Quantitative Data Analysis in Finance

Chapter · December 2016

DOI: 10.1007/978-3-319-49340-4_21

Xiang Shi

Stony Brook University

Peng Zhang

Stony Brook University

Chapter Title:

Quantitative Data Analysis in Finance

Xiang Shi, Peng Zhang and Samee U. Khan

Abstract: Quantitative tools have been widely adopted in order to extract the massive information from a variety of financial data. Mathematics, statistics and computers algorithms have never been so important to financial practitioners in history. Investment banks develop equilibrium models to evaluate financial instruments; mutual funds applied time series to identify the risks in their portfolio; and hedge funds hope to extract market signals and statistical arbitrage from noisy market data. The rise of quantitative finance in the last decade relies on the development of computer techniques that makes processing large datasets possible. As more data is available at a higher frequency, more researches in quantitative finance have switched to the microstructures of financial market. High frequency data is a typical example of big data that is characterized by the 3V's: *velocity, variety* and *volume*. In addition, the signal to noise ratio in financial time series is usually very small. High frequency datasets are more likely to be exposed to extreme values, jumps and errors than the low frequency ones. Specific data processing techniques and quantitative models are elaborately designed to extract information from financial data efficiently.

In this chapter, we present the quantitative data analysis approaches in finance. First, we review the development of quantitative finance in the past decade. Then we discuss the characteristics of high frequency data and the challenges it brings. The quantitative data analysis consists of two basic steps: (i) data cleaning and aggregating; (ii) data modeling. We review the mathematics tools and computing technologies behind the two steps. The valuable information extracted from raw data is represented by a group of statistics. The most widely used statistics in finance are expected return and volatility, which are the fundamentals of modern portfolio theory. We further introduce some simple portfolio optimization strategies as an example of the application of financial data analysis.

Big data has already changed financial industry fundamentally; while quantitative tools for addressing massive financial data still have a long way to go. Adoptions of advanced statistics, information theory, machine learning and faster computing algorithm are inevitable in order to predict complicated financial markets. These topics are briefly discussed in the later part of this chapter.

Xiang Shi, Ph.D.
Stony Brook University, Stony Brook, NY 11794, USA
e-mail: xiang.shi@stonybrook.edu

Peng Zhang, Ph.D. ✉
Stony Brook University, Stony Brook, NY 11794, USA
e-mail: peng.zhang@stonybrook.edu

Samee U. Khan, Ph.D.
North Dakota State University, Fargo, ND 58108, USA
e-mail: samee.khan@ndsu.edu

1. Introduction

1.1 History of Quantitative Finance

The modern quantitative finance or mathematical finance is an important field of applied mathematics and statistics. The major task of it is to model the finance data, evaluate and predict the value of an asset, identify and manage the potential risk in a highly scientific way. One can divide the area of quantitative finance into two distinct branches based on its tasks, (Meucci 2011). The first one is called the "$\mathbb{Q}$" area, which serves to price the derivatives and other assets. The character "$\mathbb{Q}$" denotes the risk-neutral probability. The other one is the "$\mathbb{P}$" area, which are developed to predict the future movements of the market. The character "$\mathbb{P}$" denotes the "real" probability of the market.

The first influential theory in quantitative finance is the Black-Scholes option pricing theory. Unlike public equities that are frequently traded in the market, derivatives like options often lack liquidity and are hard to be evaluated. The theory was initiated by (Merton 1969) who applied continuous time stochastic models to get the equilibrium price of equity. (Black and Scholes 1973) derive an explicit formula for option pricing based on the idea of arbitrage free market. This formula, as (Duffie 2010) called, is "the most important single breakthrough" of the "golden age" of the modern asset pricing theory. Following works by (Cox and Ross 1976), (Cox, Ross et al. 1979) and (Harrison and Kreps 1979) form the footstone of the "$\mathbb{Q}$" area. The theory is most widely applied in sell-side firms and market makers like large investment banks. Today the Black-Scholes formula is the core curriculum of any quantitative programs in university. The fundamental mathematical tools in this area are Ito's stochastic calculus, partial differential equation and modern probability measure theory developed by Kolmogorov. The security and the derivatives are often priced individually, thus high dimensional problems are often not considered in classical "$\mathbb{Q}$" theories.

Unlike the "$\mathbb{Q}$" theory which focuses on measuring the present; the goal of the "$\mathbb{P}$" area is to predict the future. Financial firms who are keen on this area are often mutual funds, hedge funds or pension funds. Thus the ultimate goal of the "$\mathbb{P}$" area is portfolio allocation and risk management. The foundation of the "$\mathbb{P}$" world is the modern portfolio theory developed by (Markowitz 1952). The idea of Markowitz's theory is that any risk-averse investor tends to maximize the expected returns (alpha) of his portfolio while the risk is under control. Other important contributions to this area are the capital asset pricing model (CAPM) introduced by (Treynor 1961), (Sharpe 1964), (Lintner 1965) and (Mossin 1966).

Financial data is fundamentally discrete in nature. In the "$\mathbb{Q}$" area, asset prices are usually approximated by a continuous-time stochastic process so that one can obtain a unique equivalent risk-neutral measure. The continuous-time process, however, has difficulties in capturing some stylized facts in financial data such as mean-reverting, volatility clustering, skewness and heavy-tailness unless highly sophisticated theories are applied to these models. Thus the "$\mathbb{P}$" area often prefers discrete-time financial econometric models that can address these problems more easily than their continuous-time counterparties. (Rachev, Mittnik et al. 2007) suggest that there are three fundamental factors that make the development of financial econometrics possible, which are: *"(1) the availability of data at any desired frequency, including at the transaction level; (2) the availability of powerful desktop computers and the requisite IT infrastructure at an affordable cost; and (3) the availability of off-the- shelf econometric software."*

Furthermore, most problems in the "$\mathbb{P}$" area are high dimensional. Portfolio managers construct their portfolios from thousands of equities, ETFs or futures. Dependence structure among these risky assets is one of the most important topics in the "$\mathbb{P}$" world. Traditional statistics are challenged by these high dimensional financial data and complicated econometric models.

Thus the big data together with related techniques is the foundation of the "$\mathbb{P}$" world, just like coal and petroleum that make the industrialization possible. And the technologies behind big data become more important as the development of high frequency trading. Just a decade ago, the major research in the "$\mathbb{P}$" area was based on the four prices: Open, High, Low, Close (OHLC) that are reported at the end of each day. Data at higher frequency was not provided or even kept by most of the exchanges. For example, commodity trading floors did not keep intraday records for more than 21 days until 6 years ago, (Aldridge 2015). Comparing to the low frequency OHLC data, the high frequency data is often irregularly spaced, and exhibits stronger mean-reverting and periodic patterns. A number of researches in econometrics have switched to the high frequency area. As an example, we use the keywords "financial econometrics" and "high frequency" to search related publications on Google Scholar®. To compare we also search the results of "financial econometrics" only. Figure 1 plots the number of the publications during each period.

One can observe that there is a tremendous growth of financial econometrics publications over the past decade. The percentage of the papers related to high frequency data is about 13% in 1990-1994 periods. This number increases to about 34% and 32% in 2005-2009 and 2010-2014 periods. Figure 1 is also an evidence of the growing importance of the big data in finance; since the high frequency data is a typical example of big data that is characterized by the 3Vs: velocity, variety and volume. We discuss these concepts in depth in the following section.

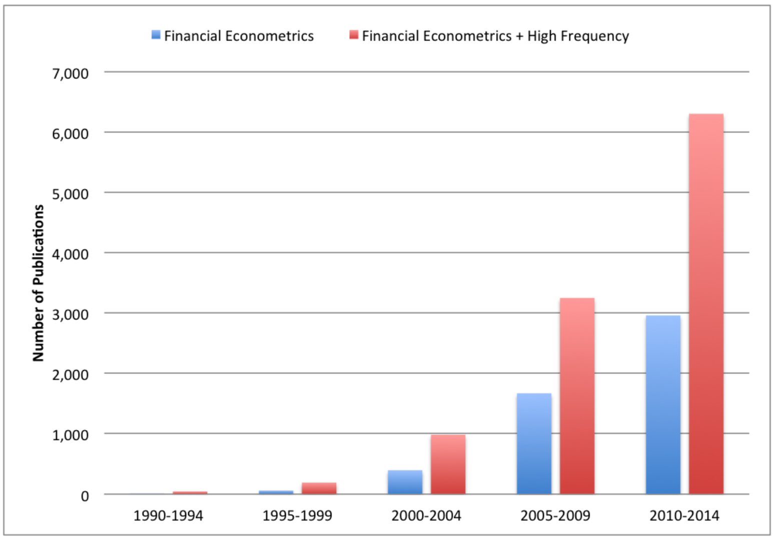


Figure 1: Number of publications related to high frequency econometrics on Google Scholar® (Data source: Google Scholar®)

1.2 Compendium of Terminology and Abbreviations

Briefly, we summarize the terminology and abbreviations in this chapter:

Algorithmic trading strategy refers to a defined set of trading rules executed by computer programs.

Quantitative data analysis is a process of inspecting, cleaning, transforming, and modeling data based on mathematical models and statistics.

Moore's law is the observation that the number of transistors in a dense integrated circuit doubles approximately every two years.

Equity is a stock or any other security representing an ownership interest. In this chapter, the term "equity" only refers to the public traded ones.

High frequency data refers to intraday financial data in this chapter.

ETF refers to exchange traded fund, is a marketable security that tracks an index, a commodity, bonds, or a basket of assets like an index fund.

Derivative refers to a security with a price that is dependent upon or derived from one or more underlying assets.

Option refers to a financial derivative that represents a contract sold by one party (option writer) to another party (option holder). The contract offers the buyer the right, but not the obligation, to buy (call) or sell (put) a security or other financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date).

Buy side is the side of the financial industry comprising the investing institutions such as mutual funds, pension funds and insurance firms that tend to buy large portions of securities for money-management purposes.

Sell side is the part of the financial industry involved with the creation, promotion, analysis and sale of securities. Sell-side individuals and firms work to create and service stock products that will be made available to the buy side of the financial industry.

Bid price refers to the maximum price that a buyer or buyers are willing to pay for a security.

Ask price refers to the minimum price that a seller or sellers are willing to receive for the security. A trade or transaction occurs when the buyer and seller agree on a price for the security.

Table 1: List of Abbreviations

TAQ data	Trade and quote data
OHLC	Traditional open, high, low, close price data
HFT	High frequency trading
MLE	Maximum likelihood estimator
QMLE	Quasi-maximum likelihood estimator
PCA	Principle component analysis
EM	Expectation maximization
FA	Factor analysis
ETF	Exchange traded fund
NYSE	New York stock exchange
AR	Autoregressive model
ARMA	Autoregressive moving average model
GARCH	Generalized autoregressive conditional heteroscedasticity model
ACD	Autoregressive conditional duration

2. The Three V's of Big Data in High Frequency Data

Big data is often described by the three V's: velocity, variety and volume, all of which are the basic characteristics of high frequency data. The three V's bring both opportunities and difficulties to practitioners in finance (Fang and Zhang 2016). In this section we introduce the concept, historical development and challenges of high frequency data.

2.1 Velocity

Telling about the velocity of the high frequency data seems to be tautology. Over the past two decades, the financial markets adopt computer technologies and electronic systems. This leads to a dramatic change of the market structure. Before 1970s, the traditional market participates usually negotiate their trading ideas via phone calls. Today most of jobs of the traditional traders and brokers are facilitated by computers, which are able to handle tremendous amount of information in an astonishing speed. For example, the NYSE TAQ (Trade and Quote) data was presented in seconds' timestamp when it was first introduced in 1997. This was already a huge advance comparing to the pre 1970s daily data. Now the highest frequency of the TAQ data is in millisecond, which is a thousand of a second. Furthermore, a stock can have about 500 quote changes and 150 trades in a millisecond. No one would be surprised if the trading speed would grow even faster in the near future because of Moore's law. As a result, even traditional low frequency traders may need various infrastructures, hardware and software techniques to reduce their transaction costs in their transactions. The high frequency institutions, on the other side, are willing to invest millions of dollars not only on computer hardware but also on real estate; since 300 miles closer to the exchange will provide about one millisecond advantage in sending and receiving orders.

2.2 Variety

With the help of electronic systems the market information can be collected not only in higher frequency but also in a greater variety. Traditional price data of a financial instrument usually consists of only 4 components: open, high, low, close (OHLC). The microstructure of the price data is fundamentally different with the daily OHLC, which are just 4 numbers out of about ten thousands trade prices of equity in a single day. For example, the well-known bid-ask spread which is the difference between the highest bid price and the lowest ask price is the footstone of many high frequency trading strategies. The level 2 quote data also contains useful information can be used to identify buy/sell pressure. Another example is the duration, which measures how long it takes for price change, can be used to detect the unobservable good news in the market. (Diamond and Verrecchia 1987) and (Easley and O'hara 1992) suggest that the lower the durations, the higher probability of the presence of the good news when the short selling is not allowed or limited. Together with the trade volume, the duration can also be a measurement of market volatility. (Engle and Russell 1998) first found the intraday duration curve that indicated the negative correlation with the U-shaped volatility pattern.

2.3 Volume

Both velocity and variety contributes to the tremendous volume of the high frequency data. And that amount is still growing. The total number of transactions in the US market has been increased by 50 times in the last decade. If we assume that there are about 252 trading days in each year, then the number of quotes observed on November 9, 2009, for SPY alone would be greater than 160 years of daily OHLC and volume data points, (Aldridge 2009). Not only the number of records, but also the accuracy is increasing. The recent TAQ prices are truncated to five implied decimal places comparing to the two decimal digits of the traditional daily price data. The size of one-day trade data is about 200MB on average; while the quote data is about 30 times larger than the trade data. Most of these records are contributed by the High Frequency Trading (HFT) companies in US. For example, in 2009 the HFT accounted for about 60~73 % of all US equity trading volume while the number of these firms is only about 2% overall operating firms, (Fang and Zhang 2016).

2.4 Challenges for High Frequency Data

Like most Big Data, high frequency data is a two-sided sword. While it carries a great amount of valuable information; it also brings huge challenges to quantitative analyst, financial engineers and data scientists. First of all, most high frequency data are inconsistent. These data are strongly depended on the regulations and procedures of the institution that collects them, which varies for different periods and different exchanges. For example, the bid-ask spreads in NYSE are usually smaller than the ones in other exchanges. Moreover, a higher velocity in trading means a larger likelihood that the data contains wrong records. As a result, some problematic data points should be filtered out the raw data; and a fraction of the whole data can be used in practice.

Another challenge is the discreteness in time and price. Although all financial data are discrete, many of them can be approximately modeled by a continuous stochastic process or a continuous probability distribution. The classical example of Black Scholes formula is based on the assumption of geometric Brownian motion price process. However this is not the case for high frequency data. The tick data usually falls on a countable set of values. Figure 2 plots the histogram of the trade price changes of IBM on Jan 10, 2013. There are about 66% of the prices are the same as the previous one. And about 82% of the price changes fall in -1 to 1 cent. Similar observation can be found in (Russell, Engle et al. 2009). Another property of high frequency data is the bid-ask bounce. Sometimes it can be observed that the prices frequently back and forth between the best bid and ask price. This phenomenon introduces a jump process that differs with many traditional models. Furthermore, the irregularly spaced data makes it difficult to be fitted by most continuous stochastic processes that are widely used in modeling daily returns. The problem becomes even harder in high dimension, since the duration pattern varies in different assets.

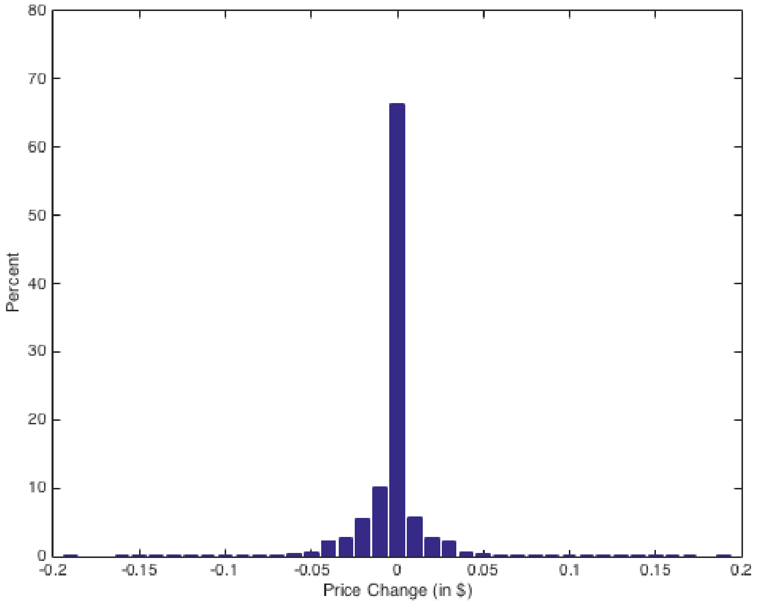


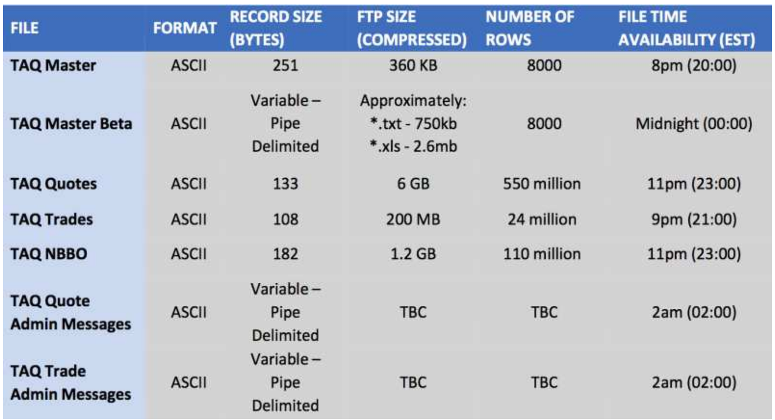
Figure 2: Histogram of the trade price changes of IBM on Jan 10, 2013

3. Data Cleaning, Aggregating and Management

Cleaning data is the first step of any data analysis, modeling and prediction. The raw data provided by data collectors is referred as dirty data, since it contains inaccurate or even incorrect data point almost surely. In addition data cleaning is sometimes followed by data aggregation that generates data with a desired frequency. The size of data is often significantly reduced after the two steps. Thus one can extract useful information from the cleaned data in a great efficiency.

In this section we take NYSE TAQ data as an example. Table 2 lists the details of daily TAQ files. The information is available on *http://www.nyxdata.com/Data-Products/Daily-TAQ*.

Table 2 Daily TAQ file details (Source: https://www.nyxdata.com/doc/243156.)



FILE	FORMAT	RECORD SIZE (BYTES)	FTP SIZE (COMPRESSED)	NUMBER OF ROWS	FILE TIME AVAILABILITY (EST)
TAQ Master	ASCII	251	360 KB	8000	8pm (20:00)
TAQ Master Beta	ASCII	Variable – Pipe Delimited	Approximately: *.txt - 750kb *.xls - 2.6mb	8000	Midnight (00:00)
TAQ Quotes	ASCII	133	6 GB	550 million	11pm (23:00)
TAQ Trades	ASCII	108	200 MB	24 million	9pm (21:00)
TAQ NBBO	ASCII	182	1.2 GB	110 million	11pm (23:00)
TAQ Quote Admin Messages	ASCII	Variable – Pipe Delimited	TBC	TBC	2am (02:00)
TAQ Trade Admin Messages	ASCII	Variable – Pipe Delimited	TBC	TBC	2am (02:00)

3.1 Data Cleaning

As we have discussed in the previous section, most of high frequency data contains certain errors. Some of them can be detected simply by plotting all the data points. Figure 3 plots all the trade prices of IBM on Jan 10, 2013. The trades not happened in regular market hours (9:30 AM to 4:00 PM) are also included in the dataset. This kind of data lacks liquidity and contains more outliers than the others; and therefore they are not considered in most data analysis. But one can also observe that there are several abnormal outliers within the regular hours.

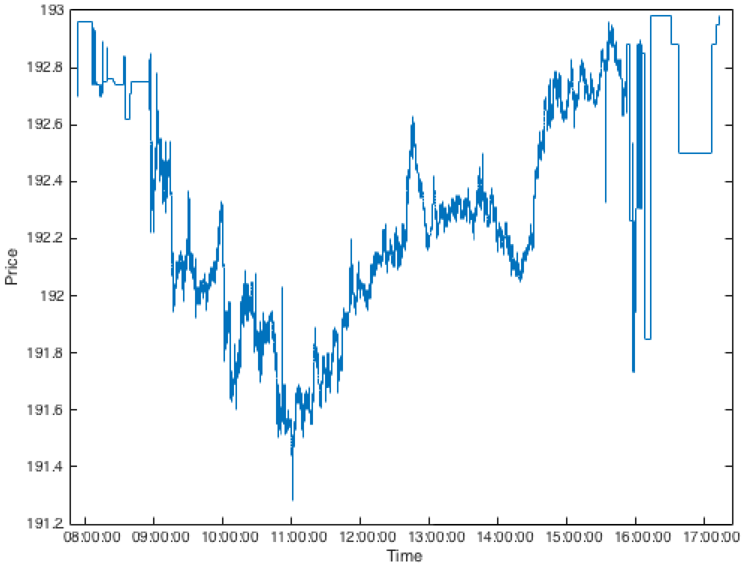


Figure 3: the trade prices of IBM on Jan 10, 2013

We introduce several numerical approaches for cleaning high frequency data. The first step is to filter out the data that potentially have lower quality and accuracy. For example, (Brownlees and Gallo 2006) suggest removing non-NYSE quotes in TAQ data; since NYSE records usually have less outlier than the non-NYSE ones as shown by (Dufour and Engle 2000). In addition, the data record that were corrected or delayed should also be removed. These kinds of information about data condition and location are listed in COND, CORR and EX columns in the TAQ data, see (Yan 2007) for details.

Consider a price sequence Error! Bookmark not defined. ⍰ where $i = 1,2, \ldots N$ with length (Brownlees and Gallo 2006) propose the following algorithm for removing outliers:

$$\mathrm{I}(|p_i - \bar{p}_i(k)| < 3s_i(k) + \phi) = \begin{cases} \text{true}, & \text{observation } i \text{ is kept.} \\ \text{false}, & \text{observation } i \text{ is removed.} \end{cases} \mathrm{f}$$

where $\bar{p}_i(k)$ and $s_i(k)$ are the α-trimmed mean and standard deviation of a neighborhood of k observations and ϕ is a positive number called granularity parameter. ϕ is to prevent p_i to be removed when $s_i(k) = 0$. As we have seen in Figure 2 high frequency data often contains many equal prices. α is a percentage number. For example, a 10%-trimmed mean and standard deviation are the average of the sample excluding the smallest 10% and the largest 10% numbers. Thus outliers and unreasonable data points have less impact on the trimmed statistics. Median can be viewed as a fully trimmed mean. (Mineo and Romito 2007) propose a slightly different algorithm:

$$\mathrm{If}\ (|p_i - \bar{p}_{-i}(k)| < 3s_{-i}(k) + \varphi) = \begin{cases} \text{true}, & \text{observation } i \text{ is kept.} \\ \text{false}, & \text{observation } i \text{ is removed.} \end{cases}$$

where $\bar{p}_{-i}(k)$ and $s_{-i}(k)$ are the α -trimmed mean and standard deviation of a neighborhood of k observations excluding p_i. (Mineo and Romito 2008) apply both algorithms to the ACD model and conclude that the performances of the two algorithms are very similar, while the second one might be better in modeling the correlations of model residuals.

The α-trimmed mean and standard deviation are the robust estimates of the location and dispersion of a sequence. The robustness depends on the choice of α. Prior knowledge of the percentage of outliers in the data is required in order to find the best α. The optimal α of each asset would be different. In some cases the α-trimmed mean and the standard deviation can be replaced by the following statistics:

$$\bar{p}_i(k) = \mathrm{median}\{p_j\}_{j=i-k,\ldots,i+k}$$

$$s_i(k) = c \cdot \mathrm{median}\{|p_j - \bar{p}_i(k)|\}_{j=i-k,\ldots,i+k}$$

where c is a positive coefficient. Outlier detecting algorithms with above statistics are sometimes called Hampel filter that is widely used in engineering. The second equation can be generalized by replacing the median by quartile with certain level. The median based $\bar{p}_i(k)$ and $s_i(k)$ are also more robust than the trimmed ones

A very important issue the data cleaning approaches is that the volatility of the cleaned data depends on the choice of methods and corresponding parameters. The volatility of many high frequency data, including equity and currency, exhibits strong periodic patterns. The outlier detection algorithms with moving window can potentially diminish or remove these patterns that are important in prediction and risk control. Thus it is crucial to consider the periodic behavior before using above algorithms directly. One way is to apply robust estimates of volatility to raw data and then remove this effect via certain adjustment. We discuss this problem in Section 4.1.

3.2 Data Aggregating

Most econometric models are developed for equally spaced time series, while most high frequency data are irregular spaced and contains certain jumps. In order to apply these models to the high frequency data, some aggregating techniques are necessary for generating equally spaced sequence from the raw data. Consider a sequence $\{(t_i, p_i)\}$ where $i = 1, \dots, N$, t_i is time step and p_i is trade or quote price. Given an equally-spaced time stamps $\{\tau_j\}$ where $j = 1, \dots, M$ and $\tau_j - \tau_{j-1} = \tau_{j+1} - \tau_j$ for all j, a simple but useful way to construct a corresponding price series $\{q_j\}$ where $j = 1, \dots, M$ is to take the previous data point:

$$q_j = p_{i_{last}}$$

where $i_{last} = \max\{i | t_i \le \tau_j, i = 1, \dots, N\}$. This approach is called last point interpolation. It assumes that the price would not change before the new data come in. (Gençay, Dacorogna et al. 2001) propose a linear interpolation approach:

$$q_j = p_{i_{last}} + \left(p_{i_{next}} - p_{i_{last}}\right) \frac{\tau_j - t_{i_{last}}}{t_{i_{next}} - t_{i_{last}}}$$

where $i_{next} = \min\{i | t_i \ge \tau_j, i = 1, \dots, N\}$. The second method is potentially more accurate than the first one, but one should be very careful when use it in practice, especially in back-testing model or strategies; since it contains the future information $p_{i_{next}}$ which is not available at τ_j.

There are several ways to deal with the undesirable jumps caused by bid-ask bounce. The most widely used approach is to replace the trade prices by the mid-quote prices. Let $\{(t_i^b, p_i^b)\}$ where $i = 1, \dots, N^b$ and $\{(t_i^a, p_i^a)\}$ where $i = 1, \dots, N^a$ be the best bid and ask prices together with their time stamps. The mid-quote price is given by

$$p_i = \frac{1}{2}\left(p_{i^b}^b + p_{i^a}^a\right)$$

where

$$t_i = \max\{t_{i^b}^b, t_{i^a}^a\}$$
$$i_b = \min\{i | t_i^b > t_{i-1}, i = 1, \dots, N^b\}$$
$$i_a = \min\{i | t_i^a > t_{i-1}, i = 1, \dots, N^a\}$$

Another approach is to weight the bid and ask by their sizes s_i^b and s_i^a

$$p_i = \frac{s_{ib}^b \cdot p_{ib}^b + s_{ia}^a \cdot p_{ia}^a}{s_{ib}^b + s_{ia}^a}$$

Once we get an equal time spaced price series $\{q_j\}$ where $j = 1, \ldots, M$, we are able to calculate the log returns of the asset:

$$r_j = \log \frac{q_j}{q_{j-1}}$$

In high frequency data, the price difference is usually very small. Thus the log returns would be very close to the real returns

$$r_j \approx \frac{q_j - q_{j-1}}{q_{j-1}}$$

There are several good reasons to consider the log returns instead of the real returns in financial modeling. First it is symmetric with respect to the up and down of the prices. If the price increases 10% and decreases 10% in terms of the log return, then it will remain the same. The real return can exceed 100% but cannot be lower than -100% while the log return does not have this limit. Furthermore the cumulative log returns can be simply represented as the sum of the log returns; this fact would be very helpful in applying many linear models to the log returns.

The last thing we want to mention here is that the size of overnight returns in equity market is often tremendous comparing to the size of intraday returns. The currency market does not have that problem. Overnight returns in equity market are often considered as outliers and removed from the data in most applications. One can also rescale these returns since they may contain useful information. But different methods in rescaling overnight returns might affect the performance of model and strategy.

3.3 Scalable Database and Distributed Processing

Cleaning and aggregating high-volume data always needs a big data infrastructure that combines a data warehouse and a distributed processing platform. To address the challenges of such big data infrastructure with emerging computing multisource platforms such as heterogeneous architectures and Hadoop with emphasis on addressing data-parallel paradigms, people have extensively been working on various aspects, such as scalable data storage and computation management of big data, multisource streaming data processing and parallel computing, etc.

Database is an essential datastore for high-volume finance data such long-term historical market data sets. In data management, the column-based database like NoSQL and in-memory database are replacing the traditional relational database management system (RDBMS) in financial data-intensive applications. RDBMS is database based on the relational model and it has been used for decades in industry. Although it is ideal for processing general transactions, RDBMS is less efficient in processing enormous structured and unstructured data, for examples, for market sentiment analysis, real-time portfolio and credit scoring in modern financial sector. Usually, these financial data are seldom modified but their volume is overwhelmed and they need to be queried frequently and repeatedly. In this, a column based database often stores time series based metadata with support of data compression and quick read. In this regard, the columnar databases are preferably suitable for time series of financial metadata. For example, when a financial engineer pulls out a time series of only a few specified metrics with a specific point, a columnar database is faster for reading than a row-based database since only specified metrics such as OHLC are needed. In this case, a columnar database is more efficient because of the cache efficiency and it has no need for scanning all rows like in a row based database. Beyond the columnar database, the in-memory database is another emerging datastore solution when performing analytics. That is, if the data set is frequently used and its size fits into memory, the data should persist in the memory for sake of data retrieving, eliminating the need for accessing disk-mediated databases. In practice, what solution is favorable should depend on the practitioner's application and available computing facilities.

In addition to data warehouse, distributed processing is equally important. Hadoop often works on Big Data for financial services (Fang and Zhang 2016). Hadoop refers to a software platform for distributed datastore and distributed processing on a distributed computing platform such as a computer cluster. Hadoop is adopted for handling the big data sets for some financial services such as fraud detection, customer segmentation analysis, risk analytics and assessment. In these services, the Hadoop framework helps to enable a timely response. As a distributed data infrastructure, Hadoop does not only include a distributed data storage known as HDFS, Hadoop Distributed File System, but it also offers a data-parallel processing scheme called as MapReduce. However, Hadoop, as a tool, is not a complete big data solution and it has its limitations like everything. For example, it is inefficient to connect structured and unstructured data, unsuitable for real-time analytics, unable to prioritize tasks when multiple tasks are running simultaneously in distributed computing platforms, and its performance closely depends on the scalability of a distributed file system which in turn limits this architecture. Apache Spark, on the other hand, is a data-processing tool and it operates on distributed data storage. Spark does not provide a distributed data storage like HDFS so it needs to be integrated with one distributed data platform. It can run on top of HDFS or it can process structured data in Hive. Spark is an alternative to the traditional map/reduce model that is used by Hadoop and it supports real-time stream data processing and fast queries. Generally, Sparks needs more RAM instead of network and disk-backed I/O and thus it is relatively faster than Hadoop. Spark often completes the full real-time data analytics in memory. However, as it uses large RAM, Spark needs a high-end machine with a large memory capacity. In the code development, Spark is a library for parallel processing through function calls and a Hadoop MapReduce program can be written by inheriting Java classes.

4. Modeling High Frequency Data in Finance

In this section we discuss the mathematical models for high frequency data. There are a number of quantitative models with different features in financial econometrics. The purpose of majority of these models is to estimate expected returns and volatility of a risky asset or portfolio. As we have discussed in the first section, expected return and volatility are the two footstones of the modern portfolio theory. Expected return, sometimes called *alpha*, is the prediction of profit and loss in the future. It is the most crucial statistics for a portfolio manager. Volatility measures variation of value change for a financial instrument or portfolio. The behavior of a portfolio whose volatility is controlled properly is more consistent than the ones with large volatility. Thus Markowitz's theory states that a portfolio may generate relatively stable revenues by maximizing its expected return and minimizing the volatility. Other useful statistics and performance measures such as skewness, kurtosis, VaR or drawdown can also be estimated by some of the following models. There a number of literatures consider portfolio selection and risk management based on these statistics. We will not discuss them in this chapter.

4.1 Volatility Curve

The intraday market exhibits a more clearly periodic pattern especially in volatility comparing to the low frequency financial data. There a number of papers propose different approaches to modeling the volatility of the high frequency data. The most common idea is to separate the volatility into deterministic seasonal part and stochastic part. The deterministic part is usually fitted by a smooth function, as (Andersen and Bollerslev 1997; Andersen, Bollerslev et al. 2000) suggest. The stochastic part can be modeled by ARCH type models, since (Engle and Manganelli 2004) discover volatility clustering effect in high frequency market.

The volatility is often considered as a hidden factor of the market. The most common way to extract seasonal volatility from the data is to compute the norms of the absolute returns. To make it clear, let an integer $K > 0$ be the period length and r_1,r_2,…,r_{KN} be a sequence of equally time-spaced log returns in N periods. Then the seasonal realized volatility can be defined as:

$$v_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left|r_{K(j-1)+i}\right|^p\right)^{\frac{1}{p}}, i = 1,2,\dots,K$$

where the exponent p is usually set to be 1 or 2. However the above representation is sensitive to the outliers. The seasonal structure could be destroyed by a single abnormal extreme value. A more robust way is to consider the quartiles of the absolute returns:

$$v_i = \text{quartile}_\alpha\left\{\left|r_{K(j-1)+i}\right|\right\}_{j=1,\cdots□}$$

where $0 \leq \alpha \leq 1$.

Seasonality with different periods can be observed from the high frequency data. As an example, (Dong 2013) considers 1-minute log returns of all the stocks in Russell 3000 on 2009. The period K is set to be 390 that is the number of minutes in each trading day. Figure 4 plots the volatility curve together with the aggregated volume curves of NYSE and NASDAQ against 390 minutes.

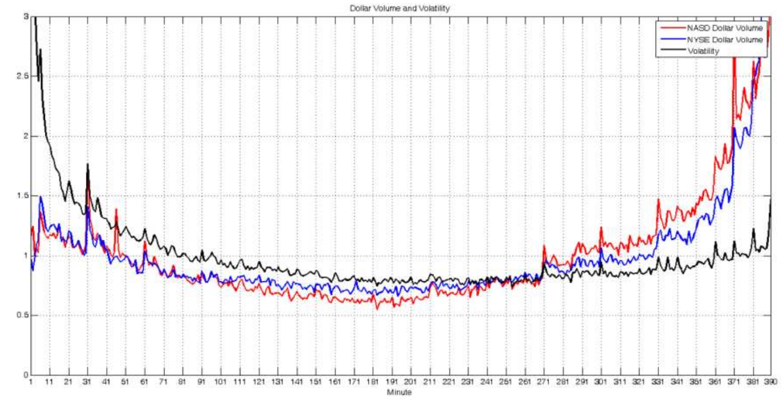


Figure 4: The volatility curve together with the aggregated volume curves of NYSE and NASDAQ against 390 minutes (Credit: (Dong 2013))

In addition (Dong 2013) discovers that there exist 5-minute spikes on the curve. This phenomena are more clear when we plot the volatility curve when $K = 60$ minutes (see Figure 5). Both volatility and volume exhibit the U-shape pattern but they are different at tails. The volatility is relatively higher at market opening and lower at the end.

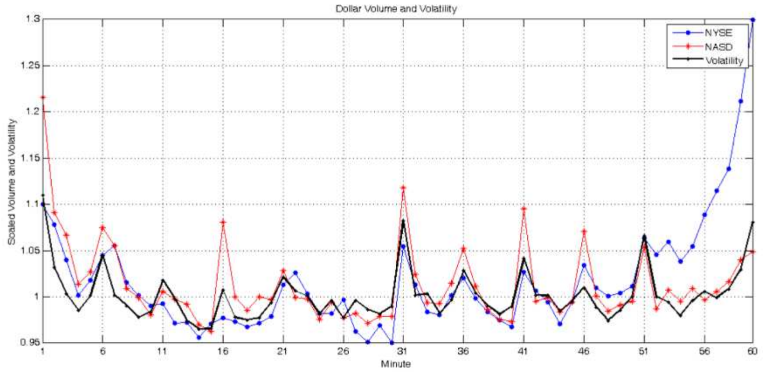


Figure 5: The volatility curve together with the aggregated volume curves of NYSE and NASDAQ against 60 minutes (Credit: (Dong 2013))

To fit the volatility curve above one can use a smooth rational function, for example:

$$f(x) = \frac{ax^2 + bx + c}{dx + 1}$$

The coefficients a, b, c, d can be fitted by least square approach:

$$\min_{a,b,c,d} \sum_{i=1}^{K} (v_i + div_i - ai^2 - bi - c)^2$$

and the de-seasonal log returns can be:

$$\hat{r}_{K(j-1)+i} = \frac{r_{K(j-1)+i}}{f(i)}$$

where $i = 1,2, \dots, K, j = 1,2, \dots, N$. As we have mentioned before, the volatility patterns may not be preserved if we apply the outlier cleaning techniques introduced in Section 3.1 before computing the realized volatility. The quartile-based realized volatility, which is a robust estimator, can be applied directly to uncleaned data. Thus instead of removing outliers in the price, one can first aggregate data and get an equal spaced return series with abnormal outliers. Then the data cleaning approach can be applied to the de-seasonal returns.

4.2 Stochastic Volatility

Despite of the deterministic periodic pattern, the volatility is stochastic and exhibits volatility clustering, i.e. large returns are likely followed by large returns regardless their directions, see (Engle and Manganelli 2004). Thus the generalized autoregressive conditional heteroscedasticity (GARCH) type models developed by (Engle 1982) and (Bollerslev 1986) would be a good choice to fit the stochastic part of the volatility. In this section we briefly introduce the idea of the GARCH (1,1) model. For simplicity let $r_i, i = 1,2, \dots$ be the de-seasonal equally spaced log returns. The GARCH (1,1) model assumes that:

$$r_i = \mu_i + \sigma_i \cdot \epsilon_i$$
$$\sigma_i^2 = \omega + \alpha r_{i-1}^2 + \beta \sigma_{i-1}^2$$

where ω, α, β are positive real numbers, and ϵ_i where $i = 1,2, \dots$ are i.i.d normally distributed with zero mean and unit variance. The drift term μ_i is the conditional expectation of r_i given all the information up to time t. There are a lot of approaches in modeling μ_i that is often called α in finance. We discuss several examples in Section 4.4.

The parameters ω, α, β should satisfy the constraint $\omega + \alpha + \beta \leq 1$ in order to make the process to be stationary. The estimation of the model is usually performed by the maximum likelihood estimator (MLE). We refer to (McNeil, Frey et al. 2005) for details. Scientific programming languages including Matlab and R have matured packages for fitting the GARCH model. In practice, the ω is often a small number close to zero; β ranges from 0.7 to 0.9 and $\alpha + \beta \approx 1$. α is usually much smaller than β, but it plays a key role in measuring the volatility sensitivity to the market impact.

4.3 Multivariate Volatility

The simplest approach to model the dependence structure of multi-assets is to compute the covariance of their returns. However, the traditional sample covariance is usually ill conditioned when the dimension is relatively high comparing to the sample size. An ill conditioned covariance matrix may lead huge errors in risk forecasting and portfolio optimization. The simplest way to improve the conditions of the sample covariance is to adjust its eigenvalues. Another method is to shrink the covariance to some well-conditioned matrix. The most famous shrinkage estimator is proposed by (Ledoit and Wolf 2003).

The third approach, which is most widely used, is to impose certain structure on the covariance. For example, one can assume that a d by d covariance matrix has the expression:

$$\Sigma = FF' + D$$

where F is a d-by-n matrix, D is a d-by-d diagonal matrix and $n < d$. The rational of the above formula is that the asset return follows the linear factor model:

$$r = Fx + \epsilon$$

where r is the d-dimensional vector of log returns, x is the vector of uncorrelated risky factors with unite variance in a lower dimension n, and ϵ is the uncorrelated errors with covariance D. Unlike the traditional factor models, the factor x does not come from real data, which are usually correlated. In this model x is some uncorrelated statistical factors that are hidden from the market. The well-known principle component analysis (PCA) is one way to extract x from the original data. Let $\hat{\Sigma}$ be the sample covariance matrix; by the singular value decomposition it can be written as:

$$\hat{\Sigma} = U\Lambda U'$$

where U is a d-by-d unitary matrix, i.e. $UU' = U'U = I$, and Λ is a diagonal matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Then we can set

$$x = \Lambda_n^{-1/2} U_n' r$$
$$F = U_n \Lambda_n^{1/2}$$

where the d-by-n matrix U_n consists the first n columns of U, and the diagonal matrix Λ_n is the first n-by-n block of Λ. In fact one can show that F is the solution of:

$$\min_F \left\|\hat{\Sigma} - FF'\right\|_2$$

where $\|\cdot\|_2$ is the induced 2-norm of a matrix. The residual matrix D can be simply written as:

$$D = \mathrm{diag}\left(\hat{\Sigma} - FF'\right)$$

The PCA is a simple standard statistical tool for dimension reduction. A potentially more précised approach to fit F and D is to apply the expectation maximization (EM) algorithm to the log returns. This approach is also known as the factor analysis (FA). The standard EM algorithm for FA proposed by Rubin and Thayer (1982) is an iterative algorithm. Let $\{r_i\}$ where $i = 1, \ldots, N$ be a sequence of vectors of log returns and $F^{(0)}$, $D^{(0)}$ be the initial inputs. Then the k-th iteration of the EM is given by:

E Step: Re-compute the conditional expectations:

$$E[x|r_i] = F^{(k-1)\prime}\left(D^{(k-1)} + F^{(k-1)}F^{(k-1)\prime}\right)^{-1} r_i$$

$$E[xx'|r_i] = I - F^{(k-1)\prime}\left(D^{(k-1)} + F^{(k-1)}F^{(k-1)}\right)^{-1} F^{(k-1)} + E[x|r_i] \cdot E[x|r_i]'$$

M Step: Update F and D:

$$F^{(k)} = \left(\sum_{i=1}^{N} r_i E[x|r_i]\right)\left(\sum_{i=1}^{N} E[xx|r_i]\right)^{-1}$$

$$D^{(k)} = \frac{1}{N} diag\left(\sum_{i=1}^{N} r_i r_i' - F^{(k)} E[x|r_i] r_i'\right)$$

The above algorithm will converge to the maximum likelihood estimator of F and D given that x and ϵ are independently Gaussian distributed. There are some variations of the classical EM algorithm that may improve the convergence speed, for example, the ECM algorithm proposed by (Meng and Rubin 1993), Donald B the ECME algorithm proposed by (Liu and Rubin 1994), the GEM algorithm proposed by (Neal and Hinton 1998) and the α-EM algorithm proposed by (Matsuyama 2003). (Jia 2013) applies the α-EM algorithm together with conjugate gradient method to the FA and shows a significant improvement in the speed.

4.4 Expected Return

The high frequency data usually have a stronger cross-sectional dependency than the low frequency one. This fact can be observed not only in the volatility but also in the expected returns or alphas. Thus the classical autoregressive (AR) models may have a better performance in the high frequency market. Let $\{r_i\}$ $i = 1,2, \ldots$ be a sequence of de-seasonal log returns equally spaced in time. The AR(p) model can be written as:

$$r_i = h_0 + h_1 r_{i-1} + h_2 r_{i-2} + \cdots + h_p r_{i-p} + x_i,$$

where $i = p + 1, p + 2, \ldots$; h_0, h_1, …, h_p are called AR coefficients or impulse response in electronic engineering and x_i are often assumed to be i.i.d zero mean normally distributed noises. Given the information up to time $i - 1$, the expectation of r_i, which is given by $h_0 + \sum_{j=1}^{p} h_j r_{i-j}$, is the alpha prediction of the AR(p) model.

The estimation of AR(p) model can be performed by the least squares method. Suppose that we have data samples with length $N > p$, the least squares method solves the following optimization problem:

$$\min_{h_0,\dots,h_p} \sum_{i=p+1}^{N} \left(r_i - h_0 - \sum_{j=1}^{p} h_j r_{i-j} \right)^2 ,$$

which can be solved explicitly:

$$\hat{h} = (R'R)^{-1} R' r,$$

where

$$r = \begin{pmatrix} r_{p+1} \\ \vdots \\ r_N \end{pmatrix},$$

and

$$R = \begin{pmatrix} 1 & r_p & r_{p-1} & \cdots & r_1 \\ 1 & r_{p+1} & r_p & \cdots & r_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & r_{N-1} & r_{N-2} & \cdots & r_p \end{pmatrix}$$

The naïve least squares method is simple, but it is not the most numerically efficient approach in the estimation of AR(p). A better alternative called Burg's method is usually considered a standard approach for estimating AR(p) systems. We refer readers to (Marple Jr 1987). Some software such as Matlab also provides build-in function for Burg's algorithm.

A generalization of the AR(p) model is so-called autoregressive moving average (ARMA) model. Similar as AR(p), the ARMP(p,q) model can be represented as

$$r_i = h_0 + \sum_{j=1}^{p} h_j r_{j-i} + \sum_{j=1}^{q} g_j x_{i-j} + x_i$$

In fact one can show that ARMA(p,q) is also a special case of AR(∞) process. However methods like least squares or Burg's algorithm cannot be applied to the estimation of ARMA(p,q) model. Instead the general maximum likelihood estimator is the standard approach for fitting ARMA(p,q) with normally distributed residuals x_i. The ARMA process often works together with GARCH model. In that case the estimations of ARMA and GARCH can be done separately. This approach is called quasi maximum likelihood (QMLE). A comprehensive introduction of the ARMA-GARCH type models can be found in (McNeil, Frey et al. 2005). (Beck, Kim et al. 2013) apply the ARMA-GARCH model to intraday data with frequency ranged from 75 to 300 seconds; and discover the heavy-tailness in the residuals of the model.

The financial data often has mean-reverting pattern. For example, the estimated h_1 of AR(p) model is usually negative. Roughly speaking, the scales of the rest parameters h_j $(j > 1)$ are small comparing to h_1, and become smaller as j increases, since the impact of historical values to the present will diminish as time goes. However, this does not mean that h_j with large j should be ignored. The aggregation of small impulse responds may have a strong impact to the prediction; since it contains information of long-term trend. (Sun, Rachev et al. 2008) find that the intraday equity data may have long-range dependence, i.e. the decay of h_j with respect to j is very slow. (Kim 2015) applies an ARMA-GARCH model with fractional heavy-tailed distributions to model high frequency data. Although neither ARMA(p,q) nor AR(p) processes can capture the long-range dependency of the data, one may approximate a long-range dependent time series by an AR(p) with large p in a finite amount of time. However as the number of parameters increases, the error of the least squares estimator or Burg's method grows tremendously, due to the Cramér–Rao bound. Thus similar as the covariance matrix estimation, one may need some biased estimators like shrinkage. (Mullhaupt and Riedel 1998) impose a specific structure called triangular input balanced form on the AR process. They show that the estimation error can be significantly reduced by adding small bias to the estimator.

4.5 Duration

Up to now we introduce how to transfer data into equal spaced series. However the frequency of the data would be reduced and certain information would be lost in the aggregation. The original data with irregular time stamps are called "ultra-high frequency" data in (Engle 2000). Consider a sequence of ultra-high frequency data $\{(t_i, p_i)\}$ where $i = 1, \dots, N$, the number of trades that occur before time t is given by $N(t) = \sup\{i | t_i \le t, i = 1, \dots, N\}$. The simplest way is to fit $N(t)$ by a homogeneous Poisson process, i.e. the probability that there k events happen between t and $t + \Delta t$ is:

$$P(N(t + \Delta t) - N(t) = t) = \frac{(\lambda \Delta t)^k}{k!} \exp(-\lambda \Delta t), k = 0,1,2, \dots$$

where λ is the instantaneous arrival rate of an event:

$$\lambda = \lim_{\Delta t \to 0} \frac{P(N(t + \Delta t) - N(t) \ge 0)}{\Delta t}$$

The Poisson process implies that the durations $\Delta t_i = t_i - t_{i-1}$ are i.i.d exponentially distributed with constant rate λ:

$$P(\Delta t \le s) = 1 - e^{-\lambda s}$$

However the Poisson process might be over simplify the problem. Similar as the volatility, duration exhibits periodicity and heteroskedasticity. (Engle 2000) shows that the duration of mid-quote prices has an n-shape curve in contrast to the volatility. The periodicity can be removed using the same approach in Section 4.1. The heteroscedasticity, however, contradicts to the assumption that λ is constant. (Engle and Russell 1998) propose an **a**utoregressive **c**onditional **d**uration (ACD) model as follows:

$$\Delta t_i = \phi_i \epsilon_i$$

$$\phi_i = \omega + \alpha \Delta t_{i-1} + \beta \phi_{i-1}$$

where ϵ_i are i.i.d positive random variables. The ACD model looks very similar to the GARCH model. The distribution of residuals ϵ_i is often set to be the exponential or Weibull distribution. It is clear that the instantaneous arrival rate λ of the

ACD model is not a constant. Simple calculation shows that given $\phi_{N(t)}$ and exponentially distributed ϵ_i:

$$\lambda(t) = \frac{1}{\phi_{N(t)}}$$

Similar as the GARCH model, the parameters of the ACD model can be fitted via QMLE, see (Engle and Russell 1998; Engle 2000).

4.6 Scalable Parallel Algorithms on Supercomputers

As we have seen, all of the computations in previous sections are based on high dimensional matrix operations. For example, multivariate least squares method is applied to fit volatility curves and AR models. Eigenvalues are important in estimating the covariance matrix.

Of these methods, matrix multiplication is the core problem as a basis for most of other methods such as least square, eigenvalue and matrix factorization. Matrix multiplication (MM) is the simplest yet most difficult problem in mathematics (Zhang and Gao 2015). The standard algorithm for MM is $O(n^3)$ but in mathematics, researchers never stop in pursuit of faster approached for multiplying matrices. For example, Strassen reduced the computing complexity to $O(n^{2.8})$ in 1969 and another breakthrough is the Coppersmith-Winograd algorithm that performs MM in $O(n^{2.4})$ operations. In addition to theoretical studies, the complex architectures of computing facilities have further escalated the difficulty for the MM implementation. For example, the task mapping in parallel computers and the task scheduling in hybrid CPU-GPU computers made the MM implementations even harder. In this regard, some data-oriented schedule paradigm is proposed and it has been applied to the MM problem on today's high-performance computing facilities (Zhang, Liu et al. 2015). Of experiments, the best-practice matrix-multiplication approach is found (Zhang and Gao 2015). Figure 6 compares the naïve and Strassen algorithms for tile-based matrix-matrix multiplication.

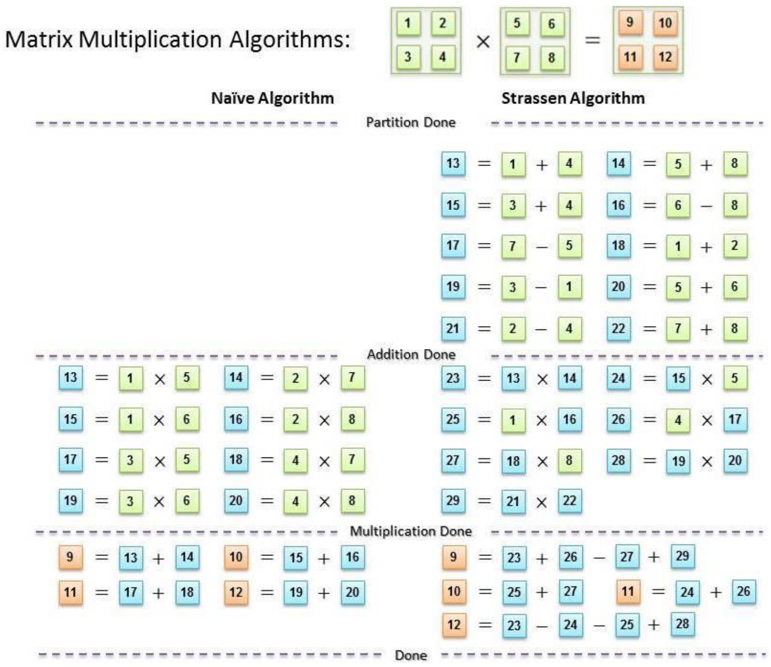


Figure 6: Comparing the naïve algorithm and Strassen algorithm for matrix-matrix multiplication

Cholesky inversion method is to compute the inverse for a positive-definite matrix. In finance, the covariance matrix is positive semidefinite. Cholesky inversion is more challenging than matrix multiplication and it consists of three successive steps: Cholesky factorization, inversion for lower triangular matrix and product of lower triangular matrices. A naïve approach is to perform three steps sequentially but its performance is very poor. To deliver better parallelism, one has to interleave these steps by adhering to the complex data dependencies. This goal could be achieved through a thorough critical path approach (Tomov, Nath et al. 2010) or a dynamic data-oriented schedule approach (Zhang, Gao et al. 2015; Zhang, Liu et al. 2015).

5. Portfolio Selection and Evaluation

Data cleaning, aggregation and modeling can all be viewed as searching valuable information from the massive data. The amount of data would be significantly reduced after each step. Expected return, volatility and other statistics are the gold extracted from raw ore. The final steps are developing trading ideas, constructing portfolios and testing strategies. Although data volume in this procedure is relatively small, there is a great need of computing speed from high frequency investors who want to execute their strategies faster than their opponents. In this section we review two different classes of strategies: Markowitz's mean variance portfolio selection and on-line portfolio selection. The first one is relatively slow but mature and well developed. The second one is simple but fast which can be potentially applied to ultra-high frequency trading.

5.1 Markowitz Portfolio Optimization with Transaction Costs

Suppose that there are d risky assets with expected return μ and covariance Σ. A self-financing portfolio is represented by a d-dimensional weight vector w that satisfies $\sum_{i=1}^{d} w_i = 1$. The well-known Markowitz portfolio states that a rational risk averse investor wants to maximize the utility function:

$$\max_{w} w'\mu - \frac{\lambda}{2} w'\Sigma w$$
$$\text{subject to:}\, e'w = 1$$

where e is a d-dimensional vector with all ones,$w'\mu$ is the expected return of the portfolio, $w'\Sigma w$ is the variance of the portfolio and $\lambda > 0$ reflects the degree of risk aversion. In high frequency market, the log return and the real return are very close, so $w'r$ with log return r can be an approximation of the real portfolio return in a short period. Thus μ and Σ in the optimization problem can be log return based mean and covariance. But this would not be true for long-term prediction. The above optimization problem can be solved explicitly. The optimal portfolio weight together with its expectation and variance changes as the risk aversion parameter λ varies. By plotting the expected return against the variance with all possible λ then we obtain the famous efficient frontier.

There are many variations of the Markowitz mean-variance portfolio strategy. One can replace the variance term $w'\Sigma w$ by other risk measures like the value-at-risk (VaR), conditional value-at-risk (CVaR) or maximum drawdown. These risk measures are often considered to be superior than the variance since they are able to capture the tail-risk. (Rockafellar and Uryasev 2000) show that the mean-CVaR problem can be transferred to a linear programming with a higher dimension. (Chekhlov, Uryasev et al. 2000) propose a similar approach for drawdown measures. However, the trade-off of these approaches is that the dimension of the problem increases tremendously by introducing auxiliary variables. CVaR for example, is often calculated via Monte Carlo; and the dimension of the auxiliary variables in the equivalent linear programming is the same as the number of Monte Carlo scenarios. Regular computers may fail to deal with this kind of problem efficiently due to the memory limitation. Under some special cases the mean-risk problem can be solved easily. For example, (Shi and Kim 2015) show that the dimension of any mean-risk problem with coherent risk measures and a subclass of normal mixture distributions can be reduced to two. In general, however, the mean-risk problem is usually very hard to solve.

The most important problem within the above strategies is that they assume no transaction cost. Transaction cost is usually ignored in the low frequency finance, but it grows dramatically as the trading frequency increases. Broker commissions, exchange fees and taxes are all major sources of the transaction cost. But the most significant one is the portfolio turnovers. For example, if the current best ask price of equity is \$10, it does not mean you are able to buy 500 shares at \$5000. The size of the best asks might just be 200 shares. The next best ask price might be \$10.1 with 300 shares. Overall the average price you paid grows almost proportionally as objective shares increases. Thus a high frequency trader may not choose to change his current position even when he observes a signal.

Even you have a perfect prediction of the expected returns and variance, the optimal mean-variance portfolio may be completely different with the current ones; and the profit would be dwarfed by the huge transaction cost in rebalancing the portfolio. Thus a constraint on the portfolio turnover is necessary in portfolio optimization problems. Suppose that your current portfolio weight is given by a d-dimensional vector $\widetilde{w}$; then the turnover is usually modeled by the 1-norm of the weight change:

$$\|w - \widetilde{w}\|_1 = \sum_{i=1}^{d} |w_i - \widetilde{w}_i|$$

Thus the mean-variance problem with transaction cost can be rewritten as:

$$\max_{w} w'\mu - \frac{\lambda}{2} w'\Sigma w - c\|w - \widetilde{w}\|_1$$
$$\text{subject to:} \, e'w = 1$$

where $c > 0$ is the degree of the turnover. The object function is neither quadratic nor smooth at the point $\widetilde{w}$. But we are able to convert it to a quadratic programming problem:

$$\max_{v} v'\tilde{\mu} - \frac{\lambda}{2} v'\tilde{\Sigma} v$$
$$\text{subject to:} \, \tilde{e}'v = 0, v \geq 0$$

where

$$\tilde{\mu} = \begin{pmatrix} \mu - \lambda\widetilde{w}'\Sigma + ce \\ -\mu + \lambda\widetilde{w}'\Sigma + ce \end{pmatrix},$$
$$\tilde{\Sigma} = \begin{pmatrix} \Sigma & -\Sigma \\ -\Sigma & \Sigma \end{pmatrix},$$

and $\tilde{e}$ is a $2d$ dimensional vector with first d elements are 1 and the rest are -1. The optimal portfolio weight w^* of the mean-variance problem with transaction cost can be represented by the optimal solution of the above problem v^*:

$$w^* = \widetilde{w} + [I, -I]v^*$$

where I is the d-dimensional identity matrix. One can show that the first d elements of v^* are the positive parts of the weight change, and the rest d elements are the negative parts of the weight change. If v_k^* >0 for some k=$1, \dots, d$, then we must have $v_{d+k}^* = 0$, otherwise v^* will not be the optimal solution. The quadratic programming has been thoroughly studied in modern convex optimization theory. Classical algorithm includes the interior-point method and trust-region method, see (Nocedal and Wright 2006). Note that $\tilde{\Sigma}$ is not of full rank, this is caused by the non-smoothness of the original problem. One may shrink the eigenvalues of $\tilde{\Sigma}$ a bit to make the problem strictly convex. Thus in practice we usually get an suboptimal solution w^*. If the value of the object function on w^* does not exceed $\widetilde{w}'\mu - \frac{\lambda}{2}\widetilde{w}'\Sigma\widetilde{w}$ then we will keep the portfolio unchanged since the potential benefit of changing the portfolio does not cover the transaction cost.

5.2 On-line Portfolio Selection

In this section we consider a portfolio allocation framework that is different from the Markowitz's theory. Let $r_{i,t}$ where $i = 1,2,\ldots,d, t = 1,2,\ldots,T$ be the log return of the i-th asset at time t, $x_{i,t} = \exp(r_{i,t})$ be the price ratio, $x_t = (x_{1,t},\ldots,x_{d,t})'$ be the price ratio vector of d assets and $w_t = (w_{1,t},\ldots,w_{d,t})'$ be the portfolio weights. We assume that the portfolio is long-only; and let $\mathcal{W} = \{w \in \mathbb{R}^d \ s.t. \sum_{i=1}^{d} w_i = 1, w_i \geq 0\}$ be the universe of all long-only portfolio weights. Suppose that the initial wealth is S_0, then the value of a portfolio with strategies: $w_1, w_2, \ldots, w_t \in \mathcal{W}$ is given by:

$$S_t(w_1,\ldots,w_t|x_1,\ldots,x_t) = S_0 \prod_{s=1}^{t}\sum_{i=1}^{d} w_{i,s}x_{i,s}$$

A general on-line portfolio selection framework proposed by (Li and Hoi 2014) is as follows:

```
ALGORITHM: On-line portfolio selection
Input: $x_1,\ldots,x_T$: Historical market sequence
Output: $S_T$: Final cumulative wealth
    Initialize $S_0$ and $w_0$
for $t = 1,\ldots,T$ do
    Portfolio manager computes a portfolio $w_t$;
    Market reveals the market price ratio $x_t$;
    Updates cumulative wealth $S_t = S_{t-1}w_t{}'x_t$;
    Portfolio manager updates his/her online portfolio selection rules;
end
```

Here are several examples of on-line portfolio strategies:

5.2.1 Buy and hold strategy

The buy and hold strategy simply does not trade anymore once the initial portfolio weight w_0 is given. The dynamic of its portfolio weight is given by:

$$w_{i,t} = \frac{w_{i,t-1}x_{i,t-1}}{\sum_{j=1}^{d} w_{j,t-1}x_{j,t-1}}$$

and the cumulative wealth is:

$$S_t(w_1, \dots, w_t | x_1, \dots, x_t) = S_0 \sum_{i=1}^{d} w_{i,0} \prod_{s=1}^{t} x_{i,s}$$

5.2.2 Constantly rebalanced strategy

In contrast to the buy and hold strategy, the constantly rebalanced strategy is to keep rebalancing the portfolio such that $w_0 = w_1 = \cdots = w_t$. Thus the cumulative wealth is:

$$S_t(w_1, \dots, w_t | x_1, \dots, x_t) \ = S_0 \prod_{s=1}^{t} \sum_{i=1}^{d} w_{i,0}x_{i,s}$$

It can used to replicate the movements of a certain market index. Constantly rebalance and buy and hold are two naïve trading strategies that are often used as benchmarks.

5.2.3 Minimax strategy

Let $y_1, \dots, y_T$ be a sequence of integers ranged from 1 to d. Given a sequence of static strategies: $v_1, \dots, v_T \in \mathcal{W}$, i.e. v_t does not depend on any information prior to t. Then we can define a probability density function of $y_1, \dots, y_T$:

$$p_T(y_1, \dots, y_T) = \frac{\sup\limits_{v_1,\dots,v_T \in \mathcal{W}} \prod_{t=1}^{T} v_{y_t,t}}{\sum_{z_1=1}^{d} \cdots \sum_{z_T=1}^{d} \sup\limits_{v \in \mathcal{W}} \prod_{t=1}^{T} v_{z_t,t}}$$

The marginal density function of $y_1, \dots, y_t$ for some $t < T$ is given by:

$$p_t(y_1, \dots, y_t) = \sum_{z_{t+1}=1}^{d} \cdots \sum_{z_T=1}^{d} p_T(y_1, \cdots, y_t, z_{t+1}, \dots, z_T)\,,$$

Given a sequence of price ratio $x_1, \dots, x_{t-1}$, the minimax strategy on t is defined as:

$$w_{i,t} = \frac{\sum_{y_1=1}^{d} \cdots \sum_{y_{t-1}=1}^{d} p_t(y_1, \dots, y_{t-1}, i) \prod_{s=1}^{t-1} x_{y_s,s}}{\sum_{y_1=1}^{d} \cdots \sum_{y_{t-1}=1}^{d} p_{t-1}(y_1, \dots, y_{t-1}) \prod_{s=1}^{t-1} x_{y_s,s}},$$

The minimax strategy is the theoretical best strategy in terms of minimizing the worst-case logarithmic wealth ratio:

$$\sup_{x_1,\dots,x_T} \sup_{v_1,\dots,v_T \in \mathcal{V}} \log \frac{S_T(v_1, \dots, v_T | x_1, \dots, x_T)}{S_T(w_1, \dots, w_T | x_1, \dots, x_T)}$$

This ratio measures the difference between the strategy $w_1, \dots, w_T$ and the best static strategy with the knowledge of future under the worst case scenario. For detailed proof and the deduction of the minimax strategy we refer readers to (Cesa-Bianchi and Lugosi 2006).

5.2.4 Universal portfolio strategy

The minimax strategy is the theoretical best on-line strategy, but it is hard to achieve in practice. The computation of the densities $p_1, \dots, p_T$ is often numerically intractable in real market. (Cover 1991) proposes a computationally efficient strategy called universal portfolio:

$$w_{i,t} = \frac{\int_w u_j S_{t-1}(u, \dots, u|x_1, \dots, x_{t-1})\mu(u)du}{\int_w S_{t-1}(u, \dots, u|x_1, \dots, x_{t-1})\mu(u)du}$$

where $S_{t-1}(u, \dots, u|x_1, \dots, x_{t-1})$ is the cumulative wealth of a constantly rebalanced strategy u; and $\mu(u)$ is a density function that can be viewed as a prior distribution of the portfolio weight. At time t the strategy updates the distribution of weight based on the performance of all possible constantly rebalanced strategies. The new strategy is just the expectation of the updated distribution. (Cover and Ordentlich 1996) show that the worst-case logarithmic wealth ratio of the universal portfolio strategy has an upper bound that increases at the speed of $O(\log T)$ as T increases.

5.2.5 Exponential gradient (EG) strategy

The universal portfolio strategy is more practical than the minimax strategy, but still computationally intractable under high dimension; since it involves the calculation of d dimensional integrals. A simple strategy called the EG strategy proposed by (Helmbold, Schapire et al. 1998) updates the portfolio weights as follows:

$$w_{i,t} = \frac{w_{i,t-1} \exp\left(\frac{\eta x_{i,t-1}}{\sum_{i=1}^{d} w_{i,t-1} x_{i,t-1}}\right)}{\sum_{j=1}^{d} w_{j,t-1} \exp\left(\frac{\eta x_{j,t-1}}{\sum_{i=1}^{d} w_{i,t-1} x_{i,t-1}}\right)}$$

The EG strategy is a gradient-based forecaster since the term $x_{i,t-1}/\sum_{i=1}^{d} w_{i,t-1} x_{i,t-1}$ can be viewed as the gradient of logarithmic loss $-\log \sum_{i=1}^{d} w_{i,t-1} x_{i,t-1}$. The upper bound of the worst-case logarithmic wealth ratio of the EG strategy grows with $O(\sqrt{T})$; but in terms of the dimension d it grows only with $O(\sqrt{\log d})$ comparing to the linear growth of universal portfolio.

The above on-line strategies are all based on the assumption that there is no transaction cost. (Györfi and Vajda 2008) propose an on-line portfolio allocation framework with transaction costs. Suppose that at time $t-1$ the net wealth of the portfolio is given by N_{t-1}. Given a new strategy w_t and price ratio x_t the gross wealth at time t is given by:

$$S_t = N_{t-1} \sum_{i=1}^{d} w_{i,t} x_{i,t}$$

However, after the rebalancing, the wealth is reduced to $N_t \leq S_t$ because of the transaction costs. Before the rebalancing the weights of each asset are given by:

$$\widetilde{w}_{i,t} = \frac{w_{i,t} x_{i,t}}{\sum_{j=1}^{d} w_{j,t} x_{j,t}}\ , i = 1, \dots, d.$$

In the previous section we simply use $\|w_{t+1} - \widetilde{w}_t\|_1$ to approximate the transaction cost. A more precise approximation should be:

$$C_t = c_s \sum_{i=1}^{d} \max\{\widetilde{w}_{i,t} S_t - w_{i,t+1} N_t, 0\} + c_b \sum_{i=1}^{d} \max\{w_{i,t+1} N_t - \widetilde{w}_{i,t} S_t, 0\}$$

where c_s and c_b are the per dollar transaction costs of selling and buying respectively. Using the fact that $N_t = S_t - C_t$ we obtain the following equation:

$$1 = \rho_t + c_s \sum_{i=1}^{d} \max\{\widehat{w}_{i,t} - w_{i,t+1} \rho_t, 0\} + c_b \sum_{i=1}^{d} \max\{w_{i,t+1} \rho_t - \widehat{w}_{i,t}, 0\}$$

from with we can solve $\rho_t = N_t / S_t$. Thus instead of S_t we obtain a sequence of net wealth:

$$N_t = N_0 \prod_{s=1}^{t} \rho_s \sum_{i=1}^{d} w_{i,s} x_{i,s}$$

The on-line portfolio allocation with transaction costs can be summarized as:

ALGORITHM: On-line portfolio selection with transaction costs

Input: $x_1, \dots, x_T$: Historical market sequence, transaction costs c_b and c_s
Output: N_T: Final cumulative net wealth

```
  Initialize $\rho_0, S_0$ and $w_0$
for $t = 1,2,\dots,T$ do
  Portfolio manager computes a portfolio $w_t$;
  Updates the net wealth $N_{t-1} = \rho_{t-1}S_{t-1}$ after rebalancing;
  Market reveals the market price ratio $x_t$;
  Updates the gross wealth $S_t = N_{t-1}w_t'x_t$;
  Portfolio manager updates his/her online portfolio selection rules;
end
```

For more on-line portfolio selection strategies we refer readers to (Li and Hoi 2014) that provide a review of recent published techniques including some pattern recognition and machine learning strategies.

6. The Future

The rise of big data in financial industry has already been dramatic in the past decade. However we have good reason to believe that it is just a start; and the adoption of big data technology together with quantitative tools still has a long way to go. Despite of the rapid growth of high frequency industry and systematic trading funds, a number of traditional financial businesses still live in the small data era. A lot of economic data that they collected are in weekly, monthly or even quarterly based. Financial analysts may spend several hours on small amount of fundamental data of a single firm; while a large percentage of the work could be done automatically by machine. In addition, there are also more hidden errors in the data that are very difficult to be detected manually, as the information from the data providers such as Bloomberg and Factset grow tremendously. Thus the chances of operational risk made by human analyst who does not have the support of advanced technology increases simultaneously.

The most widely used data analyze tool in many financial firms is Microsoft Excel together with Visual Basic for Applications (VBA), which is very inefficient to deal with large datasets. On the side, although there is a number of professional data analyzing technologies that can process big data in a great efficiency, most of them are not user-friendly and fail to provide a comprehensive visualization of the information for the financial professionals with little technological or mathematical background. Thus the future of big data in finance is likely to be more client-oriented and personalized. This requires a closer connection between the engineers, scientists, financers and bankers (Zhang, Yu et al. 2016).

Even in the rapid growing high frequency industry, the technology and theory is far from mature. A unified influential framework such as the classical Black Scholes theory is not discovered yet in high frequency finance. Here we list some potential research topics that might be crucial for the development of quantitative finance.

6.1 Advanced statistics and information theory

In contrast to the classical statistics based on unbiased statistics such as maximum likelihood estimator, biased estimators, shrinkage, Bayesian theory and prior information are getting more and more emphasis in modern statistics in finance. Financial data is highly noisy and inconsistent. And this property would just become more significant as the data size grows bigger. The behavior of financial market also changes over the time. For example, the pattern of some financial instruments is completely changed by the crisis on 2009. New phenomena like the flash crash appears as new technologies are introduced to the market. Simple models fail to capture these changes, and complicated advanced models usually introduce large estimation errors. That is the reason for which the biased estimators often have a better performance than the unbiased ones.

However introducing prior information naively could be dangerous. How to shrink the estimators of a distribution? What is the best Bayesian prior? What is the correct way to parameterize a model? All of which are challenging questions in practice. A tool that can address these problems is the information geometry developed by (Amari and Nagaoka 2007). By linking probability distributions to differential geometry one can get a better intuition of statistical models and tests. For example, (Choi and Mullhaupt 2015) investigate the linear time series model on Kähler manifold and construct a Bayesian prior superior than the traditional Jeffers' prior. Further researches in different financial econometrics can potentially improve the current models and statistic tests.

6.2 Combination of machine learning, game theory and statistics

Markowitz portfolio theory is insightful; but it is clearly not the best strategy that an investor can choice. Given a prediction model and a certain investment period, the theoretical best strategy is provided by dynamic programming, which is numerically unachievable in finance. Machine learning theory provides feasible algorithms that can approximate a dynamic programming strategy. Techniques such as deep learning achieved significant success in different areas such as Chess and Go recently. However unlike the board games, financial market exhibits strong uncertainty; and the information available to each participant is incomplete. Thus machine-learning theory based on modern statistics is necessary for decision making in finance. The on-line portfolio strategies introduced in section 5.2 are just simple examples of the theory. These strategies do not consider stylized facts like mean-reverting of the market, and ignore the transaction costs which are crucial in high frequency trading. Utilizing additional information and signals from the market is an open topic in this area, (Li and Hoi 2014).

In addition high frequency industry is highly competitive. Buying and selling assets in a short amount of time is approximately a zero-sum game, i.e. someone's gain leads to someone's loss. Even for the low frequency investors the high frequency traders introduce higher transaction costs that can affect on the long-term profit. Thus an investor may consider opponents' actions and the impact of his strategy to the market before executing his strategy. Thus game theory may provide a deeper insight to the high frequency trading than the dynamic programming of a certain utility function.

6.3 Efficient algorithms in linear algebra and convex optimization

Linear algebra and convex optimization are the footstones for modern data analysis and financial engineering. Any quantitative model in finance would not be practical without basic tools in linear algebra and optimization, such as matrix inversion, SVD, Cholesky decomposition, QR decomposition, eigenvalue problem, linear and quadratic programming. While most classical algorithms in linear algebra and convex optimization were well developed in the last century, the need of faster and more accurate algorithms keeps increasing as new technologies and new applications appear. Frist, a number of matrices in financial applications are sparse or structured. Thus algorithms specificity designed for these matrices can be more efficient than these standard approaches. Second of all, the novel heterogeneous platforms including GPU and MIC (Zhang and Gao 2015) has further escalated the computational complexities, although they have improved the computing performance.

7. Conclusion

In this chapter we review the big data concept in quantitative finance. By considering high frequency data as an example, we introduce the basic data cleaning and aggregation approaches, quantitative modeling, portfolio allocation and strategies, which are summarized by Figure 7.

The inverted pyramid structure illustrated the change of data size after each step. The three topics are also related to the 3'Vs in Big Data. First of all, raw data is voluminous. Processing and cleaning them requires efficient I/O, ranking and searching techniques. Second, we briefly introduce the typical econometric models but there exist a variety of quantitative models with different degrees of complexity. Different matrix operating and optimization algorithms are needed to deal with different types of the models. Finally, the velocity of model estimation and portfolio allocation is equally important for algorithm trading firms. Even milliseconds' difference in speed could make a huge difference for some high frequency investors. However the framework in Figure 7 is just a coarse summarization of the world of quantitative finance. More researches in market microstructure would be launched in the near future, as more types of data get involved. Appearance of the next Black Scholes theory is just a matter of time.

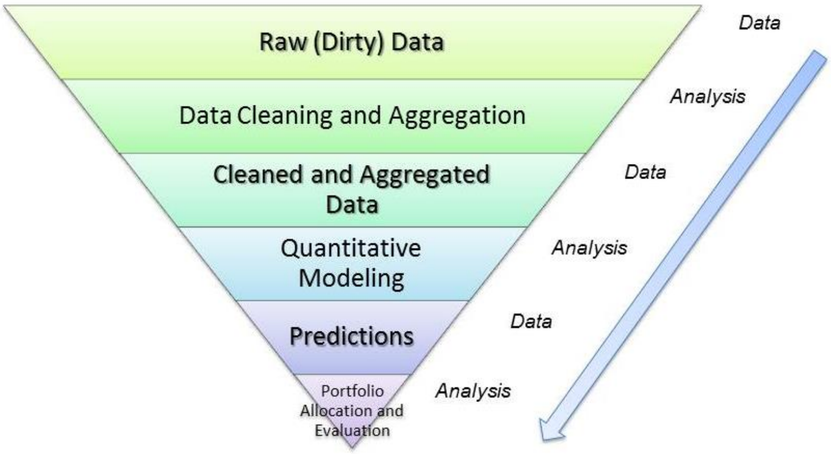

Raw (Dirty) Data
Data Cleaning and Aggregation
Cleaned and Aggregated Data
Quantitative Modeling
Predictions
Portfolio Allocation and Evaluation
Data
Analysis
Data
Analysis
Data
Analysis

Figure 7: Inverted Pyramid Structure of Quantitative Data Analysis in Finance

8. References

Aldridge, I. (2009). High-frequency trading: a practical guide to algorithmic strategies and trading systems, John Wiley and Sons.

Aldridge, I. (2015). "Trends: all finance will soon be big data finance." from http://www.huffingtonpost.com/irene-aldridge/trends-all-finance-will-s_b_6613138.html.

Amari, S.-i. and H. Nagaoka (2007). Methods of information geometry, American Mathematical Society.

Andersen, T. G. and T. Bollerslev (1997). "Intraday periodicity and volatility persistence in financial markets." Journal of empirical finance **4**(2): 115-158.

Andersen, T. G., T. Bollerslev, et al. (2000). "Intraday and interday volatility in the Japanese stock market." Journal of International Financial Markets, Institutions and Money **10**(2): 107-130.

Beck, A., Y. S. A. Kim, et al. (2013). "Empirical analysis of ARMA-GARCH models in market risk estimation on high-frequency US data." Studies in Nonlinear Dynamics and Econometrics **17**(2): 167-177.

Black, F. and M. Scholes (1973). "The pricing of options and corporate liabilities." The journal of political economy: 637-654.

Bollerslev, T. (1986). "Generalized autoregressive conditional heteroskedasticity." Journal of econometrics **31**(3): 307-327.

Brownlees, C. T. and G. M. Gallo (2006). "Financial econometric analysis at ultra-high frequency: Data handling concerns." Computational Statistics & Data Analysis **51**(4): 2232-2245.

Cesa-Bianchi, N. and G. Lugosi (2006). Prediction, learning, and games, Cambridge university press.

Chekhlov, A., S. P. Uryasev, et al. (2000). Portfolio optimization with drawdown constraints. Research Report #2000-5. Available at SSRN: http://dx.doi.org/10.2139/ssrn.223323.

Choi, J. and A. P. Mullhaupt (2015). "Geometric shrinkage priors for Kählerian signal filters." Entropy **17**(3): 1347-1357.

Cover, T. M. (1991). "Universal portfolios." Mathematical finance **1**(1): 1-29.

Cover, T. M. and E. Ordentlich (1996). "Universal portfolios with side information." Information Theory, IEEE Transactions on **42**(2): 348-363.

Cox, J. C. and S. A. Ross (1976). "The valuation of options for alternative stochastic processes." Journal of financial economics **3**(1-2): 145-166.

Cox, J. C., S. A. Ross, et al. (1979). "Option pricing: A simplified approach." Journal of financial Economics **7**(3): 229-263.

Diamond, D. W. and R. E. Verrecchia (1987). "Constraints on short-selling and asset price adjustment to private information." Journal of Financial Economics **18**(2): 277-311.

Dong, X. (2013). New development on market microstructure and macrostructure: Patterns of US high frequency data and a unified factor model framework. PhD Dissertation, STATE UNIVERSITY OF NEW YORK AT STONY BROOK.

Duffie, D. (2010). Dynamic asset pricing theory, Princeton University Press.

Dufour, A. and R. F. Engle (2000). "Time and the price impact of a trade." The Journal of Finance **55**(6): 2467-2498.

Easley, D. and M. O'hara (1992). "Time and the process of security price adjustment." The Journal of finance **47**(2): 577-605.

Engle, R. F. (1982). "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation." Econometrica: Journal of the Econometric Society: 987-1007.

Engle, R. F. (2000). "The econometrics of ultra‐high‐frequency data." Econometrica **68**(1): 1-22.

Engle, R. F. and S. Manganelli (2004). "CAViaR: Conditional autoregressive value at risk by regression quantiles." Journal of Business & Economic Statistics **22**(4): 367-381.

Engle, R. F. and J. R. Russell (1998). "Autoregressive conditional duration: a new model for irregularly spaced transaction data." Econometrica: 1127-1162.

Fang, B. and P. Zhang (2016). Big Data in Finance. Big Data Concepts, Theories, and Applications. S. Yu and S. Guo. Cham, Springer International Publishing**:** 391-412.

Gençay, R., M. Dacorogna, et al. (2001). An introduction to high-frequency finance, Academic press.

Györfi, L. and I. Vajda (2008). Growth optimal investment with transaction costs. Algorithmic Learning Theory, Springer.

Harrison, J. M. and D. M. Kreps (1979). "Martingales and arbitrage in multiperiod securities markets." Journal of Economic theory **20**(3): 381-408.

Helmbold, D. P., R. E. Schapire, et al. (1998). "On‐Line Portfolio Selection Using Multiplicative Updates." Mathematical Finance **8**(4): 325-347.

Jia, T. (2013). Algorithms and structures for covariance estimates with application to finance. PhD Dissertation, STATE UNIVERSITY OF NEW YORK AT STONY BROOK.

Kim, Y. S. (2015). "Multivariate tempered stable model with long-range dependence and time-varying volatility." Frontiers in Applied Mathematics and Statistics **1**: 1.

Ledoit, O. and M. Wolf (2003). "Improved estimation of the covariance matrix of stock returns with an application to portfolio selection." Journal of empirical finance **10**(5): 603-621.

Li, B. and S. C. Hoi (2014). "Online portfolio selection: A survey." ACM Computing Surveys (CSUR) **46**(3): 35.

Lintner, J. (1965). "The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets." The review of economics and statistics: 13-37.

Liu, C. and D. B. Rubin (1994). "The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence." Biometrika **81**(4): 633-648.

Markowitz, H. (1952). "Portfolio selection." The journal of finance **7**(1): 77-91.

Marple Jr, S. L. (1987). "Digital spectral analysis with applications." Englewood Cliffs, NJ, Prentice-Hall, Inc., 1987, 512 p. **1**.

Matsuyama, Y. (2003). "The α-EM algorithm: Surrogate likelihood maximization using α-logarithmic information measures." Information Theory, IEEE Transactions on **49**(3): 692-706.

McNeil, A. J., R. Frey, et al. (2005). Quantitative risk management: Concepts, techniques and tools, Princeton university press.

Meng, X.-L. and D. B. Rubin (1993). "Maximum likelihood estimation via the ECM algorithm: A general framework." Biometrika **80**(2): 267-278.

Merton, R. C. (1969). "Lifetime portfolio selection under uncertainty: The continuous-time case." The review of Economics and Statistics: 247-257.

Meucci, A. (2011). "'P'Versus' Q': Differences and Commonalities between the Two Areas of Quantitative Finance." GARP Risk Professional: 47-50.

Mineo, A. M. and F. Romito (2007). A Method to 'Clean Up' Ultra High-Frequency Data, Vita e pensiero.

Mineo, A. M. and F. Romito (2008). "Different Methods to Clean Up Ultra High-Frequency Data." Atti della XLIV Riunione Scientifica della Societa'Italiana di Statistica.

Mossin, J. (1966). "Equilibrium in a capital asset market." Econometrica: Journal of the econometric society: 768-783.

Mullhaupt, A. P. and K. S. Riedel (1998). Band Matrix Representation of Triangular Input Balanced Form. IEEE Transactions on Automatic Control.

Neal, R. M. and G. E. Hinton (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in graphical models, Springer**:** 355-368.

Nocedal, J. and S. Wright (2006). Numerical optimization, Springer Science & Business Media.

Rachev, S. T., S. Mittnik, et al. (2007). Financial econometrics: from basics to advanced modeling techniques, John Wiley & Sons.

Rockafellar, R. T. and S. Uryasev (2000). "Optimization of conditional value-at-risk." Journal of risk **2**: 21-42.

Russell, J. R., R. Engle, et al. (2009). "Analysis of high-frequency data." Handbook of financial econometrics **1**: 383-426.

Sharpe, W. F. (1964). "Capital asset prices: A theory of market equilibrium under conditions of risk." The journal of finance **19**(3): 425-442.

Shi, X. and A. Kim (2015). "Coherent Risk Measure and Normal Mixture Distributions with Application in Portfolio Optimization and Risk Allocation." Available at SSRN http://dx.doi.org/10.2139/ssrn.2548057.

Sun, W., S. Z. Rachev, et al. (2008). Long-range dependence, fractal processes, and intra-daily data. Handbook on Information Technology in Finance, Springer**:** 543-585.

Tomov, S., R. Nath, et al. (2010). Dense linear algebra solvers for multicore with GPU accelerators. Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on, IEEE.

Treynor, J. L. (1961). "Toward a theory of market value of risky assets." Available at SSRN: http://dx.doi.org/10.2139/ssrn.628187.

Yan, Y. (2007). Introduction to TAQ. WRDS Users Conference Presentation.

Zhang, P. and Y. Gao (2015). Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations. High Performance Computing: 30th International Conference, ISC High Performance 2015, Frankfurt, Germany, July 12-16, 2015, Proceedings. M. J. Kunkel and T. Ludwig. Cham, Springer International Publishing**:** 17-30.

Zhang, P., Y. Gao, et al. (2015). A Data-Oriented Method for Scheduling Dependent Tasks on High-Density Multi-GPU Systems. High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on.

Zhang, P., L. Liu, et al. (2015). "A Data-Driven Paradigm for Mapping Problems." Parallel Computing **48**: 108-124.

Zhang, P., K. Yu, et al. (2016). "QuantCloud: Big Data Infrastructure for Quantitative Finance on the Cloud." IEEE Transactions on Big Data.