Learning Time Series Data using Cross Correlation and Its Application in Bitcoin Price Prediction

by

Kang Zhang

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

May, 2014

[June 2014]

Author: Signature redacted

Department of Electrical Engineering and Computer Science

May 23, 2014

Certified by: Signature redacted

Professor Devavrat Shah, Thesis Supervisor

Signature redacted May 23, 2014

Accepted by:

Prof. Albert R. Meyer, Chairman, Masters of Engineering Thesis Committee

Learning Time Series Data using Cross Correlation and Its Application in Bitcoin Price Prediction

by

Kang Zhang

Submitted to the Department of Electrical Engineering and Computer Science

June, 2014

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this work, we developed an quantitative trading algorithm for bitcoin that is shown to be profitable. The algorithm establishes a framework that combines parametric variables and non-parametric variables in a logistical regression model, capturing information in both the static states and the evolution of states. The combination improves the performance of the strategy. In addition, we demonstrated that we can discovery curve similarity of time series using cross correlation and L2 distance. The similarity metrics can be efficiently computed using convolution and can help us learn from the past instance using an ensemble voting scheme.

Acknowledgements

I would like to express my sincere gratitude to my adviser Devavrat Shah, for his guidance and advice, which extends beyond this thesis into my career and life. Throughout the past year, I have learned so much from his class and numerous discussions with him. Every time I talk to him, I always feel inspired with infectious new ideas. He made my final year at MIT a much more significant part of my education experience.

A big thank you to our administrative staff Lynn Dell, who helped a lot with the administrative and logistical issues in this project, particularly the purchase of a great server.

I would also like to thank my officemate Dhruv Parthasarathy and Kuang Xu for being such wonderful friends and colleagues. I had really good times with them while working on my thesis. Certainly I really appreciate them letting me use such a loud computer in the office.

Thank you to my friends, for everything you have done for me; to my family and my girlfriend, for being always with me and proud of me.

Contents

Chapter 1

Introduction

1.1 Motivation

Time series data prediction has wide applications in the real world, ranging from the trading of financial instruments to signal processing. Time series predictions often involve noisy, multi-dimensional and structured and/or semi-structured datasets, making it difficult to combine different features to extract maximal amount of information from data. Furthermore, many predictive analyses are based on certain ad hoc models of the application, such as stochastic models, which requires extensive domain knowledge. Thus, this thesis aims to create a framework for time series prediction to address these two challenges. Specially, we will apply our original approach to the quantitative trading of Bitcoin, a virtual cryptographic currency.

1.2 Bitcoin and Quantitative Trading

Bitcoin is a digital currency based on a peer-to-peer payment protocol introduced in 2009 by developer Satoshi Nakamoto. The protocol uses a public key cryptography to allow users transfer digital currency on a decentralized network. Distributed servers collectively verify and process bitcoin transactions.

Since its inception, bitcoin rapidly gained popularity worldwide. Many internet companies, such as Zynga, Overstock, accepts bitcoin payment for their purchase of goods and services. This prompts the creation of a new financial system that facilities the use of bitcoin. One major component is exchange platforms that allow users to buy or sell bitcoin for fiat currency with other users. On an exchange, a user can post bitcoin orders, which consists of the price (in fiat currency) at which he or she willing to buy or sell bitcoins, and the amount of bitcoin they are willing to buy or sell at that price. If another user wish to trade at that price, a transaction between the two users will be made by the exchange and published to everyone else in real time. There is always a difference between the highest ask (price at which people are willing to sell) and the bid (price at which people are willing to buy). This difference is known as the spread. If one buys and then immediately sells one bitcoin, he or she will lose the amount of money equal to the spread.

Our goal is to create a profitable quantitative trading strategies by using time series prediction methods. We will try to predict the future price of bitcoin and buy if the price is likely to increase. This is a good application of time series prediction. Firstly, the market of bitcoin is complex; it involves multiple time series features including bid / ask prices, price and quantity of each trade and orders posted by users. Secondly, trading bitcoin is challenging; the many professionals are trading bitcoins and a strategy can be successful only if it is more sophisticated than other players' strategies. Thirdly, bitcoin market has the

most open access to data and a level playing field; unlike equity or future market, it has not been dominated by advanced traders with superior connection speed.

1.3 Previous Work

Latest quantitative trading strategies are usually kept secretive by trading companies. However, some traditional trading strategies are known. Mean reversion strategies assumes that that prices randomly fluctuate around the intrinsic value of the underlying product . Thus, if the price deviates from its moving average by one or two standard deviation, it should revert to the mean later. However, sometimes price movement is caused by an actual change in the value of the underlying product, in which case the mean reversion requires more sophisticated framework to determine the cause of the price movement.

Momentum strategies assumes that price movements usually continues in its direction of movement for some period of time. This is because of a psychological factor: when the price is increasing, investors, seeing the upward trend, tend to buy, which causes a further increase in price. The most common algorithm for determining the start of a trend is Moving Average Convergence Divergence (MACD). The MACD is calculated by subtracting the 26-day exponential moving average (EMA) of price from the 12-day EMA. A 9-day EMA of the MACD, called the "signal line", is then plotted on top of the MACD, functioning as a trigger for buy and sell signals. When the MACD falls below the signal line, it indicates a downward trend. Conversely, when the MACD rises above the signal line, it

indicates an upward trend, which suggests that the price is likely to experience upward momentum.

Pattern recognition based technical analysis, also known as charting, is another common strategy. It assumes the existence of support and resistance levels - prices at which there are many buyers and sellers respectively. For example, if the current price is $990, it will face support downward pressure to cross $1000. But once it crosses $1000, it is likely to continue to increase. Various patterns, frequently derived from intuition and experience, have been used to identified such support and resistance level. For example, Head and Shoulder (three peaks, with the middle peak higher than the other two) is a pattern which indicates that the price failed to cross the resistance level and is likely to fall.

1.4 Our Approach

Tradition trading strategies generally uses a set of ad hoc parameters to describe data and then make further predictions. However, simple parametric models requires large amount of domain knowledge and are often unable to capture some aspects of dataset. To resolve this, we propose a non-parametric method to learn and predict time series. We assume that the future will resemble the past. and we can predict the future by comparing the present state with an ensemble of past instances. Furthermore, if we have some knowledge about the system and believe some parametric indicators may be useful, it is possible to combine parametric indicators with non-parametric indicators to create a

stronger predictor. This allows us to make time series predictions with only a partially specified model.

Chapter 2

Data Collection

2.1 Data Crawling

We collect bitcoin market data from all major exchanges using their public Application Programmer Interface (API) or customized scripting tools, as shown in table 1.

Exchange	Country	Currency
796	Hong Kong	US dollar
bitfinex	Hong Kong	US dollar
bitstamp	United Kingdom	US dollar
btce	Bulgaria	US dollar
btcchina	China	Chinese Yuan
chbtc	China	Chinese Yuan
huobi	China	Chinese Yuan
okcoin	China	Chinese Yuan

Table 1. Bitcoin exchanges

From each exchange, we collect each bitcoin trade, which includes a timestamp, trade type (buy or sell), price, and quantity traded. Furthermore, we gather the lowest ask price and lowest bid price every 2 seconds. The crawler has been running on a dedicated server since Feb 2014 and has crawled 137 million data points with a size of over 300 gigabytes in three months.

2.2 Data Storage

Bitcoin price data are stored in a relational database using a MySQL server running on a 960 gigabyte solid state drive. Data from different exchanges are stored in separate tables. Rows are indexed by the timestamp for fast retrieval. Running such huge and complex requires periodical maintenance, during which the database may not be accessible. Thus, a script is added to monitor the health of the database. When the database fails, the crawler will save data in a temporary file, which will be dumped into the database when it becomes available again.

Chapter 3

Algorithm

3.1 Overview

Our Bitcoin trading algorithm consists of two phases, namely, signal phase and prediction phase, as shown in figure 1. In the signal phase, two moving averages of price with different window widths are monitored. Once the slow moving average crosses the fast moving average, a trading signal is fired and we proceed to the prediction phase. In the prediction phase, we attempt to predict the future price movement with a logistic regression classifier trained with historical data. We will execute a trade if a positive prediction is made.

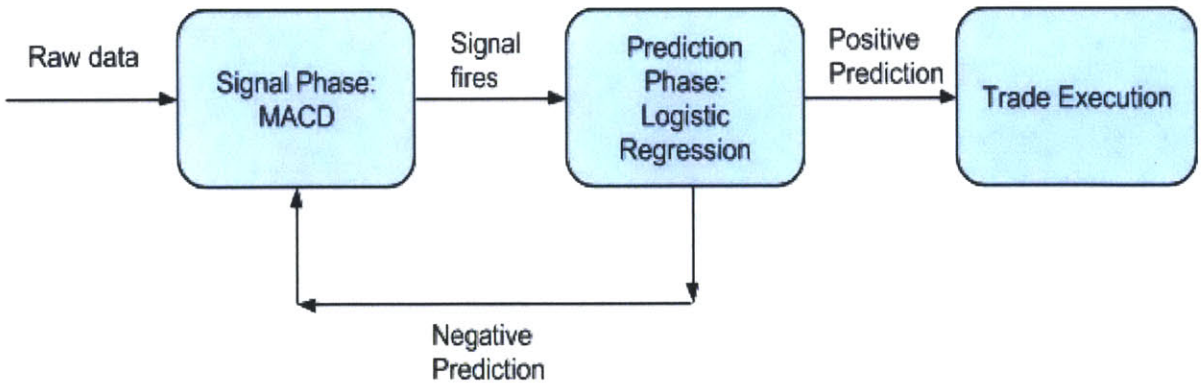


Figure 1. Overview of the algorithm

3.2 Preliminary Data Processing

Raw data points are acquired at regular time intervals, usually around two seconds. However, due to network latency and other sources of noise, the time intervals between data points can vary and data points may contain duplicates. For the convenience of data handling, a new time series with fixed time interval of 10 seconds is constructed. Each point in the new time series is mapped to the closest data point in the raw time series before the point, as shown in figure 2. This is to ensure the new time series obeys the causality constraint and do not have any information about the future. This procedure introduces a little delay, but it is insignificant compared to the trading timescale discussed later.

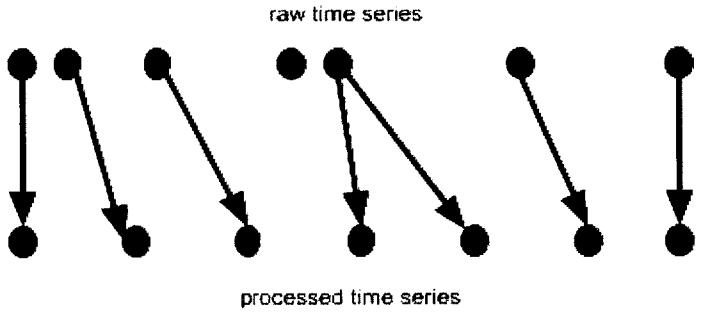


Figure 2. Mapping from the raw time series to a new time series

3.3 Signal Phase

In the signal phase, moving average convergence divergence is used. Two moving averages of window widths 7 minutes and 26 minutes are computed as $s_1 = m(p, 12)$ and

$s_2 = m(p, 26)$, where p is the time series of price and m is the moving average function given by $m(p, \Delta)[t] = \frac{1}{\Delta}\Sigma_0^{t-\Delta+1} p[i]$.

The signal s is the smoothed difference between the fast and slow moving average, mathematically, $s = m(s_1 - s_2, 9)$. The signal fires whenever s cross zero from negative to positive, indicating the reversal of the decreasing price movement.

3.4. Prediction Phase

3.4.1 Cross Correlation

One important aspect of the algorithm is non-parametric learning from historical data. Given a sequence of data points, we would like to discover similar sequences in the past. Normalized cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. Consider two sequences f and g of length n with mean and standard deviation being $(\overline{f}, \sigma_f), (\overline{g}, \sigma_g)$ respectively, their correlation metric is computed as

$$f * g = \frac{1}{n}\Sigma \frac{(f[i]-\overline{f})\cdot(g[i]-\overline{g})}{\sigma_f \sigma_g}$$

Now consider a pattern g of length m and a time series f of length n (n >> m). One can compute this cross correlation metric for every subsequence of size m in the time series, and output an array of size n. This array reflects which portion of the time series resembles the pattern.

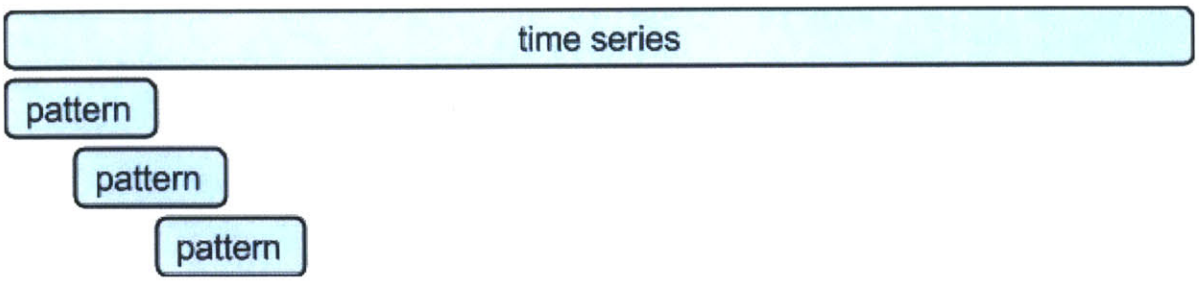


Figure 4. Comparing the pattern with a time series

Vectorized computation can evaluate this cross correlation array efficiently. The term $\Sigma f[i]g[i]$ is efficiently computed by the Numpy convolution library. The other terms are simply vectorized moving average and moving square average.

$$f * g = \frac{1}{n\sigma_f \sigma_g}(\Sigma f[i]g[i] - \bar{f}\Sigma g[i] - \bar{g}\Sigma f[i] + \bar{f}\bar{g})$$

3.4.2 L2 Distance

Another useful distance metric that measures how similar two sequences are is the standard L2 distance. Mathematically,

$$L2(f,g) = \frac{1}{n}\Sigma(f[i] - g[i] + \bar{f} - \bar{g})^2$$

Similarly upon expanding the quadratic term, this metric can be efficiently computed using convolution and vectorized moving average and moving square average.

In the test data, the anti correlation between cross correlation and L2 distance is as strong as -0.7, as shown in figure 5.

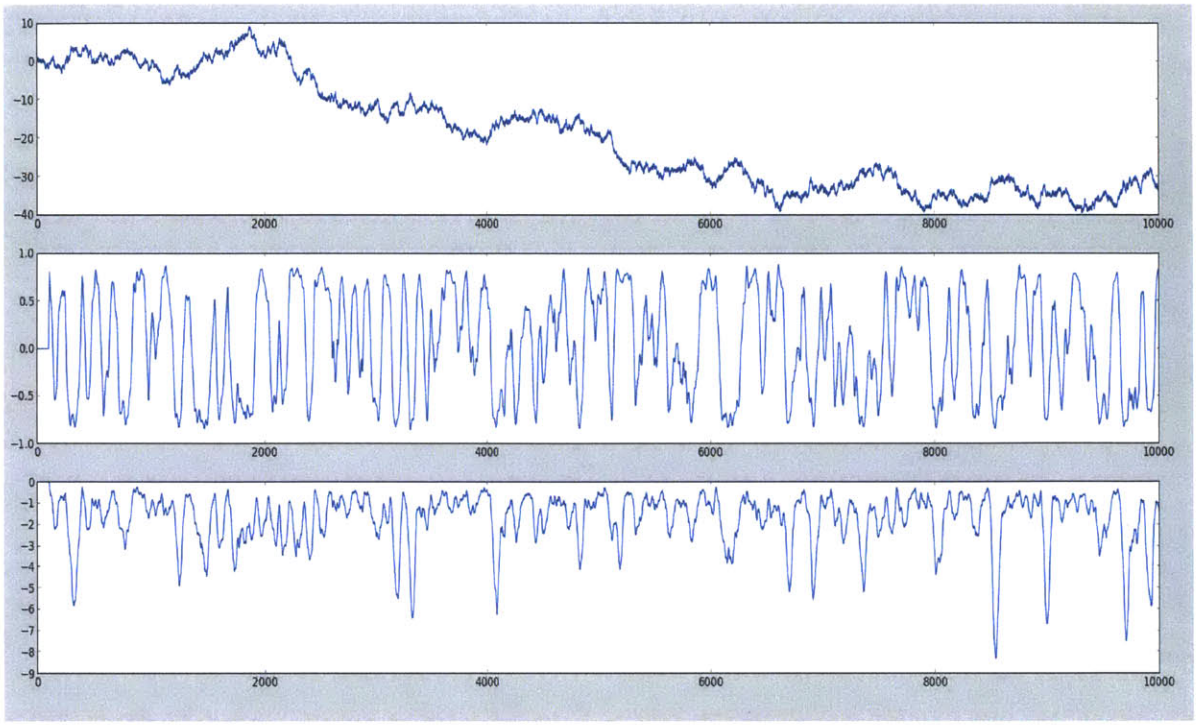


Figure 5. Top: original time series, middle: cross correlation between the pattern and each subsequence of the time series, bottom: L2 between the pattern and each subsequence of the time series

3.4.3 Non-parametric Learning of Time Series Data

Consider a pattern p of m data points and a time series x of n data points. Each m-length sub-sequence of x responds to a response variable y. We wish to predict the response corresponding to pattern p. In this specific application, x is the time series of bitcoin price and y is the price in two hours. Pattern p is is the last m points of x.

For each subsequence of x, we may compute its cross correlation with p, obtaining an correlation array. In this array, 100 highest local maxima are selected and their corresponding response values are extracted, as shown in figure 7. The predicted value for the pattern p is computed as the weighted average of these 100 samples, mathematically, $\Sigma w[i] \cdot y[i] / \Sigma w[i]$, where w is the cross correlation value.

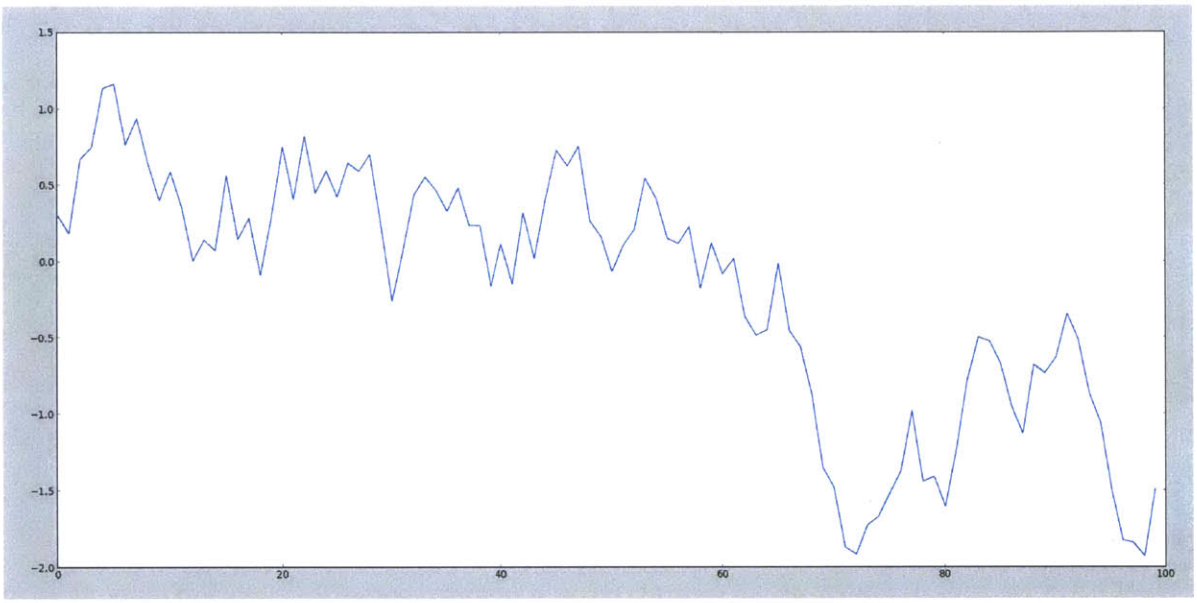


Figure 6. A randomly generated pattern

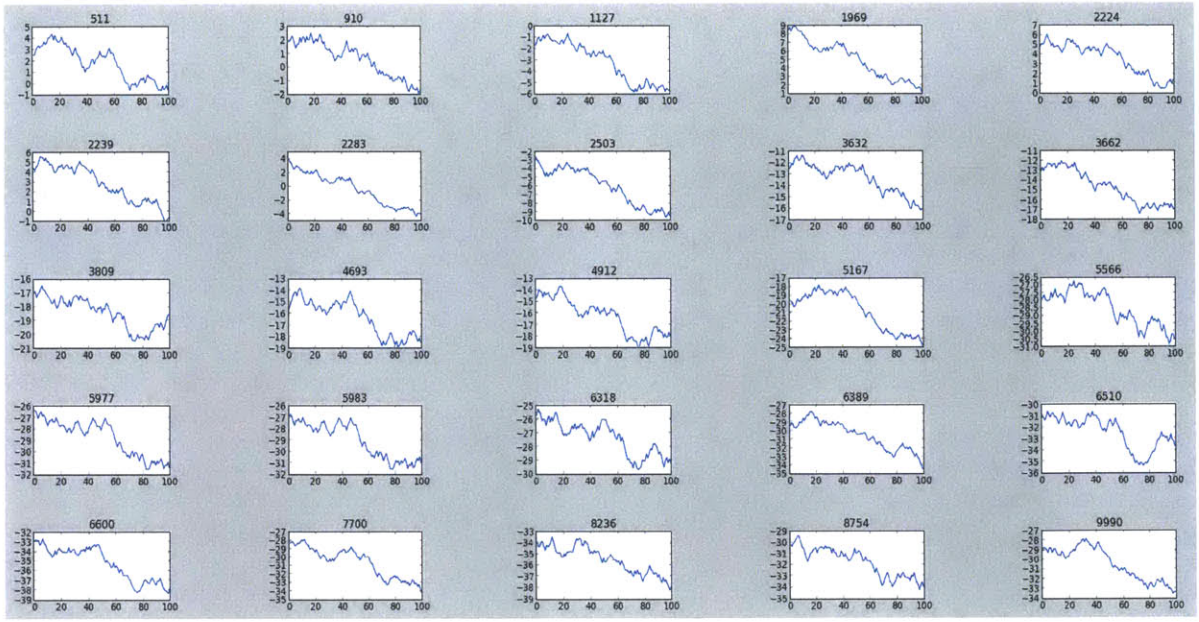


Figure 7. Similar patterns identified using cross correlation

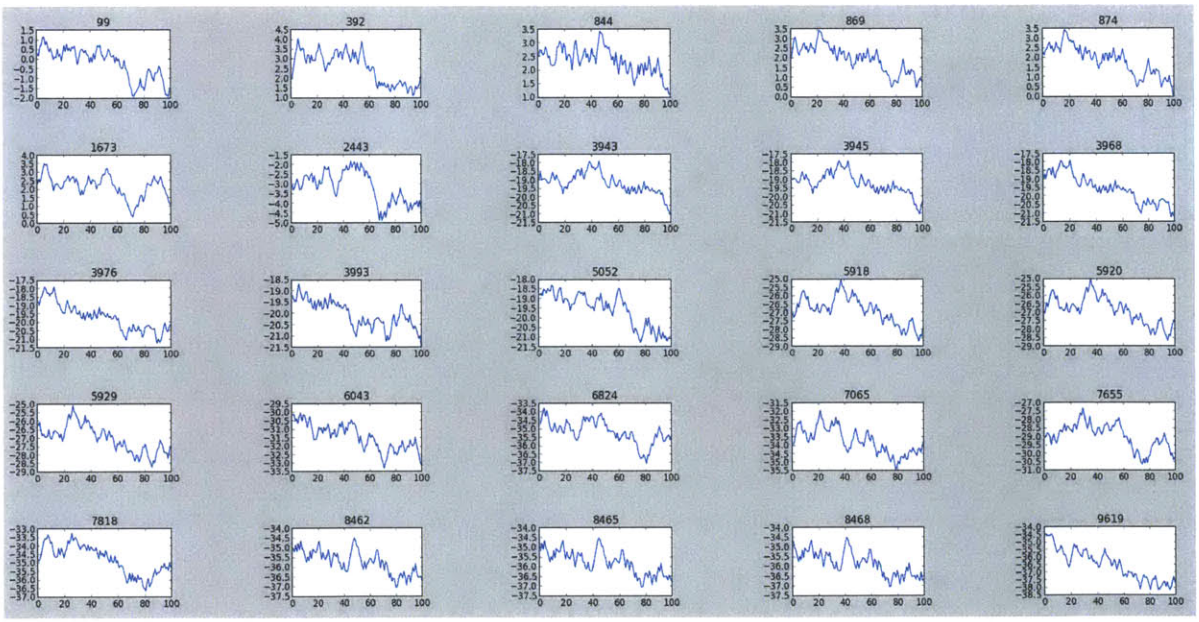


Figure 8. Similar patterns identified using L2 distance

Similarly, a prediction can be made using the L2 distance metric. However, 100 lowest local minima are selected, as shown in figure 8. The prediction is computed as $\Sigma(exp(w[i]) \cdot y[i])/\Sigma exp(w[i])$.

3.4.4 Prediction Algorithm

Once a signal fires, an ensemble of indicators are used to make a prediction. There are two types of indications, direct ones and indirect ones. Direct indicators include total trade volume, the number of trades and net trade volume. Indirect indicators are predictions from non-parametric learning using cross correlation and L2 distance. These forms a set of five features. A logistic regression model is trained to predict the direction of future price movement. The first 20% data are used to train non-parametric model, the next 40% data are used to train the logistic regression model and the rest are used to evaluate the performance.

3.5 Simulation

A simulation framework is set up to realistically test the strategy. Each trade will either buy or sell exactly one bitcoin at bid or ask respectively. The maximal position allowed is one bitcoin and there is no short selling. Once a bitcoin is bought, it is sold when the reverse signal fires, or in two hours, whichever earlier. Simulation is performed on bitcoin versus chinese yuan market and trade data collected from Feb 2014 to May 2014 from Okcoin exchange.

Chapter 4

Results and Discussion

4.1 Linearity of the Variance of Price Movements

The price movement of bitcoin resembles brownian motion. In each unit time window, the price movement seems to be independent of price movements in previous time windows. Let σ^2 be the variance of price change in one unit time. If the price movements in each unit time are independent, then the variance over k unit time will be $\Sigma_1^k \sigma^2 = k\sigma^2$. Thus we expect the variance increases linearly with respect to the length of our time windows if price movements are independent.

The variances of price for different time windows are measured and as shown below. The linearity suggests that bitcoin price movements are relatively independent, possibly due to a large random component. In other words, any predictive signal, if any, is likely to be hidden in idiosyncratic noise. This is a result of the competition among professional traders; any obvious signals have been utilized for their trading.

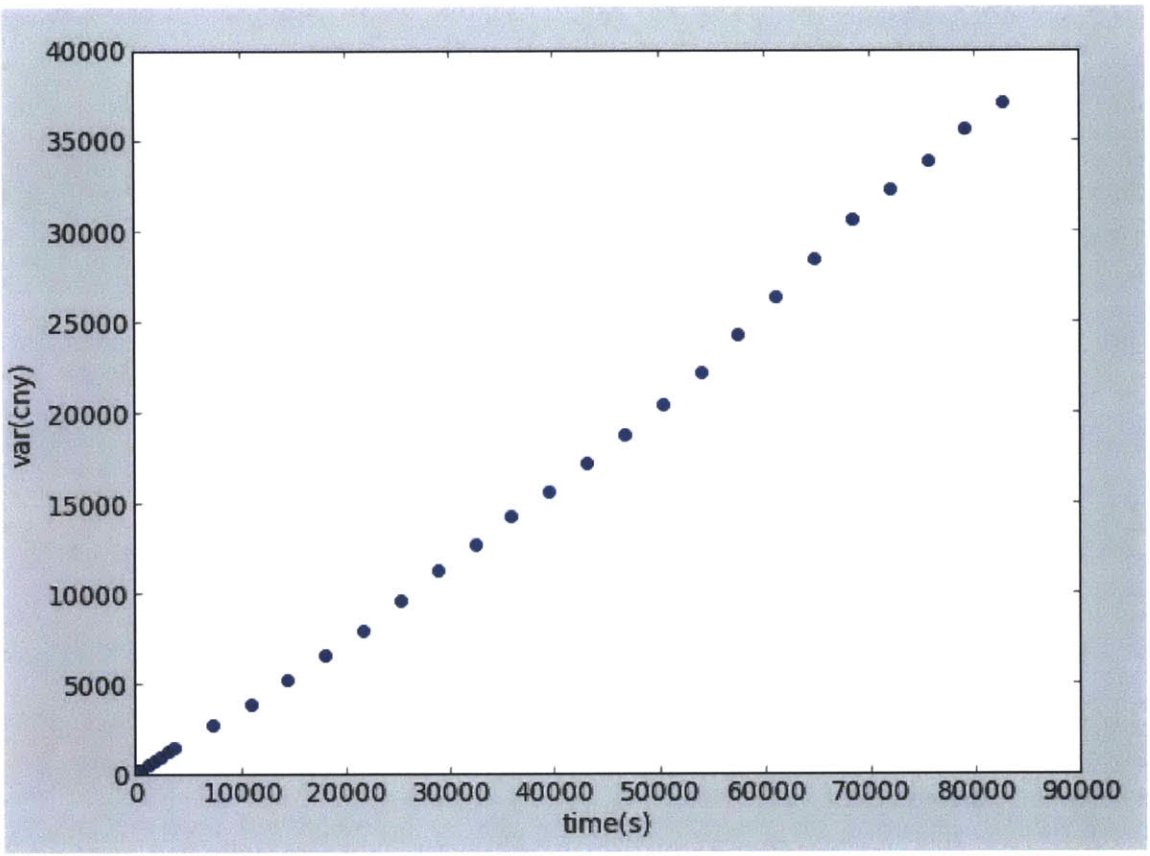


Figure 9. Linearity of the variance of price movements

4.2 Determination of Trading Frequency

In general, quantitative trading can operate in vastly different time horizons. Depending on the strategy, a position may be held for sub-seconds to days. In our specific case, it is desirable to have a short holding period and execute many trades. This reduces variance in our results and make the strategy more resilient against abrupt market crashes. However, the execution of a trade involves a cost, the bid-ask spread. This fixed cost is measured to be 3.3 yuan on average. Thus, a trade is profitable only if it makes more than

3.3 yuan. This can limit our holding period because the variance of price is directly proportional to the holding period. If the period is too short, it is unlikely for the price to change much. We can derive a feasible time horizon using some intuitive deduction. Suppose the standard deviation of price over a period of time is σ, the cost of trading is c, and our direction prediction accuracy is $0.5+\varepsilon$. The expected profit will be $2\varepsilon\sigma - c$. Given that ε could be less than 0.03 and c is approximately 3. The value of σ needs to be greater than 50. Thus, the standard deviation of price change needs to be as large as 30 yuan. Figure 10 shows the relationship between the standard deviation of price change and time period. The holding period is approximated to be 2 hours.

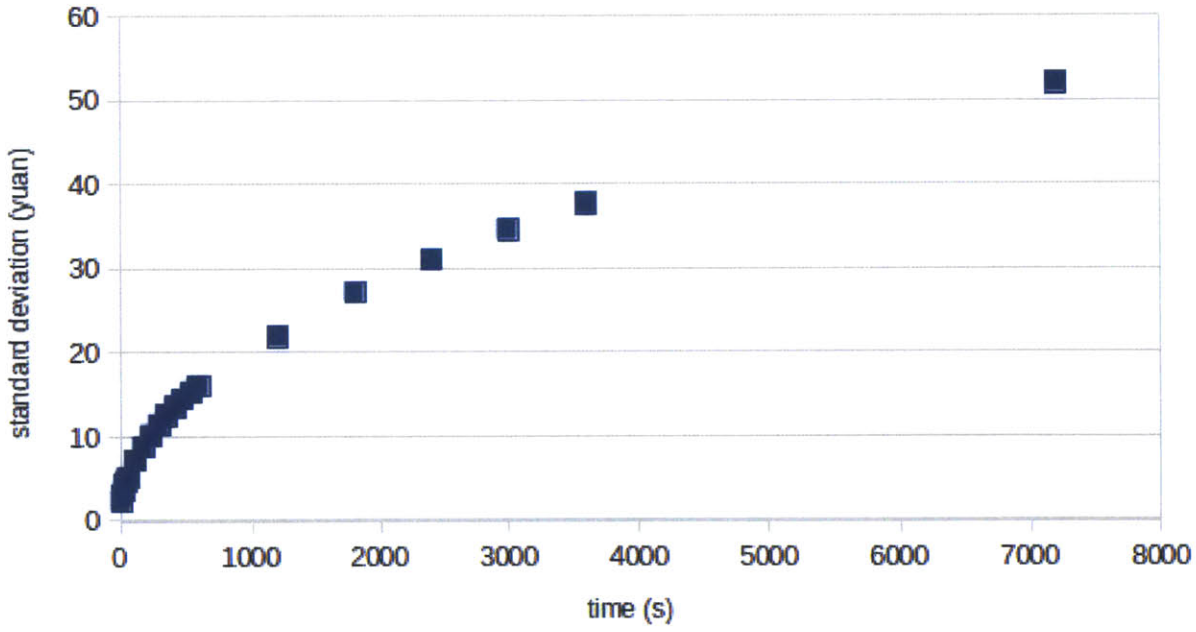


Figure 10. Relationship between the standard deviation of price change and time period

4.3 Simulation Results

During the simulation period, the price of bitcoin dropped by approximately 20%, from 3471 yuan to 2774 yuan. Thus, as a reference to the performance of our strategy, a naive buy-and-hold strategy will lose 697 yuan.

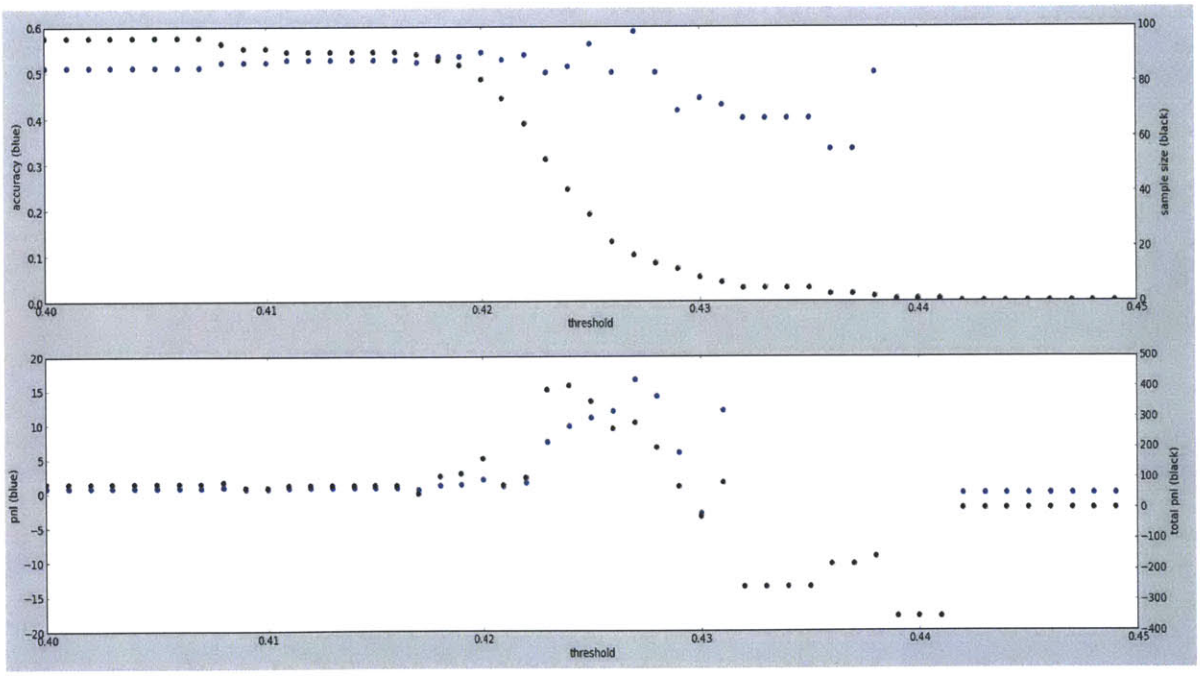
Strategies in the simulation perform much better than a naive strategy. If we simply use the MACD signal , i.e., trade whenever the signal fires, the strategy can break even after taking trading cost into consideration. However, if we apply the logistical regression model to each signal firing events, and selectively trade on them a substantial profit can be achieved. Table summarizes the performance of different strategies. For the purpose of comparison the logistic regression model is tuned to filter half of all fire events.

Strategy	Number of trades	Total Profit	Average Profit
Base Signal	100	65	0.65
Parametric Indicator	50	390	7.80
Non-parametric Indicator	50	585	11.7
Combined Indicator	50	650	13.0

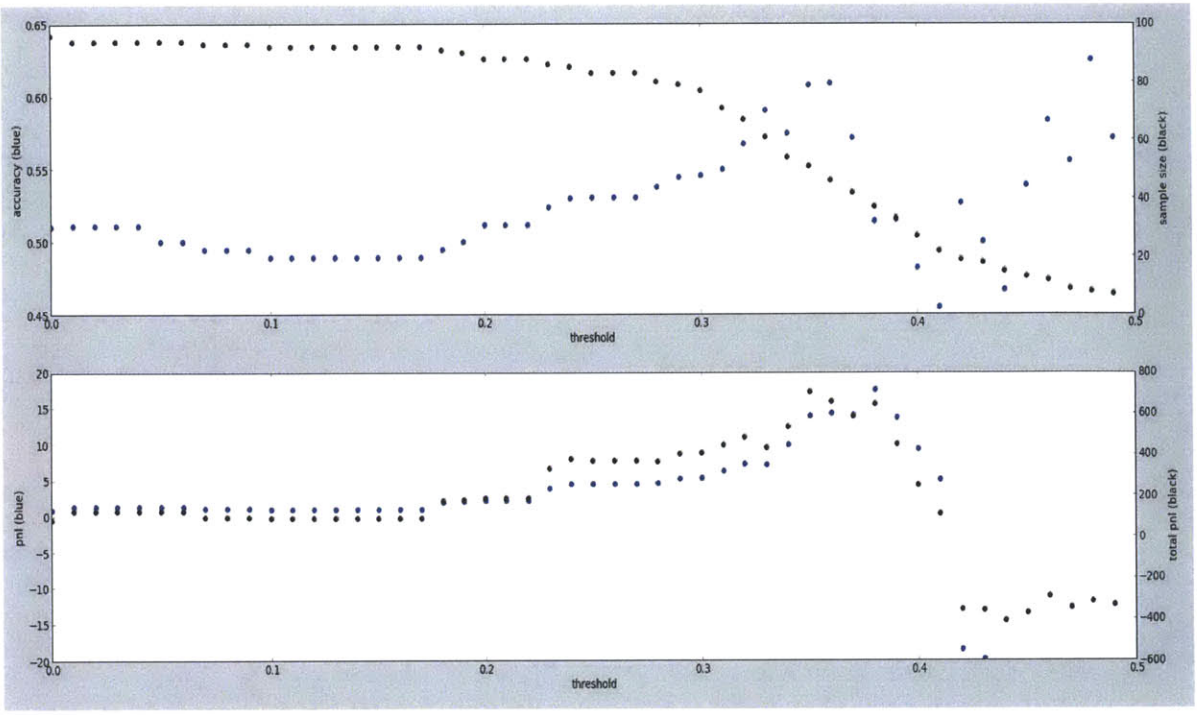
Table 2. Summary of the performance of strategies

4.4 Effect of Selectivity Threshold of the Logistical Regression

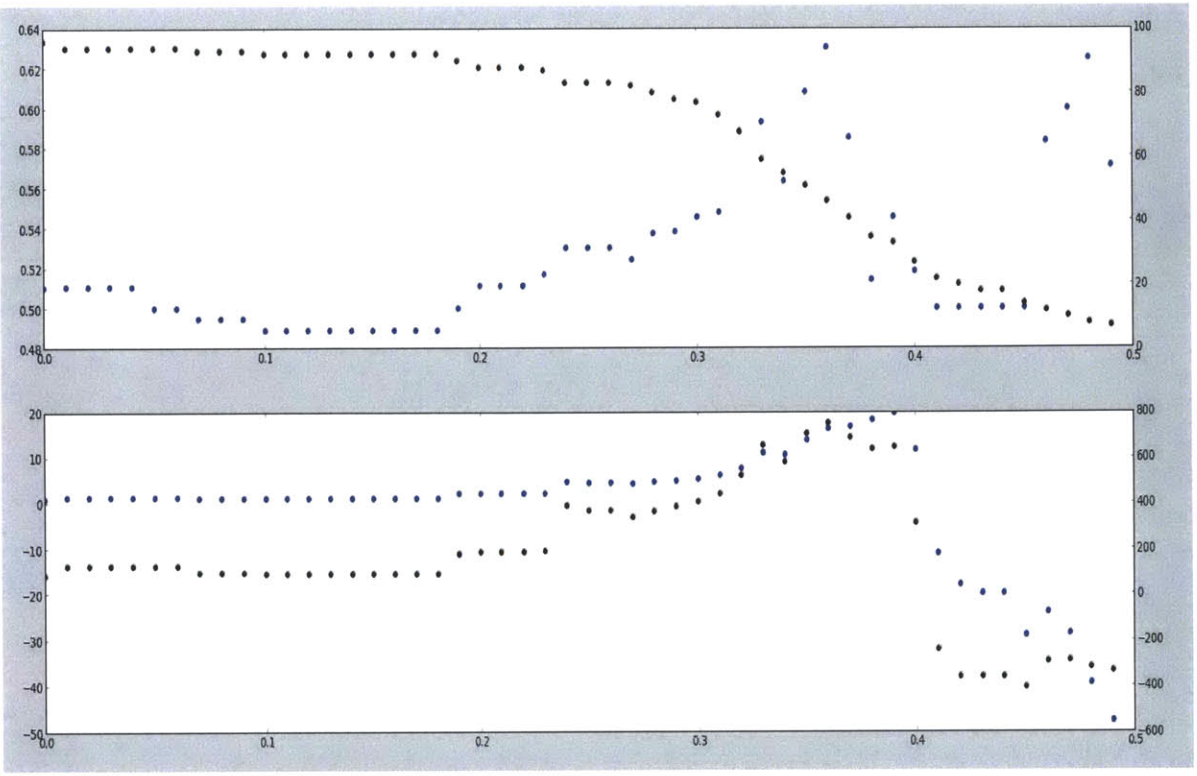
The selectivity threshold for the logistical regression determines whether an instance is to be classified as positive or negative given its probability from the regression. A higher threshold will accept less instances, but there will be less false positives. As shown in figure, for all three strategies, the number of trades decreases as the threshold increases as shown in figure 11. The directional accuracy increases as there are less false positives, that is, firing events which is followed by a drop in price. This leads to an increase both in average profit and total profit as losses are avoided. Figure shows that the average profit per trade is inversely related to the number of trades. It is noted that this relation does not hold when the number of trade is small. This is likely because with a small sample size the variance is so large that the average profit is no longer accurately measured.



(a) logistical regression using parametric indicators



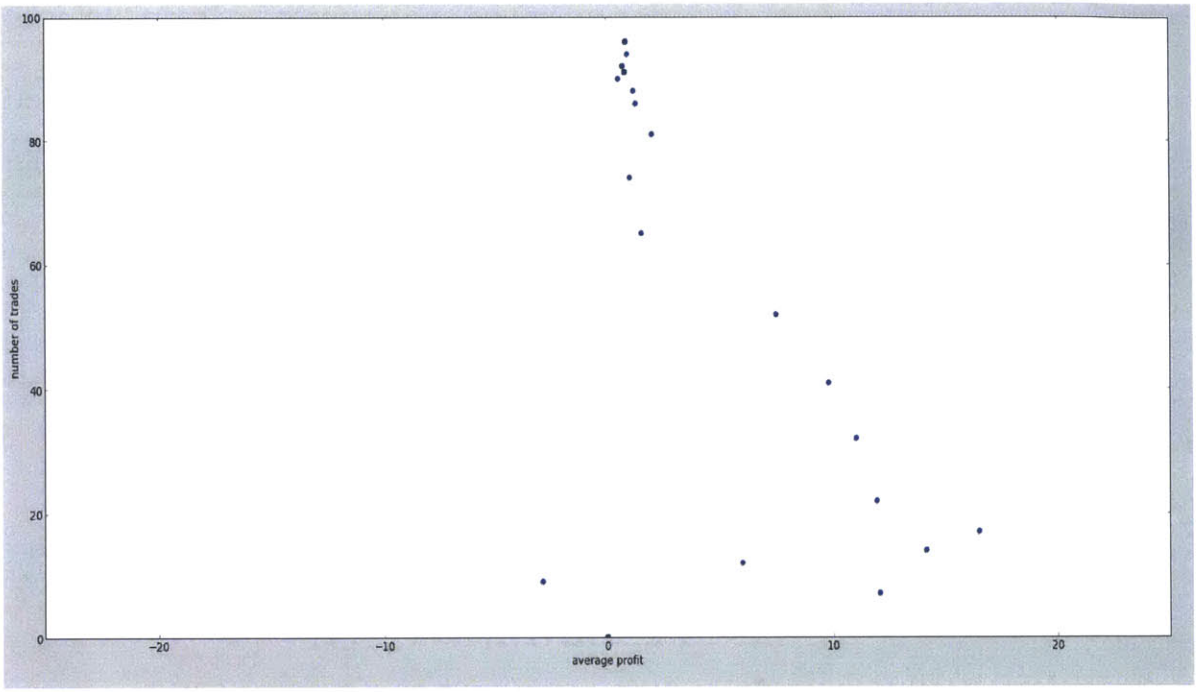
(b) logistical regression using non-parametric indicators



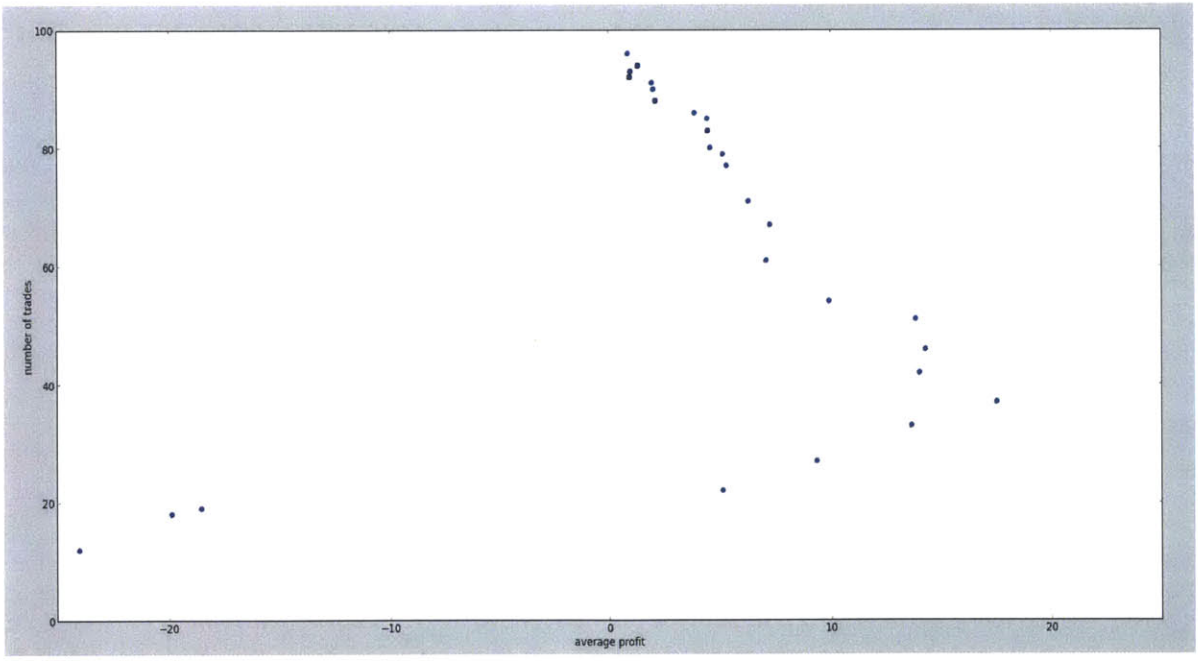
(c) logistical regression using combined indicators

Figure 11. Relationship between logistical regression threshold and the number of trades, directional accuracy, average profit, and total profit.

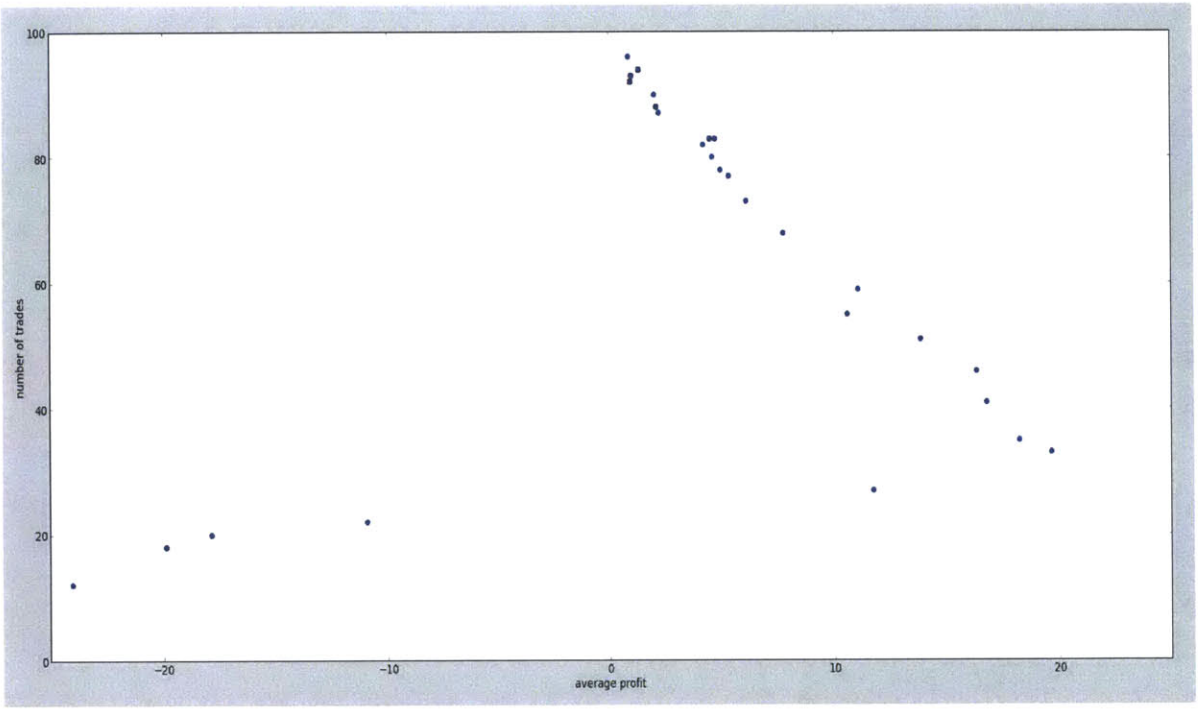
There is a trade-off between the total profit and the number of trades. In order to achieve a higher profitability, the number of trades needs to be reduced. However, with a smaller number of trades the variance of the strategy increases, leading to a greater uncertainty in the profit. Thus, the determination of the threshold should also depend the risk aversiveness of the trading style of an individual.



(a) logistical regression using parametric indicators



(b) logistical regression using non-parametric indicators



(c) logistical regression using combined indicators

Figure 12. Relationship between the number of trades and the average profit

4.5 Comparison of Parametric and Non-parametric Indicators

To compare the performance of parametric indicators and non-parametric indicators, we can examine the frontier formed by their average profit and the number of trades as shown in figure 12. This is because a strategy can have different results under different thresholds and the logistical regressions with different feature sets lead to different responsiveness of the threshold. Notably, the frontier corresponding to non-parametric indicators significantly

extends beyond that corresponding to parametric indicators, indicating a better performance. Furthermore, combining parametric and non-parametric indicators can lead to a further improvement in performance. This is likely because the two types of the indicators can extract different aspect of information. The former focuses on the change in quantities, which is treated as state variables, whereas the latter focuses on the evolution of quantities and learn from the trajectory.

Chapter 5

Conclusion

5.1 Summary of Contributions

In this work, we developed an quantitative trading algorithm for bitcoin that is shown to be profitable. The algorithm establishes a framework that combines parametric variables and non-parametric variables in a logistical regression model, capturing information in both the static states and the evolution of states. The combination improves the performance of the strategy. In addition, we demonstrated that we can discovery curve similarity of time series using cross correlation and L2 distance. The similarity metrics can be efficiently computed using convolution and can help us learn from the past instance using an ensemble voting scheme.

5.2 Future Work

There are still plenty of information in the bitcoin market which is yet utilized. One of the most features that can be incorporated in the future is the full structure of the order book. This can reveal the willingness of players to buy and sell. Another potential area for future investigation is the application of convolution to discover patterns in the data. We could

compute the pairwise distance between each subsequence in the time series, and identify clusters of subsequences that exhibits similar behaviors in the future.