, which has 301 days in total. The data for each day is a linear time series of 4 kinds of order book events, namely new order, amend order, execution and cancel order. Each event is detailed with a timestamp in millisecond precision and information like price, quantity, etc. In this chapter, we focus on the trades, which map to the execution events.

5.3.1 Basic Statistics

Table (5.1) shows the statistics drawn from the data. The first column gives the average daily volume for each instrument. $\bar{q}$ and $\bar{N}$ represent the average size and the number of executions over the whole data set, respectively. $q_{99\%}$ and $\bar{N}_{99\%}$ are the key quantities for identifying the big trades in our analysis, which map to the 99% quantile size of the all executions for each instrument, and the number of executions whose size are no less than this quantile. The last column, $\bar{s}(\%)$, is the percentage that the total volume of the big trades are of the total volume of all executions.

	Symbol	$\bar{V}$	$\bar{q}$	$q_{99\%}$	$\bar{N}$	$\bar{N}_{99\%}$	$\bar{s}(\%)$
1	ASIANPAINT	62523.60	11.26	52.00	5423.20	187.40	42.82
2	GRASIM	62786.40	11.09	59.40	5490.20	210.20	41.96
3	SIEMENS	131793.80	18.15	126.20	7246.20	130.60	22.56
4	ULTRACEMCO	145134.00	27.25	83.80	5380.00	222.40	50.08
5	ACC	255955.80	20.29	107.60	13530.60	327.40	29.43
6	DRREDDY	296023.00	24.07	126.20	12762.40	249.40	41.28
7	BANKBARODA	323663.60	33.61	217.40	9683.40	262.80	30.27
8	PNB	387539.40	28.56	182.40	13347.40	332.80	24.20
9	MARUTI	451697.80	19.07	120.80	24169.00	497.60	23.80
10	BPCL	502686.20	27.82	190.60	18638.80	364.00	29.49
11	HEROMOTOCO	526557.80	18.64	125.00	28469.20	476.60	19.32
12	SUNPHARMA	630443.60	25.11	200.80	24709.40	439.20	32.73
13	BAJAJ-AUTO	637511.40	17.10	139.60	37446.60	534.00	18.38
14	KOTAKBANK	695176.40	40.01	244.60	17149.20	339.40	29.09
15	HCLTECH	1033019.00	42.81	318.80	23535.20	427.00	23.37
16	RANBAXY	1046025.80	44.56	264.80	23278.00	423.20	42.94
17	LUPIN	1090506.20	22.12	164.40	53418.80	929.00	28.66
18	CIPLA	1242221.00	51.24	442.80	25291.40	480.60	27.59
19	GAIL	1381669.60	42.91	403.20	36163.00	549.60	31.21
20	WIPRO	1408333.60	51.42	351.60	26269.40	662.80	35.44
21	AMBUJACEM	1438243.60	95.51	696.60	15701.20	287.60	26.81
22	HINDUNILVR	1784827.20	48.17	428.00	37891.60	632.40	24.95
23	CAIRN	1805503.00	54.37	454.40	33865.20	680.20	31.23
24	M&M	1806344.20	36.77	304.60	48939.40	847.00	19.96
25	ONGC	1865860.20	51.92	537.80	35368.60	587.80	23.09
26	HDFCBANK	1962802.60	51.91	465.00	37931.60	596.60	19.19
27	JINDALSTEL	2121696.60	54.82	450.00	37127.20	927.60	25.55
28	LT	2200917.60	26.90	224.80	81728.40	1231.40	19.36
29	POWERGRID	2226197.00	133.90	1147.00	17211.40	294.20	30.81
30	HDFC	2302705.20	46.50	446.40	50409.00	807.40	23.54
31	RELINFRA	2326729.40	47.16	399.20	48550.60	831.60	18.20
32	INFY	2425315.40	20.05	129.40	124219.80	1861.20	21.34
33	TCS	2428709.20	31.61	239.00	76177.60	1351.60	25.38
34	SBIN	2484716.60	23.18	158.60	107704.40	1478.40	19.28
35	AXISBANK	2552638.20	32.38	250.00	79306.60	1253.80	17.77
36	BHEL	3129585.20	55.97	524.80	55969.60	707.80	19.99
37	SESAGOA	3397875.60	83.19	652.60	40906.40	670.40	20.15
38	TATAPOWER	3824317.80	188.43	1058.00	21134.80	478.00	36.00
39	NTPC	4137708.60	162.52	1553.20	25066.20	385.40	30.41
40	RELIANCE	4293211.60	40.31	287.80	106484.60	1670.60	21.08

41	ICICIBANK	4534299.80	46.38	312.40	97080.80	1504.60	18.96
42	COALINDIA	4662075.40	77.98	524.00	59870.60	1044.80	36.36
43	ITC	5204424.20	114.89	888.60	44902.60	1039.00	34.96
44	TATASTEEL	6334154.00	55.19	494.00	111828.20	1760.20	16.87
45	BHARTIARTL	6858319.80	76.29	782.40	89823.20	1100.60	23.18
46	HINDALCO	10507460.20	149.06	1217.40	70102.20	1267.20	21.62
47	IDFC	10557410.00	163.04	1431.80	64096.00	938.60	18.97
48	TATAMOTORS	13941216.20	104.41	1000.00	135389.40	2251.80	27.88
49	DLF	18727914.20	158.33	957.40	109897.80	3021.00	35.42
50	JPASSOCIAT	19925042.80	320.44	2563.00	61617.80	1325.40	26.53

TABLE 5.1: Statistics of Big Trades

Obviously, we see that such big trades take up a big chunk of the whole, with approximately 10% in number of all executions that take about 20% to 40% of the volume. This clearly indicates that the big trades can have a big impact on the trading activity.

We also get several other views from the simple statistics. They show a general trend that, with a few exceptions, the larger the average daily volume is, the larger the average trade size and number are. We plot this trend in Figure 5.1. The figure shows 4 plots, each of them outlines the trend of one of the 4 columns of Table 5.1 ordered by the average daily volume size of the individual stocks. The 4 columns are $\bar{q}$, $q_{99\%}$, $\bar{N}$, $\bar{N}_{99\%}$, respectively. From this observation it seems that the impact of big trades among the stocks could be similar, thus we do not try to split the stocks into different profiles in this chapter. Indeed that there are a few exceptions that suggest it would make more sense to have some policy for grouping similar stocks together, I leave this as future work.

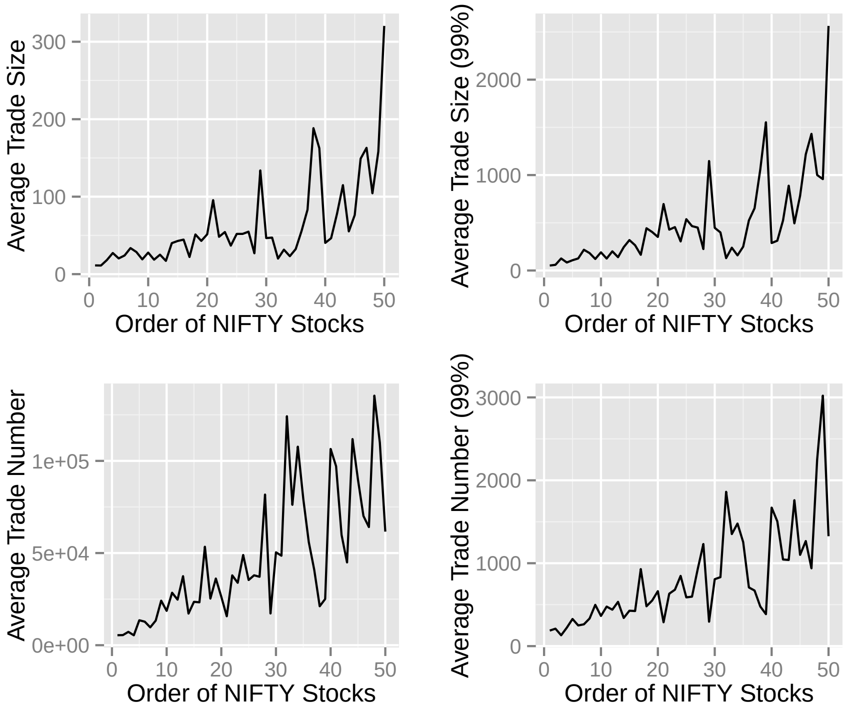


Figure 5.1: Trends in Volume Statistics

Figure 5.2 plots the trend for the percentage of big trades in total daily volume for the NIFTY stocks. Here we do not see the increasing trend as in Figure 5.1, which confirms the intuition that the distribution of big trades does not depend on the daily volume for each stock.

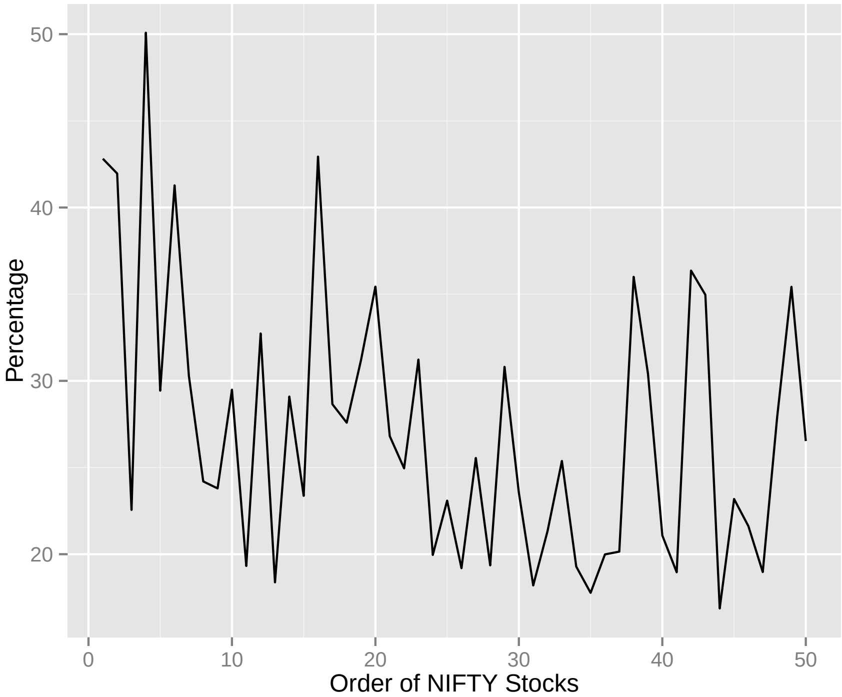


FIGURE 5.2: Big Trades Percentage in Total Daily Volume

For each stock, we also would like to know how the trade sizes are distributed. We arbitrarily pick 4 stocks and plot their data for the day January 3, 2012 in Figure 5.3. Each sub-figure represents one stock with its symbol as the title. The curves outline for each quantile level of the trade size how much the percentage of the volume traded below that size takes of the total daily volume. One can note that there is a persistent pattern that the bigger trades tend to take a greater percentage of the total volume, e.g. less than 50% percent of the daily volume were traded under the 80% quantile size, for all 4 stocks.

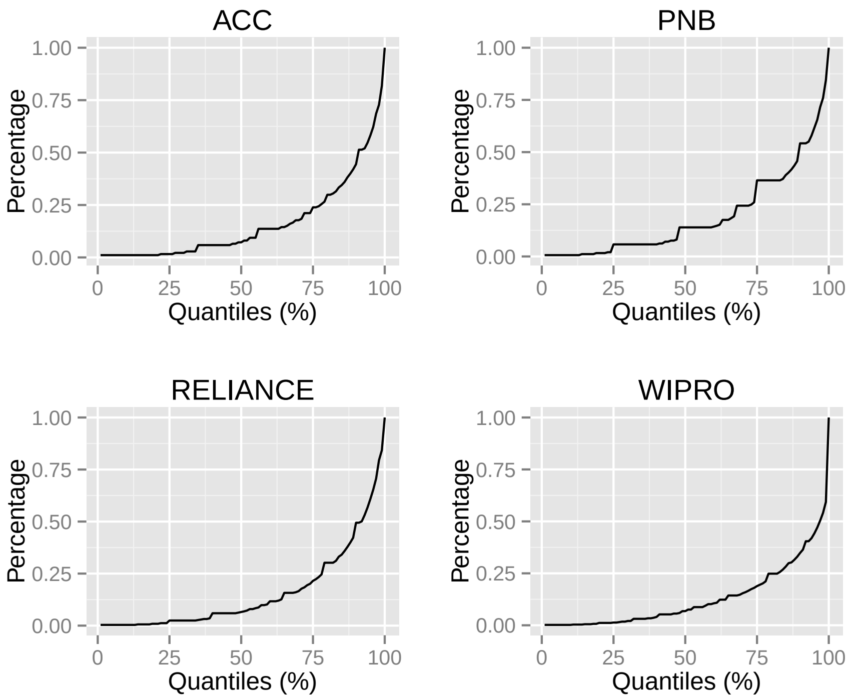


FIGURE 5.3: Percentage Of Each Quantile Level

5.3.2 Response Time Statistics

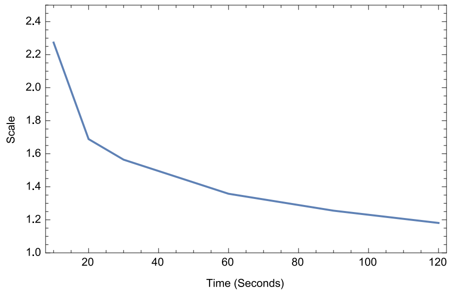
In this section I present the results from a simple statistical study of the response of the big trades $(t_k, \dot{q}_k)$ that were defined in Section 5.2.1. The purpose is to visualize the impact of such big trades on market liquidity. One would expect to see some stimulation effect right after the events, which may last for a while and then hopefully decay to the level before the trades arrived. The results in this section show how this effect actually looks.

To visualize this effect on real data for different stocks, I pick up 4 stocks with different liquidity levels and take the day March 1, 2012 as a sample. The 4 stocks are GRASIM, ACC, RELIANCE, DLF, ordered by their total daily volumes. The information extracted from the data is outlined in Figure 5.4, Figure 5.5 and Figure 5.6.

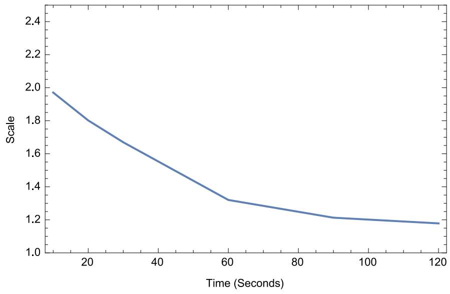
Figure 5.4 shows the response function extended using algorithm Static. I set the window size W to 1 minute, number of look-back windows L to 2, the time T for measuring response to 2 minutes, and the number of break points for $(t_k, t_k + T]$ to 6. The actual break points are set at 10-second, 20-second, 30-second, 1-minute, 1.5-minute and 2-minute. For each point, I measure the execution rate s_i and divide it by the average execution rate S_w, to get the scale as the value of the response function $R(t)$ at that point. For the response function value at each point we make a polynomial interpolation to approximate $R(t)$.

From the plots, we see clear stimulation and decay effects for all 4 stocks. This confirms the intuition that a big trade would bring short term volume increase and the effect would not last forever. It is also interesting that none of the 4 stocks statistically go back to the

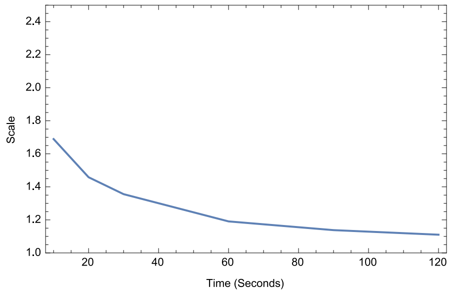
1-minute average execution rate in 2 minutes after the big trades. They present different decay speeds, this makes sense and suggests that we build individual profiles for each stock.



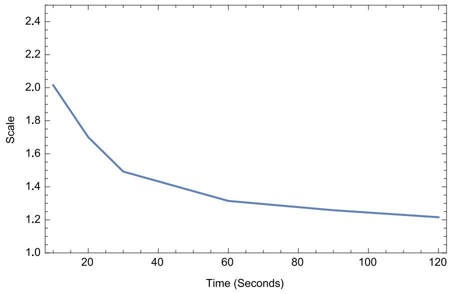
(a) GRASIM



(b) ACC



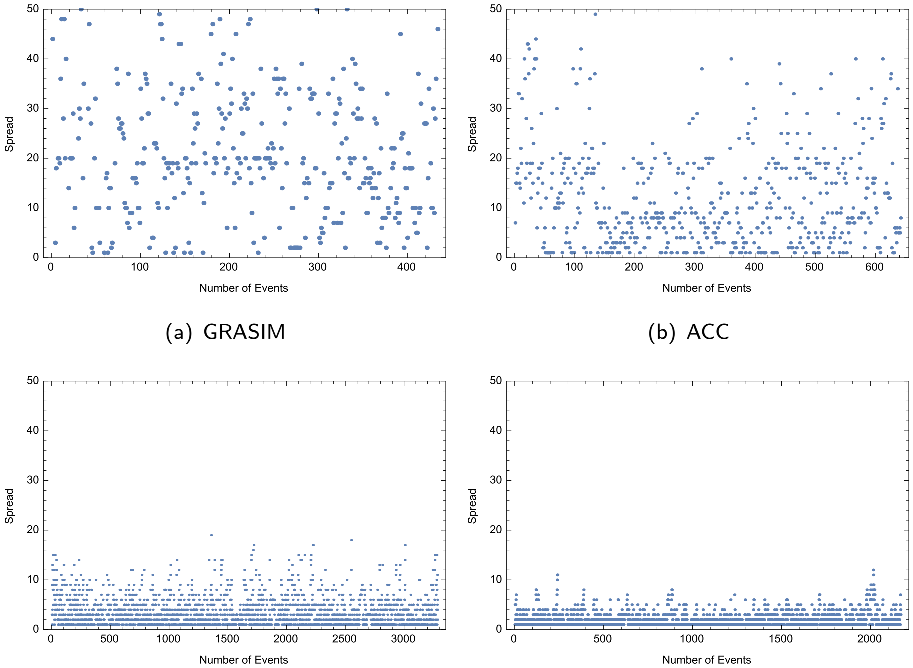
(c) RELIANCE



(d) DLF

FIGURE 5.4: Response of Big Trades on Volume

Figure (5.5) presents the spread distributions for all the big trades found in the data. The x-axis represents the number of big trade events, for example in graph (a) we see there were more than 400 big trade events occurred for stock GRASIM, while in graph (d) we see there were more than 2000 big trade events for stock DLF. The y-axis represents the spread measured as the number of ticks at the time of each big trade events occurred. From the graphs we first see that the more liquid stocks (RELIANCE and DLF) have narrower range of spread for the big trade events than the less liquid stocks (GRASIM and ACC). This is as expected because the more liquid the stock is, the lower the spread is. Then we can observe that for all the four stocks, the big trade events have different level of spread. That means some big trade events happened when the spread was small (e.g. 5), while some big trade events happened when the spread was big (e.g. 10). Because spread is a part of transaction cost, higher spread can reduce the willing of traders to trade and thus make impact to the liquidity too. Therefore, I try to classify the big trade events with the spread level at the time they occur, to reduce the interference from the spread.

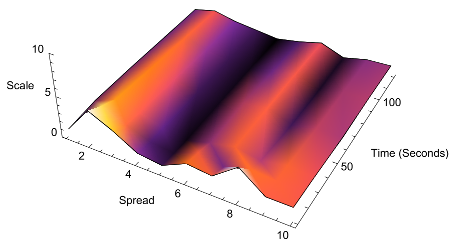


(a) GRASIM (b) ACC

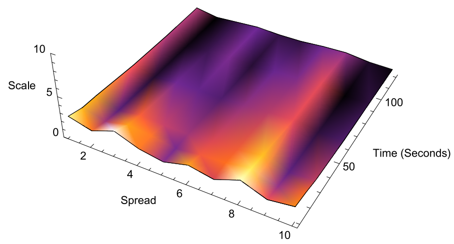
(c) RELIANCE (d) DLF

FIGURE 5.5: Spread before Big Trades

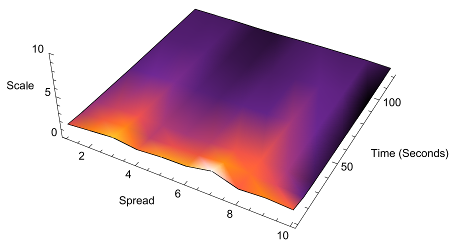
Figure (5.6) presents the response function of big trade events at different spread levels. The axis marked as "Spread" represents the level of spread. For each level, the axis "Scale" and the axis "Time (Seconds)" compose a 2-Dimension graph as the ones shown in Figure (5.5). I restrict the spread to range $[1, 10]$ to make the graphs comparable. From the graphs we can see for each stock, the response functions at different spread levels have different shape. This finding confirms that the spread-based classification makes sense.



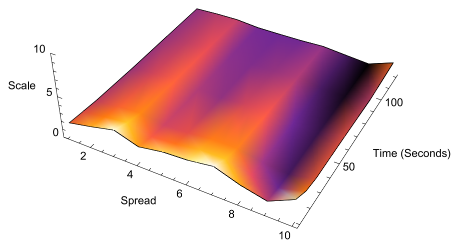
(a) GRASIM



(b) ACC



(c) RELIANCE



(d) DLF

FIGURE 5.6: Response of Big Trades joint with Spread

5.4 Performance

In this section we present the performance of the algorithms listed in Section 5.2.2. The results were obtained using the approach described in Section 5.2.3. In order to build initial statistics or model parameters for each algorithms, we split the data set into two parts, one for initialization and another for testing. The 301 days in the data set are divided into one set of 30 days and another of 271 days. The 30 days set is only used to initialize the algorithms and not included in the testing process. For the 271 days in the testing set, we get 50 performance results for the NIFTY stocks for each day. This gives us $271 \times 50 = 13550$ sample results in total for each algorithm. The final performance indicators (MAPE, 95% quantile, standard deviation) are calculated over these 13550 numbers.

Table 5.2 show the results for the four algorithms. The rows are ordered by the MAPE results from the worst to the best. From the table we see the two algorithms with sophisticated models, i.e. ARMA and Hawkes perform worse than the two with plain statistics. This could come from the difficulty to fit the models nicely with the data (as we have seen in Chapter 4). Without further investigation, these model based algorithms add very little value to the forecasting task, especially given the fact that they require more energy to design the process and calculation for calibration and estimation.

The algorithm based on a spread map performs slightly better than the static one. This suggests that further optimization is possible with respect to of other order book properties such as imbalance of top of book, order book depth, volatility, etc. We plan this as future work.

	MAPE(%)	Q95(%)	Standard Deviation(%)
ARMA	78.742	114.357	22.692
Hawkes	74.742	96.202	12.934
Static	74.662	99.925	13.454
Spread	74.514	96.141	12.525

TABLE 5.2: Performance

5.5 Conclusion

In this chapter I have modeled the impact of big trade events on liquidity. I have developed four algorithms to predict the liquidity impact. All the four algorithms give reasonable result, while the one built on historical statistics with a spread-based classification works best.

I started by giving proper definitions of big trade events, the liquidity response to those events, and the response function over a time range. I have verified with real market data from NSE that the big trade events do have impact on liquidity, and the impact decays over time.

I have developed four algorithms to predict the liquidity impact. The static algorithm uses the historical statistics to predict the impact. The Hawkes algorithm uses the Hawkes process to model the impact. The ARMA algorithm uses ARMA model to forecast. The spread algorithm is based on the static algorithm but classify the big trade events by the spread level when they arrive. The spread algorithm performs the best.

Future work is to research other factors that can have impact on the prediction algorithms. I have shown with empirical results that some stocks have different features than the other, grouping similar stocks together and building specific group profile will be worth to investigate.

Chapter 6

Conclusion

6.1 Summary

I have worked on liquidity modeling in limit order book markets from macro-level daily data to micro-level order book data. All the studies have been done with a scientific and quantitative approach, driven by empirical findings and evidence. With this discipline I have found several cases that it are extremely difficult for complicated mathematical models to beat the ones based on just simple static statistics. One thing I have learnt from these surprising results is that looking at the basic statistics of data should always be the first thing to do, before any theoretical modeling work has been conducted.

I have shown all the algorithms that I have built for liquidity prediction in limit order book markets. I have addressed each of the research questions mentioned in Chapter (1) with a prediction algorithm that either significantly improves classic algorithms, or provides reasonable prediction results.

6.2 Main Findings

Liquidity in the limit order book markets is a concept composed by different properties. Trading strategies need liquidity information such as how much volume will be traded in a future time, how will the volume be distributed in the future time, what impact can trade events have on liquidity, etc. In the previous chapters I have demonstrated how to model such information and do empirical study on real market data.

6.2.1 Total Volume

A good prediction algorithm for daily total volume helps the trading strategies to have a view of the market liquidity before market open. Daily total volume is a low frequency data that has one data point per day, and thus can be modeled with standard time series techniques but I found that there is still space for improvement.

I implemented several models with simple algorithms or from literature work, and then developed a new algorithm based on an ARMA model with a decomposition of the total market volume and the percentage of each individual stock. I have shown empirical studies to illustrate the rationale and design of each algorithm. To test and evaluate the performance of the algorithms, I have provided a framework to run the algorithms on real market data. The new algorithm systematically outperforms the other classic ones.

6.2.2 Intraday Volume Distribution

The limit order book markets are continuous markets, which means there are continuous trading activities during market hours. Thus the intraday volume distribution is an important feature of market liquidity, especially when it follows some seasonality pattern rather than pure random walk.

I have verified with real market data that there is indeed a U-shape seasonality in the intraday volume distribution. Then I tried to build prediction algorithms to model the distribution to improve VWAP trading strategies.

I have found that trying to systematically perform better than the traditional strategy based on static moving average curve can be a very tough task. I have provided massive empirical evidences to demonstrate such difficulty. Having understanding of this difficulty, I have tried another direction to find improvement. With the relaxation on the requirement of trading side for a given order, I have developed a strategy that trades on both buy and sell side for the same order. This strategy shows great improvement over the traditional strategy.

6.2.3 Microstructure Modeling

I looked at the market data at microstructure level. I tried to model the relationship between individual trade events and identify if there is any predictive pattern in the relationship.

I have applied a theoretical model from stochastic processes, i.e. the Hawkes processes which model a self-exciting feature. I started by introducing the mathematical foundation of the processes, and then making exploration of how to calibrate data with both parametric approach using maximum likelihood estimation, and non-parametric approach by solving a Wiener-Hopf equation system. From empirical analysis on real market data, I found that there is an obvious self-exciting feature. However fitting the market data to the model presents poor goodness of fit, which suggests there are features in data not fully captured by the model.

6.2.4 Impact of Big Trade Events

The information of individual trade events reveals lots of insight of limit order book market. The individual trade events are different against each other, they arrive at different time, are executed with different prices and sizes. They may affect the market and the following events with the information they reveal. In particular, the ones that have big trade sizes can have impact on market liquidity.

I have studied the liquidity impact made by the big trade events. I have verified the impact with empirical study on real market data. I have developed four algorithms to model and predict the response function of such events. The rationale behind each algorithm was illustrated and supported by empirical studies on data. All the four algorithms give reasonable prediction results, while the algorithm built on historical statistic with a spread-based classification performs the best.

6.3 Future Work

The total volume prediction algorithm can be extended to use finer level of information, such as intraday volatility, open and close price of each stock, number of order book events, etc. Thus it can capture more information from historical data.

The bidirectional buy/sell strategy gives significant improvement over traditional VWAP trading strategies. However because it trades in both buy and sell side, the resulted traded quantity may exceed the original order quantity. Future work is to minimize this exceeding quantity by optimizing the timing of switching the trading direction.

Although I have verified that the NSE data does present self-exciting feature with the non-parametric kernel estimation method, fitting the parametric univariate Hawkes process to the data failed. One future work is to investigate the cause of the failure and develop a proper model to fit the data. Another work is to use the multivariate version of Hawkes process to study the correlations between data of different stocks.

To better predict market impact for big trades, we can try to group the market data by similar stocks, e.g. liquid ones and illiquid ones, and try to extract common features to improve the prediction within the group.

The data of limit order book markets is amazing for research purpose not only because of its big quantity, but also the richness of information revealed at the microstructure level. Besides liquidity study, there are many other aspects that deserve further investigation and exploration, such as prices, volatility, order book imbalance, etc. Especially with consideration to the combination of the huge amount of data and the microstructure details, we still call for theoretical tools to do further anatomy on such data.

It is also worth mentioning that there is a rapid evolution in the algorithmic trading and high frequency trading space. Heavy employment of computers in the trading business produces lots of challenges in all sorts of aspects, such as latency arbitrage, artificial intelligence trading algorithms, regulatory policies etc.

Bibliography

Aase, K.-G. (2011). Text Mining of News Articles for Stock Price Predictions. Master's thesis, Norwegian University of Science and Technology.

Alfonsi, A., Fruth, A., and Schied, A. (2010). Optimal Execution Strategies in Limit Order Books with General Shape Functions. *Quantitative Finance*, 10(2):143–157.

Almgren, R. and Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40.

Bacry, E., Dayri, K., and Muzy, J. (2012a). Non-Parametric Kernel Estimation for Symmetric Hawkes Processes. Application to High Frequency Financial Data. *The European Physical Journal B*, 85(5).

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J. (2011). Modeling microstructure noise using hawkes processes. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 5740–5743. IEEE.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J. (2012b). Scaling Limits for Hawkes Processes and Application to Financial Statistics. *arXiv.org*, 1202.0842v1.

Bacry, E. and Muzy, J. (2013). Hawkes model for price and trades high-frequency dynamics. *arXiv.org*, 1301.1135.

Bacry, E. and Muzy, J. (2014). Second order statistics characterization of Hawkes processes and non-parametric estimation. *arXiv.org*, 1401.0903.

Bayraktar, E. and Ludkovski, M. (2014). Liquidation in Limit Order Books with Controlled Intensity. *Mathematical Finance*, 24(4):627–650.

Białkowski, J., Darolles, S., and Le Fol, G. (2008). Improving VWAP strategies: A dynamic volume approach. *Journal of Banking & Finance*, 32(9):1709–1722.

Bollen, J., Mao, H., and Zeng, X.-J. (2010). Twitter mood predicts the stock market. *arXiv.org*, (1):1–8, 1010.3003v1.

Bowsher, C. G. (2007). Modelling Security Market Events in Continuous Time: Intensity Based, Multivariate Point Process Models. *Journal of Econometrics*, 141(2):876–912.

Carlsson, J., Foo, M., Lee, H., and Shek, H. (2007). High Frequency Trade Prediction with Bivariate Hawkes Process. Technical report.

Cartea, Á., Jaimungal, S., and Ricci, J. (2011). Buy Low Sell High: A High Frequency Trading Perspective. *ssrn.com*, 1964781.

Chandrapala, P. (2011). The Relationship Between Trading Volume and Stock Returns. *Journal of Competitiveness*, 3:41–49.

Chavez-Demoulin, V. and McGill, J. A. (2012). High-frequency financial data modeling using Hawkes processes. *Journal of Banking & Finance*, 36(12):3415–3426.

Christensen, H. L. and Woodmansey, R. (2013). Prediction of Hidden Liquidity in the Limit Order Book of GLOBEX Futures. *The Journal of Trading*, 8(3):69–95.

Cont, R. and Larrard, A. D. (2012a). Order Book Dynamics in Liquid Markets: Limit Theorems and Diffusion Approximations. *arXiv.org*, 1202.6412v1.

Cont, R. and Larrard, A. D. (2012b). Price Dynamics in a Markovian Limit Order Market. *arXiv.org*, 1104.4596.

Cont, R., Stoikov, S., and Talreja, R. (2010). A Stochastic Model for Order Book Dynamics. *Operations Research*, 58(3):549–563.

Darolles, S. and Fol, G. L. (2003). Trading Volume and Arbitrage. Working Papers 2003-46, Centre de Recherche en Economie et Statistique.

Engle, R. and Russell, J. R. (1998). Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66(5):1127–1162.

Filimonov, V. and Sornette, D. (2013). Apparent criticality and calibration issues in the Hawkes self-excited point process model: application to high-frequency financial data. *arXiv.org*, 1308.6756.

Fruth, A., Schoeneborn, T., and Urusov, M. (2011). Optimal Trade Execution and Price Manipulation in Order Books with Time-Varying Liquidity. *arXiv.org*, 1109.2631v1.

Hardiman, S. J., Bercot, N., and Bouchaud, J. (2013). Critical reflexivity in financial markets: a Hawkes process analysis. *arXiv.org*, 1302.1405v2.

Hautsch, N. (2002). Modelling Intraday Trading Activity Using Box-Cox-ACD Models. Technical report.

Hautsch, N. and Huang, R. (2012). On the Dark Side of the Market. *ssrn.com*, 2004231.

Hawkes, A. (1971a). Point Spectra of Some Mutually Exciting Point Processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 33(3):438–443.

Hawkes, A. (1971b). Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90.

Hewlett, P. (2006). Clustering of Order Arrivals, Price Impact and Trade Path Optimisation. *Workshop on Financial Modeling with Jump processes, Ecole Polytechnique*.

Humphery-Jenner, M. L. (2011). Optimal VWAP trading under noisy conditions. *Journal of Banking & Finance*, 35(9):2319–2329.

Junca, M. (2011). Stochastic Impulse Control on Optimal Execution with Price Impact and Transaction Cost. *arXiv.org*, 1103.3482v2.

Kakade, S. M., Kearns, M., Mansour, Y., and Ortiz, L. E. (2004). Competitive algorithms for vwap and limit order trading. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 189–198, New York, NY, USA. ACM.

Karpoff, J. M. (1987). The Relation between Price Changes and Trading Volume: A Survey. *Journal of Financial and Quantitative Analysis*, 22(01):109–126.

Kearns, M. and Nevmyvaka, Y. (2013). Machine Learning for Market Microstructure and High Frequency Trading. In O'Hara, M., Lopez de Prado, M., and Easley, D., editors, *High Frequency Trading - New Realities for Traders, Markets and Regulators*, pages 91–124. Risk Books.

Kissell, R. (2013). *The Science of Algorithmic Trading and Portfolio Management*. Academic Press.

Konishi, H. (2002). Optimal slice of a VWAP trade. *Journal of Financial Markets*, 5:197–221.

Lehalle, C. (2013). Market microstructure knowledge needed for controlling an intra-day trading process. In Fouque, J. and Langsam, J. A., editors, *Handbook on Systemic Risk*, pages 549–578. Cambridge University Press.

Lehalle, C. and Tapia, J. F. (2011). Optimal Portfolio Liquidation with Limit Orders. *arXiv.org*, 1106.3279v2.

Lewis, E. and Mohler, G. (2011). A Nonparametric EM Algorithm for a Multiscale Hawkes Process. *Joint Satistical Meetings 2011*.

Loveless, J. (2013). Barbarians at the gateways. *Communications of the ACM*, 56(10):42–49.

Loveless, J., Stoikov, S., and Waeber, R. (2013). Online algorithms in high-frequency trading. *Communications of the ACM*, 56(10):50–56.

Marsan, D. and Lengliné, O. (2008). Extending Earthquakes' Reach Through Cascading. *Science*, 319(5866):1076–1079.

McCulloch, J. and Kazakov, V. (2007). Optimal vwap trading strategy and relative volume. Technical Report 201, Quantitative Finance Research Centre, University of Technology, Sydney.

Meucci, A. (2011). "P" versus "Q": Differences and Commonalities between the Two Areas of Quantitative Finance. *GARP Risk Professional*, pages 47–50.

Mills, T. C. (1991). *Time Series Techniques for Economists*. Cambridge University Press.

Nagar, A. and Hahsler, M. (2012). Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams. In *International Proceedings of Computer Technology and Science*, pages 91–96.

Nevmyvaka, Y., Feng, Y., and Kearns, M. (2006). Reinforcement learning for optimized trade execution. In Cohen, W. W. and Moore, A., editors, *Proceedings of the 23th International Conference on Machine Learning (ICML-06)*, pages 673–680.

Nyström, E. J. (1925). *Über die numerische Integration von Differentialgleichungen:(Mitgeteilt am 23. Sept. 1925 von E. Lindelöf und KF Sundman)*. Societas scientiarum Fennica.

Ogata, Y. (1978). The Asymptotic Behaviour of Maximum Likelihood Estimators for Stationary Point Processes. *Annals of the Institute of Statistical Mathematics*, 30(1):243–261.

Ogata, Y. (1981). On Lewis' Simulation Method for Point Processes. *Information Theory, IEEE Transactions on*, 27(1):23–31.

Ogata, Y. (1988). Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes. *Journal of the American Statistical Association*, 83(401):9–27.

Papangelou, F. (1972). Integrability of Expected Increments of Point Processes and a Related Random Change of Scale. *Transactions of the American Mathematical Society*, 165:483–483.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition.

Schumaker, R. P., Zhang, Y., and Huang, C.-N. (2009). Sentiment analysis of financial news articles. *4th Annual Conference Behavioural Models & Sentiment Analysis Applied to Finance*.

Shek, H. (2010). Modeling High Frequency Market Order Dynamics Using Self-Excited Point Process. *ssrn.com*, 1668160.

Toke, I. M. and Fabrizio, P. (2012). Modelling Trades-Through in a Limited Order Book Using Hawkes Processes. *Economics: The Open-Access, Open-Assessment E-Journal*, 6(2012-22).

Winne, R. D. and D'hondt, C. (2007). Hide-and-Seek in the Market: Placing and Detecting Hidden Orders. *Review of Finance*, 11(4):663–692.

World Federation of Exchanges (2012). 2011 WFE Market Highlights. Technical report.

Zhang, L. (2013). Sentiment analysis on Twitter with stock price and significant keyword correlation. Master's thesis, The University of Texas at Austin.