Identifying Active Trading Strategies in the Bitcoin Market

CS145 Team CryptoBot

Alex Lew
California Institute of Technology
Pasadena, CA
alew@caltech.edu

Walker Mills
California Institute of Technology
Pasadena, CA
wmills@caltech.edu

ABSTRACT
The goal of this paper was to attempt to characterize the traders in the bitcoin market. Specifically, this paper attempted to analyze whether any technical-analysis based trading strategies were being used traders in the bitcoin market, and if so, which strategies were being employed and to what degree. The paper focused on simple trading strategies based on well-understood indicators such as EMA/SMA crossover. In order to test these strategies, we constructed a framework that allowed us to define different bots based on TA functions and use historical data to backtest the strategies. We examined volume at the time intervals these strategies indicated as a buy/sell opportunity to volume during other similar time intervals in order to determine if the market behaved differently at the predicted buy/sell opportunities.

1. INTRODUCTION TO BITCOIN
The subject we wish to study in this paper is the bitcoin market. bitcoin itself is the creation of Satoshi Nakamoto; his paper on bitcoin finalized the concept and led to the actual implementation of the framework. Bitcoin is a type of new currency that falls under the class "cryptocurrency". The currency consists of nodes being run by individuals that are all connected to the internet and thus form a network. The main innovation of the currency is the blockchain; the blockchain is a complete log of all the transactions ever made by any node in the bitcoin network. Using this log, the creation and subsequent transferring of every single bitcoin in existence can be traced. The transaction data is packaged into blocks that are added one by one to the blockchain by bitcoin miners. Each block has a Merkle Root that is generated from the hash of the preceding block and the transactions included in the block. The miner that generates a valid hash that matches the Merkle Root has produced a valid block and can attach this to the blockchain and receive the reward: currently, 25 bitcoin. These blocks are then broadcast to the network so that everyone can include all the transactions in their local log. The protocol accounts for possible forks and other eventualities and ensures that everyone has a log consistent with everyone else's. Thus, it is very easy to verify where each bitcoin was last sent by the network. Bitcoin has no value intrinsically; its value is based on its potential as a payment system that allows for fast transactions as well as its potential as a financial investment. In the last 2-3 years, bitcoin has received more and more media publicity; it has experience a growth of value from less than a cent to over 1200 dollars per bitcoin at its peak.

1.1 Character of Bitcoin Market
Most experts agree that as it currently exists, bitcoin has yet to make any impact on the way that payments are currently processed. The wire-transfer network was only recently extended to handle electronic wires a few decades ago and since then, most of the infrastructure necessary to process payments through traditional banking institutions has been developed and made available. Today, most payments that happen online occur when people use their credit/debit cards to handle transactions. Some companies, such as Paypal or Square, make it easier to process payments but they still leverage the underlying system of banking institutions. Bitcoin completely bypasses banking institutions since it relies only on the internet to broadcast a transaction to the network. However, the infrastructure to convert more common forms of currency i.e. USD to bitcoin are lacking or nonexistent. Thus, much of the current interest in bitcoin is not because of bitcoin's utility as a currency but rather as a financial commodity that might provide return on investment.

The extreme volatility of the price in the past has drawn many individuals to the bitcoin market to invest in bitcoin. However, the extreme volatility and the fact that bitcoin was only created 6 years ago has discouraged most large financial institutions to ignore bitcoin as a potential investment. We posit that most of the traders currently trading bitcoin are essentially daytraders and not institutions. Because of this, the types of trading strategies are much less complex than those employed by large firms on the more established stock/commodity/bond markets. Moreover, bitcoin is interesting as a financial instrument in that there are no fundamentals to be analyzed. Most stocks and bonds can be analyzed based on some trait of the instrument; stocks have P/E ratio's and dividends while bonds have return percentage's and ratings from financial institutions like Moody's. However, bitcoin has no fundamentals to be measured. The only measurement that is even vaguely related is the market cap of bitcoin used for transactions; the sentiment here is that the more popular bitcoin becomes as a payment system, the more each bitcoin will be worth intrinsically. This is however a weak correlation at best and is not very helpful to the day trader. As a result, any sort of financial analysis of bitcoin is very lacking and not helpful in developing a trading strategy. Thus, we see that many bitcoin traders either rely on their "gut feeling" about the market based on news of bitcoin adoption/rejection or other events in the bitcoin ecosystem i.e. Mt Gox's insolvency announcement. These traders are simply trading on the perceived sentiment of the market. Others rely on strategies based purely on charts and trade data; this is a technical analysis(TA) based approach and it has become popular among bitcoin traders due to a lack of any real alternatives. While many of the TA indicators and strategies were developed for stocks and don't apply to bitcoin, others can still be considered viable measures and indicators when applied to bitcoin.

1.2 Bitcoin Exchanges

Currently, all of the trading in the bitcoin market occurs on various online bitcoin exchanges. The way these exchanges operate is they essentially maintain a liquid pool of bitcoin and fiat so that people can withdraw their bitcoin and fiat at any time. Individuals who wish to trade on the exchange do so by depositing bitcoin through a transaction to the exchange's wallet or by making a wire transfer to the exchange's bank account. The exchange then credits your account on their framework with that amount of money. You can then submit market or limit orders that are placed in the exchange's orderbook. The orders that you have placed will be filled as soon as your buy/sell order can be matched to a corresponding one. Most exchanges only offer this limited structure for placing orders; some allow more complex orders including the option to go long/short on a stock and to employ leverage as well as options on order such as fill-or-kill, etc.

The largest bitcoin exchange was previously Mt. Gox; in February 2014, Mt. Gox announced that due to a coding issue, they had lost much of their bitcoin holdings and filed for bankruptcy. Other exchanges in the past have also had issues with losing coins and going bankrupt: Bitfloor also closed in April 2013. The largest bitcoin exchanges are currently OKCoin, BitStamp, Bitfinex, btc-e, and BTC-China.

1.3 Technical Analysis

Our hypothesis is that many traders are using basic and well-understood technical analysis functions. TA functions fall under 2 main categories: chart/sentiment-based and trend/data based. Chart/sentiment-based indicators are commonly defined as certain chart patterns; one of the most basic, a doji is when the opening and closing prices for a time period are the same but the price ranged both up and down during the time period. The theory is that this signal indicates that the impetus on the buy and sell sides has reached some sort of balance and thus if any trend previously existed, then this might be a turning point in that trend. The rest of the chart/sentiment indicators are similarly defined chart patterns that are associated with some estimate of the sentiment of the market based on the chart patterns. These indicators are simply based on existence; if we observe the chart pattern, then we conclude that the market is about to do such and such.

The other class of TA functions are based more on calculations from data. An easy example is the SMA crossover strategy. SMA stands for simple moving average; it is a trendline that is created by averaging the prices of the last x periods together for each point. An SMA crossover pattern involves 2 SMA trendlines: a shorter time-period one and a longer time-period one. The idea here is that if the short term SMA line crosses over the long term one, then this indicates that the short term market sentiment is more positive than the long term trend has been and thus we can expect the price to continue to climb for at least a short time. Similarly, if the short-term line crosses below the long-term line, then the short-term rate of growth would seem to be slowing and thus selling some amount of your portfolio might be advisable. Other trend/data based indicators include Chaikin oscillator, MACD, and Linear Regression.

2. GOALS

The goal of this project is to attempt to characterize the bitcoin market and specifically what strategies are being used by traders. We are also interested in what percentage of the trades come from traders using these TA strategies. We hypothesize that due to the nature of the bitcoin market and its relative infancy, the strategies employed by traders will be based on simple and well-understood TA functions. In order to accomplish this, we selected a few TA functions to examine and to search for evidence of usage of in the bitcoin market. In order to accomplish this, we first needed to construct a framework that would allow us to define algorithmic trading bots based on TA functions. Once this framework was constructed, we then used historical data from the bitcoin exchanges to backtest the bots and find the points that they suggested trading at. Using this data, we then examined similar points to see if the points predicted by our bots were statistically significant in any ways.

2.1 Technology

One of our reach goals was to develop a trading service based on this research, so with this in mind, we wrote a highly scalable, sophisticated framework through which to run our trading bots. For the purposes of this project, we have created a distributed platform for running bots, along with a centralized web interface capable of allowing users to create their own custom bots. The majority of our codebase is in C++, with some amount of Python for our Django-based website, as well as data processing. Our framework leverages Apache Thrift, an interface definition language and binary communication protocol, to provide RPC (remote procedure calling) capabilities for coordinating the assignment of trading bots to a heterogeneous computing cluster. All of our services and trade data are stored using NuoDB, a real-time database designed for cloud-scale applications.

All of our design decisions for the framework were made with scalability, extensibility, and security in mind. Each node in our compute cluster hosts a Thrift server capable of handling numerous requests simultaneously without blocking. We chose Thrift over its competitor, Google's Protocol Buffers, because it is both an IDL and RPC system, whereas Protocol Buffers just define an IDL. NuoDB is fully ACIDcompliant, and uses a key-value store behind a traditional SQL interface for performance reasons; it was developed explicitly to meet the demands of cloud-scale web applications, and as such was designed to handle large numbers of concurrent connections safely and efficiently. We chose it based on our goal of real time bots because unlike MySQL, which requires the use of transactions, functioning essentially as a locking primitive, to accomplish concurrent operations, NuoDB is by default capable of handling massive numbers of concurrent reads, writes, and updates; in the context of this project, this capability would be used to allow large numbers of bots to concurrently access trade data, even as it is inserted into the database as soon as it is broadcast live from active Bitcoin exchanges. Furthermore, because of its unique architecture, composed of separate transaction managers for handling SQL queries, and storage managers for handling disk I/O, when properly configured, it is a geo-distributed and fault-tolerant system without the hassle of sharding; each database, however distributed, is represented by a single logical unit, easily accessible through a C++ API. Additionally, our bots are integrated with Django's user authentication systems in order to provide secure access through our web interface.

2.2 Framework

We provide a simple client-facing library for the creation of fully customizable bots. Each bot is created through an interface class which packs the provided parameters into a control structure composed of rules; a bot may have an arbitrary number of rules, each of which may be composed of one or more technical indicators. For the purposes of this work, supported rules include SMA crossover, as described in Section 1.3; eventually, more rules will be supported, but that is beyond the scope of this paper. Once a rule set has been defined, the bot interface can be 'run.' Upon receiving the run command, the interface class first archives its rule set as a binary stream, and writes that object through to the database. Next, it requests a compute node from the network controller. Each server in our compute cluster is assigned a workload based on the bots it's currently hosting, updated as these workloads may shift, and persisted to the a table in our database, treated in this instance as a bulletin board. Workloads are calculated for each bot, more more appropriately, each rule; given the small number of currently implemented rules, workload was assigned empirically, although in the future, more accurate estimates will be made based on metrics such as instruction count, CPU cycles, and depending on the actual homogeneity of our compute cluster, perhaps performance benchmarks. Hosts are chosen based on a simple priority queue; the least busy compute node is chosen by the network controller and returned to the interface. Now, the interface class connects a Thrift client to the multithreaded server running on the assigned compute node and sends the run command, along with that bot's id.

Upon receiving a command to run a bot, one of the threads from the Thrift server's pool forks a new child process to handle the actual work. To do so, the child initializes a new worker, constructed with the id passed to the Thrift server. The worker then begins by first retrieving the appropriate rule set from the database, as stored by the interface and deserializing it to produce its instruction set. Records are updated in the database to reflect the new child process, since these workers are designed to run indefinitely, or until stopped by the user, and the worker commences its task. The rule set is an entirely self-contained logical unit; that is, it contains all the information necessary to run itself. The worker's only job is essentially to iterate through its rule set, and evaluate each rule. To stop a worker, one again uses the interface to send a stop command, which identifies the compute node hosting the worker corresponding to that bot id, and sends it a stop command. Once received by the Thrift server, the Unix process is killed, and the worker terminates; the interface then cleans up the records for that bot in the database.

Rules are also assigned an action and an amount. The action can be either buy, sell, or watch (i.e., gather data), depending on the user's interpretation of a rule's result. Until Mt. Gox's recent bankruptcy, these bots were also capable of interfacing with the Mt. Gox HTTP API through our middleware, which would have allowed them to execute trades autonomously, or with human confirmation, if desired. In practice, however, this project is far from ready for testing with actual money in a real market, so it was not deemed necessary to reimplement the interface for interacting with another exchange or exchange(s) once Mt. Gox was no longer a viable option.

Because our framework is written in C++, it is easily wrapped using Boost.Python to provide a seamless interface from native Python code. In this way, our user portal, written using Django, acts as an intermediate between the users' inputs and our client-side libraries, taking in user-defined parameters to define a rule set. Under this architecture, the webserver is the client, and the compute cluster is the server, as used above. Although in practice they may not be, our code was written under the assumption that the webserver, database server, and compute nodes are separate physical machines.

2.3 Scalability

At this point, the only inhibition to full-scale deployment, besides the obvious lack of features as a commercial product, is our lack of hardware. In theory, the systems we've built, if properly provisioned, are capable of handling large numbers of bots acting simultaneously.

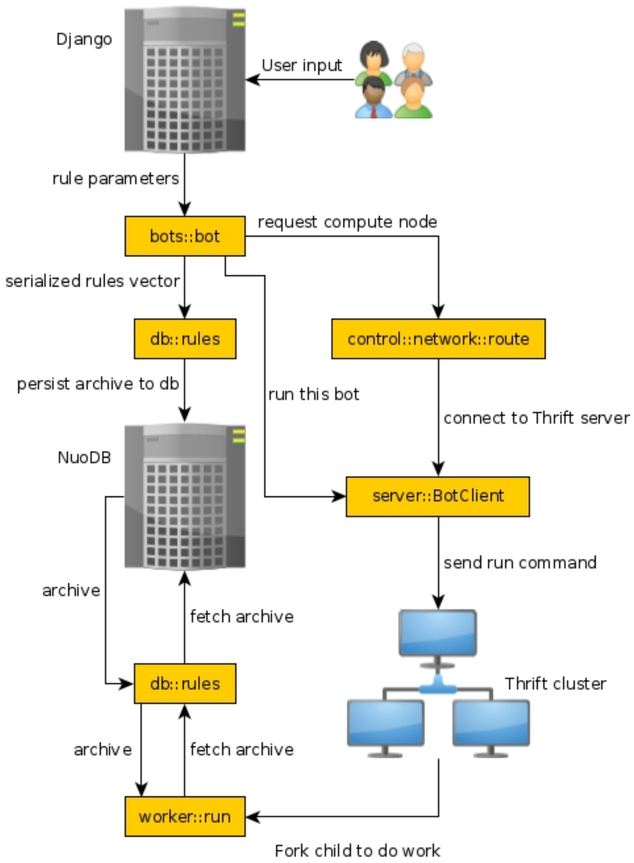
2.4 Raw Data

The raw data for our bots was provided directly by the exchanges through their API's. Using python scripts, we downloaded all trade data from Bitstamp and btc-e. The data consisted of every single trade made on these exchanges. Due to the relative newness of the bitcoin market, many of the exchanges have experienced long periods of very low trade volumes, particularly toward 2011 when they were first started. Up until roughly 2013, the volumes of the time periods vary widely. This is significant because it means that the volumes may not be normally distributed, since the market was not robust enough to produce a normal distribution in the volumes.

2.5 Selection of trading strategies

We were only able to test for patterns correlating to a few trading strategies. We decided to focus on studying the

Figure 1: Framework Control Flow



simplest trading strategies. The strategies that we chose were the following:

1. SMA Crossover
2. EMA Crossover

In the context of technical analysis functions, SMA stands for Simple Moving Average. This indicator tracks the current trend of the market; it is simply an average of the closing prices over the last n days of trading. The n is a variable that represents the time period associated with the indicator; common values of n range from 5 to 50. EMA stands for Exponential Moving Average. This is a similar indicator, only it weights data points from closer to the present more heavily; this is essentially a weighted SMA. The crossover signal is based on the idea that when the indicator based on a shorter time period crosses above the longer time period indicator, then this indicates that the short-term sentiment of the market is more positive than it has been in the long term. Thus, short term growth is a reasonable expectation, and so an SMA up-cross is a buy signal. Similarly, when the short-term indicator crosses below, the sentiment at the moment is understood to be more negative than it has been over the long term; this represents a sell opportunity. The same idea holds for EMA as well.

The time period selected for the indicator is based on the time scale that the trader acts on. If the individual is a day-trader who is looking to open and close positions on an hourly basis, then the short-term SMA's are more relevant since they capture the smaller trends in the price. Likewise, if a trader is only interested in long-term trades on the order of a week or a month, then short-term SMA's produce too much noise and are inconsistent trading signals. We tested a variety of time periods for each of our indicators, all within the range of 5 to 50. In the future, we are looking at testing a wider variety of technical indicators.

Figure 2: Example of an SMA

2.6 Bot Data Collection

Using the price history data, we backtested the algorithmic bots that we created for each of the SMA/EMA crossover pairs. The result was a set of points at which the short-term SMA/EMA crossed above/below the long-term SMA/EMA. This set of points is the set of time periods during which traders who are employing the SMA/EMA crossover strategy would trade during. For each set of SMA/EMA indicators, we produced a set of crossover points.

3. DATA ANALYSIS

Our hypothesis is that if a significant portion of the market is employing these strategies, then the volume at our predicted buy/sell opportunities should be significantly higher than the volume at other similar time periods with no crossover. Our raw data is simply the volume of the time periods calculated from aggregating the price data into time period data points.

An important part of the data analysis was attempting to narrow down the points that we would compare each of our crossover points to. We selected points that were similar based on 2 criteria:

1. Price volatility / standard deviation of prices within time period
2. Strength of signal/trend

3.1 Selecting for price volatility

In analysis of price data, an important measure of the behavior of the price is volatility. When an instrument's price ranges widely over a short period of time, then this instrument has high volatility. Conversely, if the instrument's price remains fairly constant, then the volatility is low. There is a very clear implicit connection between price volatility and volume over a time period. If the price volatility is higher, then this implies that the price has been ranging over the time period, which implies that the equilibrium between the buy orders and the sell orders is changing quickly. This happens when a lot of either buy or sell orders are submitted at once or when the sentiment of the market has swung strongly in one direction. When the price is moving more quickly, more people are inclined to either enter or exit the market than when the price is stagnant; as such, the volume should be positively correlated with price volatility. The most common way to measure the price volatility over a time period is to calculate the standard deviation of the prices from the trade data. Thus, in order to select points that are similar in price volatility to our crossover points, we calculated the standard deviation of every single time period, and then selected time periods that had a standard deviation within 1% of the standard deviation of our crossover point.

3.2 Selecting for strength of signal/trend

The slope of the SMA/EMA indicator is essentially a measure of how fast the indicator is changing; that is, how different are the more current points from the trailing ones. If, for a 10 day SMA, the first 8 days are relatively constant but the price moves quickly during the last 2, then the slope of the SMA/EMA will be much higher. The slope of the short-term SMA/EMA as it crosses through the long-term SMA/EMA is thus a measure of how strong the sentiment is. If the slope of the short-term SMA/EMA is much higher than that of the long term, then that means that the short-term sentiment is significantly more positive than if the slope's were roughly equal. Thus, another consideration in our selection of similar points was to attempt to control for the strength of the signal at the crossover point vs. the slope of the short-term SMA/EMA at various other points. We selected points with a slope value ± 0.05.

3.3 Analysis of Data Sets

After gathering all our data, we produced a set of similar points for each of our crossover points for each of our SMA/EMA pairs. That is, a corresponding similar points set was produced for every single crossover point produced by every single strategy. We then tested whether the volume of the set of similar points was statistically different from the volume at the crossover points.

We did this using a one-sample t-test with the volume of the crossover point as our expected mean.

3.4 Example Calculation

Here, we demonstrate the basic form of one of the statistical tests that were performed on our data sets.

Research Hypothesis: The mean volume of our sample will be lower than the expected mean (mean at our crossover point)

Null Hypothesis: The mean volume of our sample will be greater than or equal to the expected mean (mean at our crossover point)

Crossover Time Period: 1316685600
Size of Similar Point Dataset: 10
Test: One-sample T-Test

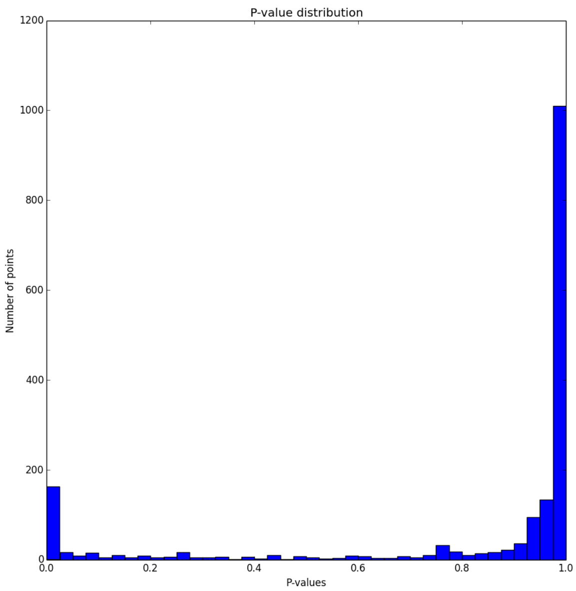
P-values:
Two-tailed: 0.103898
Left-tailed: 0.948051
Right-tailed: 0.051949

Based on the hypothesis, we see that the p-value we're interested in is the left-tailed p-value. Here, the left-tailed p-value is 0.948; we cannot reject our null hypothesis. Thus, we see that for this crossover point, our conclusion is that the volume is not higher than it is at similar points.

4. RESULTS

Across all of our trading strategies, we found a total of 1741 crossover points. After performing a similar t-test to the one above on all of them, we produced 1741 p-values. The distribution of these p-values gives us insight into whether our hypothesis was true and if so, on what scale. That is; how many of our points had p-values that suggest our hypothesis was true. Below is the plot showing the distribution of p-values for our points.

Figure 3: P-value Distribution

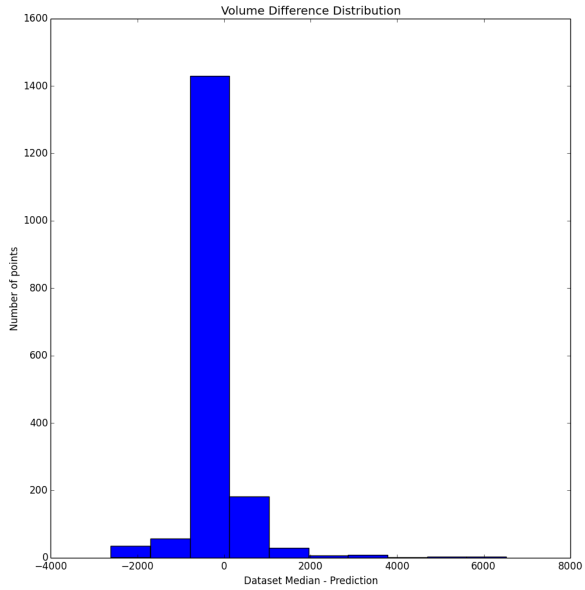


4.1 Analysis of degree of effect

Another number of interest to us is the degree to which the volumes in the similar points differ from the volume at the crossover point in each case. This number is a measure of how much the market is deviating from what a normal volume would be at the crossover points. If this difference is large, then that implies that a more significant part of the market is employing this strategy. Conversely, if the difference is small, then the implication is that only a very small part of the market is employing this strategy.

We chose to take the difference between the median of the volumes in the similar points set and the volume of the crossover point. The volumes over time periods range widely within the similar points sets; thus, the mean may have been strongly affected by outliers that are not representative of the data set. For this reason, we chose to use the mean instead to calculate this measure. The plot below describes the distribution of this difference across all the crossover points.

Figure 4: Median vs. expected Mean Distribution



5. DISCUSSION OF RESULTS

Right away, we can see that the results do not support our hypothesis. In fact, in over 1000 out of 1700 cases, the left-tailed p-value fell into the 0.95 - 1.0 category. This indicates that in these cases, our hypothesis was almost definitely false. This result is corroborated by the distributions of the median volume vs. crossover volume difference. The volume difference distribution is clearly centered below 0 and most of the points fall below 0. A negative value of the difference indicates that the median volume was higher than the crossover volume, so this shows that most of the points had a negative difference as well. Overall, these results indicate that the the volume at the crossover points was consistently lower than volumes at similar prices.

The implications here are very interesting. To see this, consider the reverse of our hypothesis: that nobody is employing these strategies. Then, we would expect that the volume of the crossover points should be very similar to other points in the dataset since there would be nothing differentiating the crossover points from any other points. Under these conditions, we would expect the p-values to be normally distributed around 0.5. This is because the similar point sets menas should differ from the crossover point mean within a normal distribution. However, the results that we see are wildly different from this. The fact that this many points have a p-value between 0.95 and 1.0 implies that the opposite hypothesis is true for these points: that the volume at the crossover point is significant lower than the mean volume of the similar point set. The reason for this is unclear, but there are a few possibilities.

1. Inconsistent / Non-viable data

2. Counterstrategy is popular

3. Flawed assumptions

5.1 Inconsistent / Non-viable data

Many of the data entries from earlier in the price history consist of 0 trades. Thus, the average price isn't define and thus calculating the SMA/EMA with these points produces a confusing and misleading indicator. The price points during those time periods could be assumed to be similar to those of surrounding time periods, but since the SMA takes the average price of the time period as part of the calculation, this information is lost. This means that while the price may not have changed much, the SMA/EMA indicator could be much more strongly affected by these zero-trade time periods than it should be. Thus, many crossover signals could have resulted from erroneous or misleading SMA/EMA behavior. Essentially, the SMA/EMA is not suited to calculating a trend when there are time periods of no trading. This is much less of a problem on the stock market, where most instruments have a decent amount of volatility. However, many bitcoin exchanges had very little volume when they were first started, and thus the data includes a large portion of these confounding zero-trade time periods.

These zero-trade time periods would produce more crossover points during low-volume periods. Thus, it would be expected that the volume of the similar point dataset would be signfiicantly higher since our crossover points are essentially noise and do not represent an actual trading strategy. A way to solve this would be to only consider crossover points that occur after the exchanges have the volume to provide liquidity and no zero-trade time periods occur.

5.2 Counterstrategy is popular

Assuming that the results correctly represent the behavior of the market, then we have the conclusion that the points selected by the SMA/EMA crossover strategy are particular in some way. If nobody is actively trading with these strategies, then another explanation is that strategy that is used by some signficant amount of the market predicts buy/sell opportunities at times that are not when the SMA/EMA crossovers occur. That is, a strategy that recommends trading at times that do not coincide with the SMA/EMA crossovers is very popular among traders. This is a reasonable theory; the main issue with the SMA/EMA crossover strategy is that it is a lagging indicator. This means that the buy/sell opportunities predicted by the SMA/EMA crossover strategies lag behind the optimal buy/sell opportunity. This is because a crossover cannot be identified until it has already occured; thus, the optimal time to buy or sell has already passed by the time the crossover strategy identifies it. Thus, other strategies that are either lag-free or predictive would result in trading at times that do not coincide with the time periods predicted by crossovers. If we hypothesize that such a strategy is indeed popular among traders, then the expected results would be exactly what we've observed. Since our analysis only considered 2 of the simplest technical indicator strategies, its clear that if we expanded our search, we may be able to identify the counter-strategy that we see proof of.

5.3 Flawed assumptions

One of the key assumptions of a t-test is that the samples are normally distributed. However, as we previously discussed, the nature of the bitcoin market has varied widely since it was created. For the first few years of trading, there simply wasn't enough volume or interest to create a fluid market. Trades remained unfilled for long periods of time simply because there weren't enough orders to provide liquidiy. This is due to the fact that the stock market rewards liquidity providers, but any rewards for liquidity providers on the bitcoin exchanges were typically lower than the fee the exchanges charged to trade. Thus, the assumption that the volumes of the datasets were normally distributed was potentially incorrect.

6. FUTURE GOALS

The minimum goals for this project were met, but there are many directions that this project could be extended in. Some of our future goals include:

1. Include more TA indicators
2. Cleanup / Enhance Dataset
3. Use other statistical tests that don't rely on normally distributed samples
4. Develop an accurate method for determining a bot's expected workload

Additionally, with these goals in mind, we will most likely rewrite a significant portion of our codebase, with a goal of improving ease of use and ease of extensibility. As it currently stands, rules are fairly costly to implement, and relatively rigid in structure. In the future, we hope to make improvements to our control structure design in order to ease the addition of more technical indicators, remove some inefficiencies inherent in the architecture design, and allow for greater flexibility when combining control units. Notably, we wish to reduce the amount of information that must be stored about a rule set, so that they can be easily stored in terms of rule types and parameters, rather than as archived objects, which would reduce data transfer across the network and improve cross-language compatibility.

7. BIBLIOGRAPHY

1. Chan, Ernest P. Quantitative Trading: How to Build Your Own Algorithmic Trading Business. Hoboken, NJ, USA: John Wiley & Sons, 2009. Print.

2. Crack, Timothy Falcon. Basic Black-Scholes: Option Pricing and Trading. London, UK: T.F. Crack, 2009. Print.

3. Haritsa, Jayant R., and Krithi Ramamritham. "Real-Time Database Systems in the New Millenium." Real-Time Systems 19.3 (2000): 205-08. Print.

4. Haritsa, Jayant R., Michael J. Canrey, and Miron Livny. "Value-based Scheduling in Real-time Database Systems." The VLDB Journal 2.2 (1993): 117-52. Print.

5. Hvasshovd, Svein-Olaf, Øystein Torbjørnsen, Svein Erik Bratsberg, and Per Holager. "The ClustRa Telecom Database: High Availability, High Throughput, and Real-Time Response." Proceedings of 21th International Conference on Very Large Data Bases. Zurich, Switzerland. 1995. Print.

6. Leshik, Edward A., and Jane Cralle. An Introduction to Algorithmic Trading: Basic to Advanced Strategies. Chichester, West Sussex, UK: Wiley, 2011. Print.

7. Mullender, Sape J. "Process Management in Distributed Operating Systems." Experiences with Distributed Systems: International Workshop, Kaiserslautern, FRG, September 28-30, 1987: Proceedings. Ed. Jürgen Nehmer. Berlin, FRG: Springer-Verlag, 1988. Print.

8. Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System." www.bitcoin.org. 31 Oct. 2008. Web.

9. Narang, Rishi K. Inside the Black Box the Simple Truth about Quantitative Trading. Hoboken, NJ, USA: Wiley, 2009. Print.

10. Nguryen, Fiona, and Dennis Ferenzy. Bitcoin - Tulip Mania or Revolution. IIF Capital Markets Monitor: 9-10. 8 Jan. 2014. Web.

11. Proctor, Seth. "A Technical Whitepaper." www.nuodb.com. NuoDB, 15 Oct. 2013. Web. 18 Feb. 2014.

12. Tseng, Shin-Mu, Y. H. Chin, and Wei-Pang Yang. "Scheduling Value-Based Transactions in Real-Time Main-Memory Databases." Proceedings of First Workshop on Real-Time Databases: Issues and Applications. Newport Beach, CA, USA. 1996. Print.