Introduction to Programming
Master Program in Data Science
and Advanced Analytics 2018/19

Modeling Cryptocurrency Market Movements

Adriano Rodrigues - M20180430 - m20180430@novaims.unl.pt
Daniel Teixeira - M20180087 - m20180087@novaims.unl.pt
Guilherme Barros - M20180083 - m20180083@novaims.unl.pt

Abstract: The cryptocurrency markets are in their very early stages and managed to capture the attention of many people not previously involved in exchange markets because it allowed easier interfaces to buy and sell. Many people claim it behaves just like regular markets because it generates the same type of data and also involves group psychology, so they claim same patterns will emerge. Since it is in its early stages of maturity as an exchange market, it is expected not to be as competitive as more mature ones like Forex, Stocks, Options and Commodities markets, this gives us an opportunity to try to model and predict it because the movements would be more 'organic' than other markets. For this project, there are several cryptocurrency exchanges that provide historical trade data that can be queried and used. Other data points like historic news and forums/chat posts could also give us an edge on foreseeing the market. The final goal is to create a trading robot that can use the models developed to try to 'profit' on a simulated environment, either our own or on an online service like BitMEX's Testnet.

Keywords: cryptocurrency; cryptocurrency model; cryptocurrency market movements

Statement of Contribution: Adriano has the knowledge of Cryptocurrency and he is the owner of the project proposal, he was the mentor of the group. Guilherme had a very good contribution as he had a lot of experience in stock market. Daniel was key in discussing how statistically significant our result were, since the team had little to no knowledge of the usage self organizing maps before this project. All members have discussed the entire project trying to find the best solution for the proposal.

Introduction to Programming

Master Program in Data Science
and Advanced Analytics 2018/19

I. Introduction

Throughout human history, the need to understand phenomena and to predict them has always been a concern. To make decisions in agriculture (planting, fertilizer application, etc.), meteorology (weather forecasting) Understand and predict phenomena, where information (data) is collected in equal time intervals (hours, days, weeks, etc.), over a longer or shorter period is a particular facet of statistics, the analysis of time series.

The prediction methods analyze past values to predict future values. Forecast were simple extrapolation of a global value, adjusted by the time. The autoregressive model (AR) was created, in which the depended on past values, nevertheless, the linear models are insufficient for series analysis because most of the real series have strong non-linearity.

With the advancement of technology, several methods of forecasting were developed, buy the way, the method used in this project was the self organizing map, it is not a specific method, but it is powerful.

The objective of the project is to analyze the movement and the currency and find moments for possevies purchase or sales, with that we look for a way to invest and profit.

II. Data

a. Description

The collected data was the fulfillments of buy/sell orders on the exchange Binance. Since we could derive technical indicators by, in summary, grouping the data into the more common candle values and feeding the resulting series in a mathematical function.

The original data was structured in a simple text file with each line containing an order fulfillment in a json structure, for example:

```
{"a":31617230,"p":"8273.84000000","q":"0.25200600","f":36600539,"l":36600539,"
T":1524182402353,"m":true,"M":true}
```

a is the execution id, **p** is the execution price, **q** how much of the currency has been traded, **f** and **l** are the involved order ids, **T** is the timestamp, **m** denotes if the buyer was the *maker* of this transaction, and **M** if at that time that was the best price available in the order books.

b. Extraction

Binance provides an API that can be queried with an easy set of restrictions. So we wrote a scrapper to query the endpoint to collect a month’s worth of data. Initially with a python script but, given the familiarity, later a JavaScript one was employed (*data_grabber.py and data_grabber.js)*.

The resulting data was then stored in a PostgreSQL database by creating the queries (*database_filler.py*) and executing them on the database.

c. Transformation

Prior to applying the indicators to the price data to have the full data we wanted to work with, we needed to combine it in some way where moments could be compared to each other. In *traditional* market analysis, there are various approaches on interpreting what the data is “showing” to a trader. In order to pander to more than one approach, the “technical analysis” interpretation and the “group psychology” ones, we restricted the grouping of raw data to 5 seconds, which was later increased to 10 to shrink the data volume.

Since there is no boundary on the values, we couldn’t use regular Normalization techniques. We found out that in similar cases, a *Log Return* is calculated to, basically, give a number indicating the direction of movement that is fairly normal. For the indicators the data was grouped in 1 minute and 1 hour and applied the indicator functions from the TA-Lib[0] Python Library. Later after analysing the result, we decided to drop the 1 hour data, and try to diversify the kinds of indicators used.

On both runs, we applied the Log Return on all the indicators that were about price, other values were normalized according to their ranges. The resulting dataset ended up with a strange mix of ranges and we weren’t sure if it would affect the Self Organizing Map performance negatively.

At the first run, we applied the Simple Moving Average (SMA) 50, SMA 100, SMA 200, Exponential Moving Average (EMA) 13, EMA 21, EMA 55, Relative Strength Index (RSI) 2, RSI 14 and the Moving Average Convergence Divergence (MACD) on both the minute and the hour grouping.

At the second run, we applied the EMA 13, EMA 21, EMA 55, RSI 2, RSI 4, MACD, Ultimate Oscillator, Williams' %R and the On Balance Volume (OBV).

The resulting data was then appended in the database. (*schema.sql, ta_applier.py)*

Here's a brief description of all the indicators used:

The **Simple Moving Average** [1];

Is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average.

The **Exponential Moving Average** [2];

An exponential moving average EMA is a type of moving average MA that places a greater weight and significance on the most recent data points.

The **Moving Average Convergence / Divergence** [3];

Is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. Traders may buy the security when the MACD crosses above its signal line and sell, or short, the security when the MACD crosses below the signal line.

The **Signal Line** [4];

Are used in technical indicators, especially oscillators, to generate buy and sell signals or suggest a change in a trend. Often times, signal lines are moving averages of a technical indicator, such as the moving average convergence-divergence MACD and stochastics oscillator.

The profit compared to the previous day;

The loss compared to the previous day;

The **Relative Strength Index** [5];

Is a technical indicator used in the analysis of financial markets. It is intended to chart the current and historical strength or weakness of a stock or market based on the closing prices of a recent trading period.

The **Ultimate Oscillator** [6];

Is a technical indicator to measure momentum across multiple timeframes. By using the weighted average of three different timeframes.

The **Williams %R** [7];

Is a momentum indicator that is the inverse of the Fast Stochastic Oscillator. It reflects the level of the close relative to the highest high for the look-back period.

The **On-Balance Volume** [8];

Is a momentum indicator that uses volume flow to predict changes in stock price.

III. Analysis Procedures

The next step was to train a Self Organizing Map to see it would identify good buy or sell moments, we used the MiniSom[9] implementation of the map.

On both runs, we generated a visualization on the map and plotted, what in *hindsight* would have been a good entry point either to buy or sell. Given the objective of the project, a short entry-exit distance was prioritized, as it was expected to operate on a fairly fast rate for small returns. We also did both runs with its' standard configurations and rule-of-thumb for parameters and a slightly variation where we tried to make the map wider than tall, another where explicit initialization was performed and yet another where dimensionality was reduced.

After the map is generated and the moments plotted over it, we marked the 100 previous moments to see if, on the map, the moment "walks" into an important buy/sell moment mapped or is at least identified as such. (*som_runner.py*)

IV. Results and Discussion

In the end, depending on the configurations the map ended up looking wildly different, but kept the characteristics for the points we were interested in:

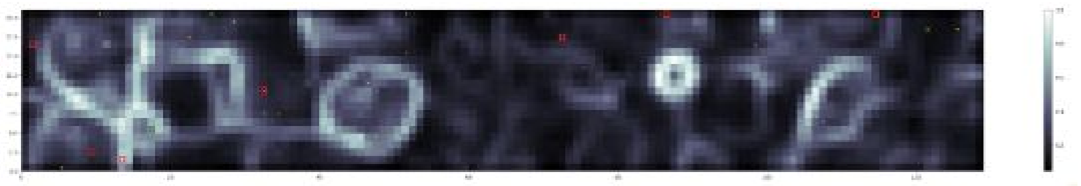

Image 1: Widened map with explicit initialization

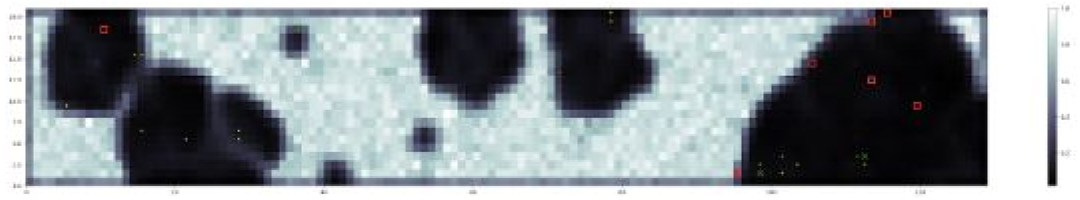
Image 2: Widened map without explicit initialization

Red squares represent what we classified as viable sell spots while green Xs were viable buy spots, yellow dots are points that preceded a good buy/sell points. Overlaps are not displayed.

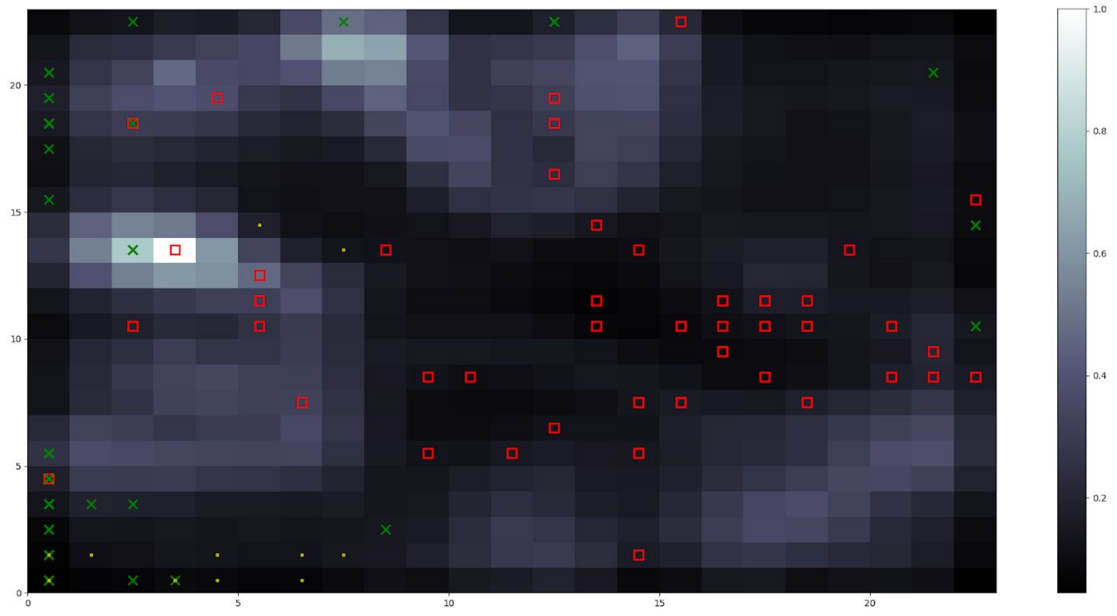

Image 3: Compact initialized map

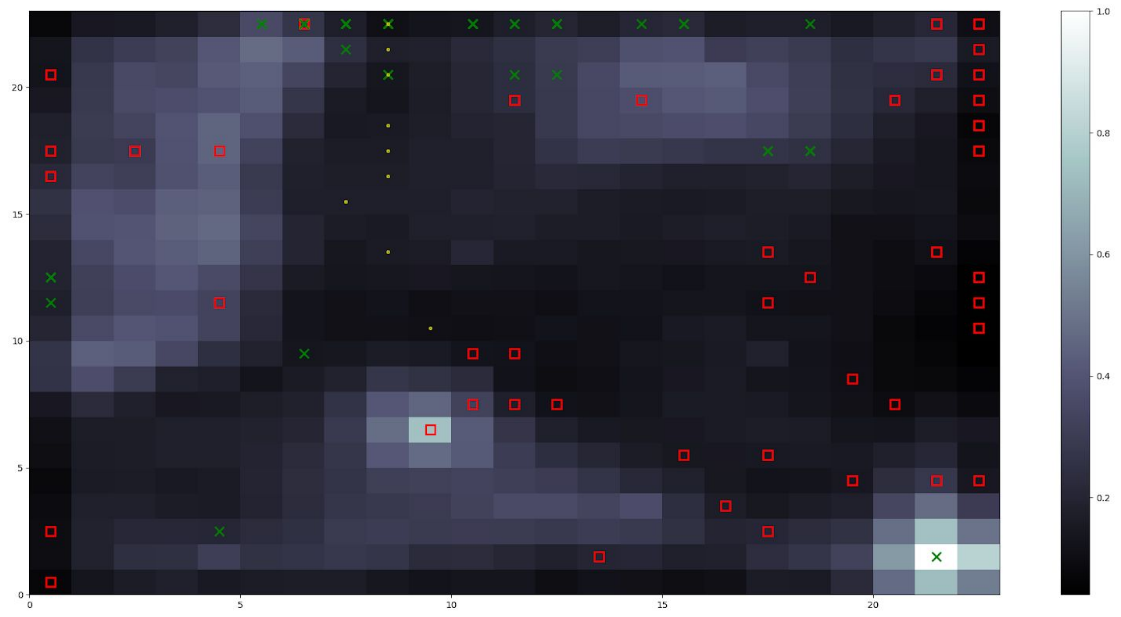

Image 4: Compact non-initialized map

Per behavior of the Self Organizing Map, we would only be interested in the spots where the number is closer to **1.0** and, no matter what we tried, the points were sometimes arguably clustered, but still scattered all over "uninteresting" nodes. Moreover, there wasn't a visible path created by the preceding data points. So drawing a conclusion here would be impossible.

V. Conclusions

If we managed to obtain maps that would identify the market moments we intended to find, the next step would have been to save a proposed map as a file so it could be loaded into a new program that would query real time market data and post buy/sell orders in a simulated market to check it's accuracy and maybe then tweak the system. Unfortunately, our approach didn't yield good results and further exploration needs to be done before putting this system against the market in any way.

VI. References

[0] https://www.ta-lib.org/

[1] https://www.investopedia.com/terms/s/sma.asp

[2] https://www.investopedia.com/terms/e/ema.asp

[3] https://www.investopedia.com/terms/m/macd.asp

[4] https://www.investopedia.com/terms/s/signal_line.asp

[5] https://www.investopedia.com/terms/r/rsi.asp

[6] https://www.investopedia.com/terms/u/ultimateoscillator.asp

[7] https://www.investopedia.com/terms/w/williamsr.asp

[8] https://www.investopedia.com/terms/o/onbalancevolume.asp

[9] https://github.com/JustGlowing/minisom