Using Twitter Data and Sentiment Analysis to Predict Future Values of Cryprocurrencies

Ryan Walker

Abstract—This paper presents numerical schemes for gathering, processing and correlating sentiment data to actual cost for a given crypocurrency over a given unit of time in the interest in finding a time-lagged correlation.

1 Introduction

SENTIMENT analysis techniques have been used for stock market predictions in the past with mixed success [1]. We are of the opinion that for a technique like this to work there are three major requirements.

1) High volume of sentiment source
2) Strong correlation to trader action and community opinion
3) Low quantity of non-deterministic value changing artifacts news reports, earnings, company announcements, etc...

In most cases, point one and three are mutually exclusive, meaning if there is a high volume of people talking about a stock publicly $\frac{100k+}{Day}$, the company will typically be publishing earning reports, posting news, etc... These are important for investors, but cannot be accurately modeled as they are considered random artifacts.

As there are still news artifacts regarding Cryptocurrencies, they are less common and typically have less of an impact because unless there is a major problem with the algorithm, they are mostly subjective, unlike an earning report or other financial documents.

The aim of this paper is to attempt to validate what has been listed above, and use the data to make further informed buys and sells.

1.1 Gathering Sentiment Data

The main source of sentiment data was from twitter. A Python module *tweepy* [2] was used to gather tweets and then bin them into cryptocurrencies of interest, the volume was anywhere from $1.2k\frac{tweets}{hr}$ to $24k\frac{tweets}{hr}$ depending on the currency of interest. For this paper we were mostly interest in BTC, as the tweet volume is the highest, in the past five months, we have gathered over 21 billion tweets on Bitcoin alone.

Once the data is gathered it's put into a time series dataset with a period of 30 seconds - from that point Python *NLTK* (Natural Language Toolkit) [3] is used to perform sentiment analysis to rate each tweet and make a net sum per unit time. Figure 1 shows the output of what I have described above.

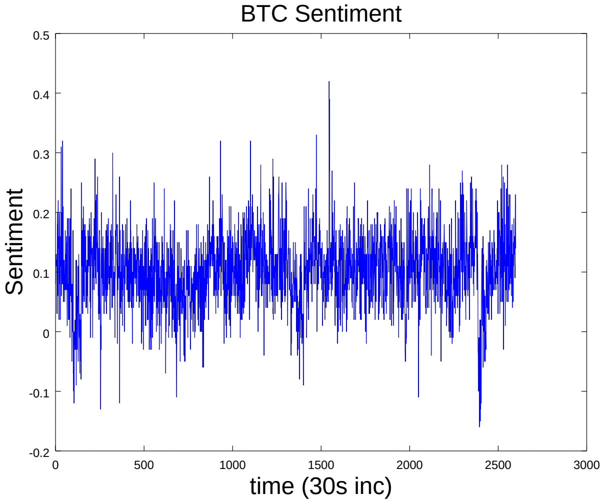

Fig. 1. Raw Sentiment, October 9th 2017

After a little filtering Figure 2 - it's possible to compare the sentiment and values time series.

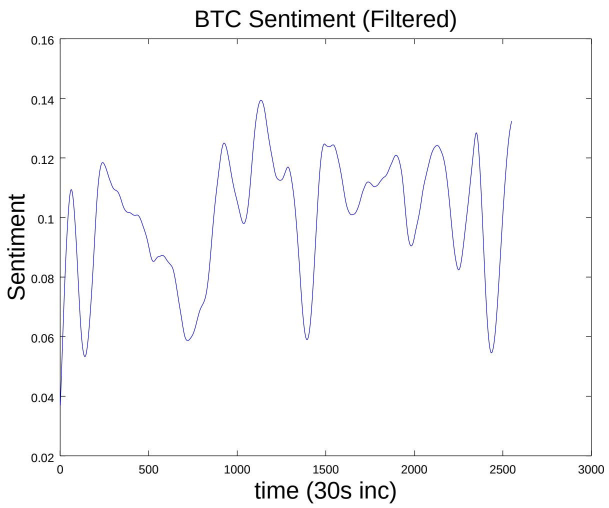

Fig. 2. Filtered Sentiment, October 9th 2017

1.2 Timewise Correlation

In the beginning, the plots were manually shifted forward in time until they 'looked right', later a numerical method was used for correlation.

The timewise correlation was done by what we call the time correlation vector k which is defined as equation 1

$$k_j = \frac{\sum_{i=0}^{n} x_i - y_{i+j}}{n-j}, \text{ where j runs from 0 to n} \tag{1}$$

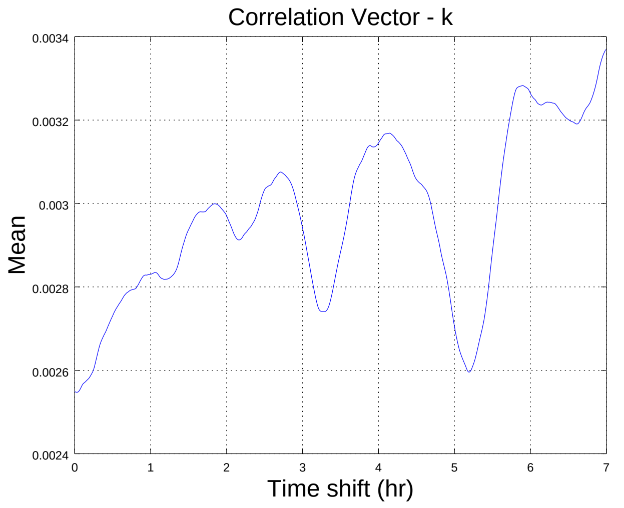

Fig. 3. Time Correlation vector k, October 7th 2017

Where $x = \frac{dCost}{dt}$ $y = \frac{dSent}{dt}$, the reasoning for using derivatives for the correlation is simply because the sentiment values are effectively a floating quantity - where the magnitude is not believed by us to have any direct relationship to the magnitude of cost. However, our findings have lead us to believe that a changing value of sentiment correlates changing cost.

k is a minimizing function where the lowest magnitude indicates the highest level of correlation between sentiment and cost. As seen in Figure 3, k has a global minima around 5.2hrs, which I define as $\tau_L = 5.2hrs$ or the effective 'Time lag' between high values of changing sentiment and high values of changing cost.

It has been observed that τ_L has been repeatable over a given unit of time, Figure 4 shows the vector sum of many k from November 2nd to November 17th (15 k vectors). In addition there are some interesting patterns appearing as local minama at $\sim 1.8hrs$ and $\sim 3.2hrs$.

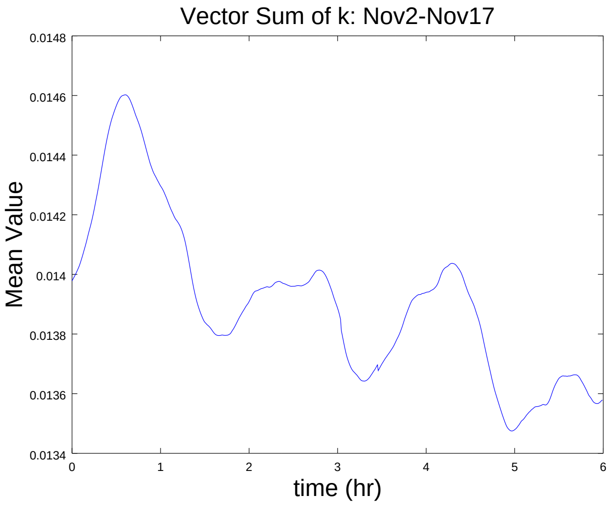

Fig. 4. Mean Time Correlation vector k, Nov2 - Nov17, 2017

Figure 5 shows the sentiment shifted forward in the by τ_L. It can be observed that rates of high change and local minima and maxima are matched between the two datasets. There are further similar plots Figure: 7 - 10

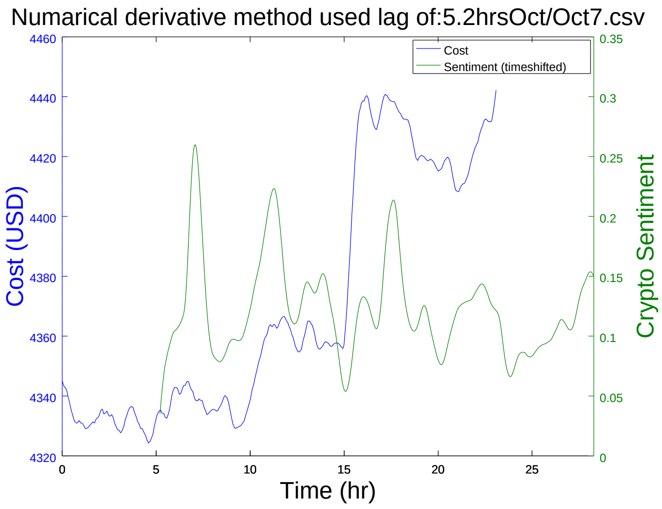

Fig. 5. Time Shifted Sentiment, October 9th 2017

1.3 Realtime Implementation

Moving into a real time implementation was simple enough, previously the data post processing was being done in Octave, for the real time implementation Python was used. The application followed this structure,

1) Fixed length of data was gathered, used 10hrs
2) Steps above were executed, filtered, k found and τ_L found
3) Once τ_L was known, and you were at time $t = \tau$ you have $\tau - \tau_L$ worth of 'useful data' that you could make predictions on
4) Buy and sell triggers were found (see below)
5) Push notifications are sent to a mobile device using pushover [5]

The crux of the real time application is finding the trigger points (Item 4), these are values or grouping of values that indicate that something is going to happen in τ_L time. As seen in Figure 5 high slope values of sentiment can be used to infer changing cost values in time τ_L. So high slope positive and negative slopes values were taken and labeled as buy and sell triggers respectively, this is one possibly method of finding the triggers, but further work should be done to understand if this is the best.

This was the first crack at the real time app, it's currently running and sending out advice that is being used for HFT (high frequency trading)

2 Future Work

2.1 Optimisation problem

There are a lot of parameters in the real time application that are in need of optimisation.

- τ - the amount of data to collect in the real time app before starting the post operations
- Number of buy/sell triggers, per τ
- Stronger understanding of the Time Zone Inference (See appendix)
- Threshold for buy/sell triggers, what counts a 'high sentiment'
- Filter cutoff

Using all the archived data that we have been collecting over that past 5 months we could write an iterative application that tunes each parameter to reduce the error. This would be extremely computational expensive most likely require offloading to AWS or another similar service.

2.2 Machine Learning

It's possible that there are correlations in the dataset that are not distinguishable by humans. Treating the problem like a black box and using a deep neural net it might be possible to gain a deeper understanding of trends and stronger correlations. In addition - we believe this is also a strong candidate for a Bayesian inference problem.

2.3 Public Accessibility

Conceivability all the work that has been done could be rolled into a pay to access platform that users can subscribe to help make further informed trading decisions.

3 Conclusion

A lot is happening in the crypto currency landscape right now, as newly opened exchanges have given people access to invest in the idea of a decentralised currency. This gives reason to believe that in addition to understanding the tech, it's also interesting to understand what people are thinking about the tech. This is important because the majority vote will determine the most widely adopted currency.

This was partially demonstrated in the work done above, but there's much more work that can be done . If we are to better understand public perception methods other than just Twitter - we believe it's possible to see an ever better fit. Aggregating a larger dataset could be used to understand the limitations of currencies and how volume spikes effect their network.

4 Appendix

4.1 Time Zone Inferance

There are also highly observable daily oscillations in tweet volume as seen in Figure 6. These can be used to infer time zones of maximum influence.

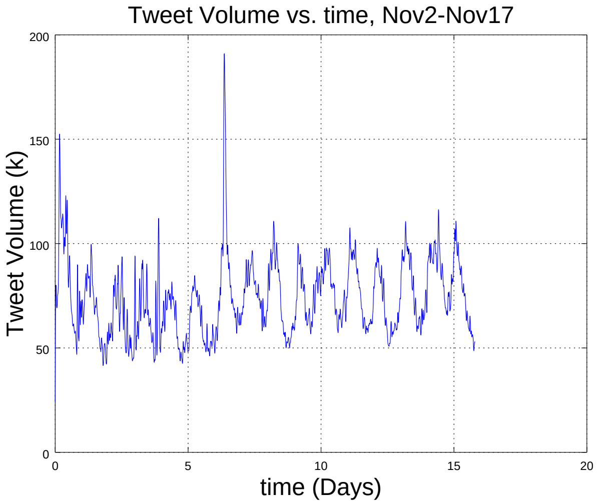

Fig. 6. Tweet volume, Nov2 - Nov17, 2017

4.2 Additional Data

References

[1] Anshul Mitta and Arpit Goel, *Stock Prediction Using Twitter Sentiment Analysis*
[2] https://github.com/tweepy/tweepy
[3] http://www.nltk.org/
[4] https://www.gnu.org/software/octave/
[5] https://pushover.net/

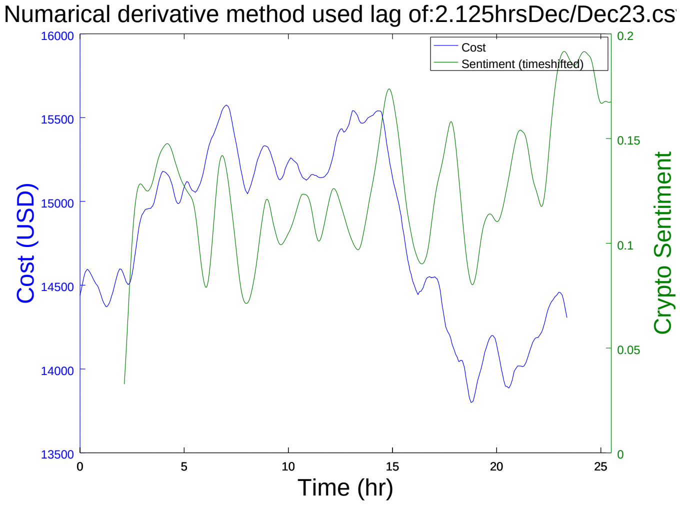

Fig. 7. Dec23

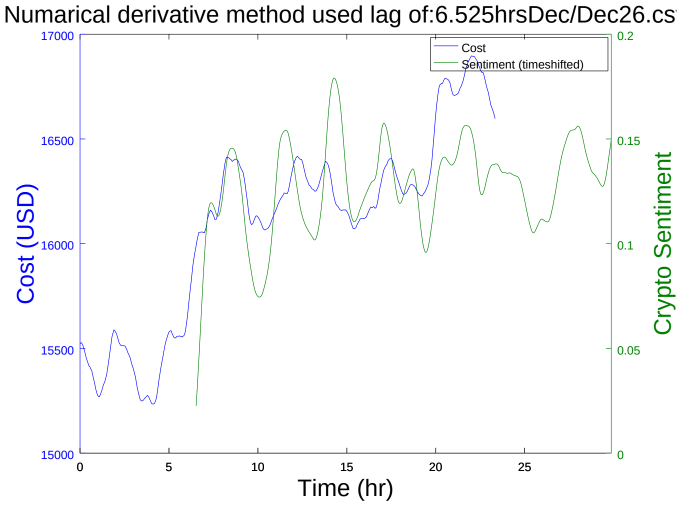

Fig. 8. Dec26

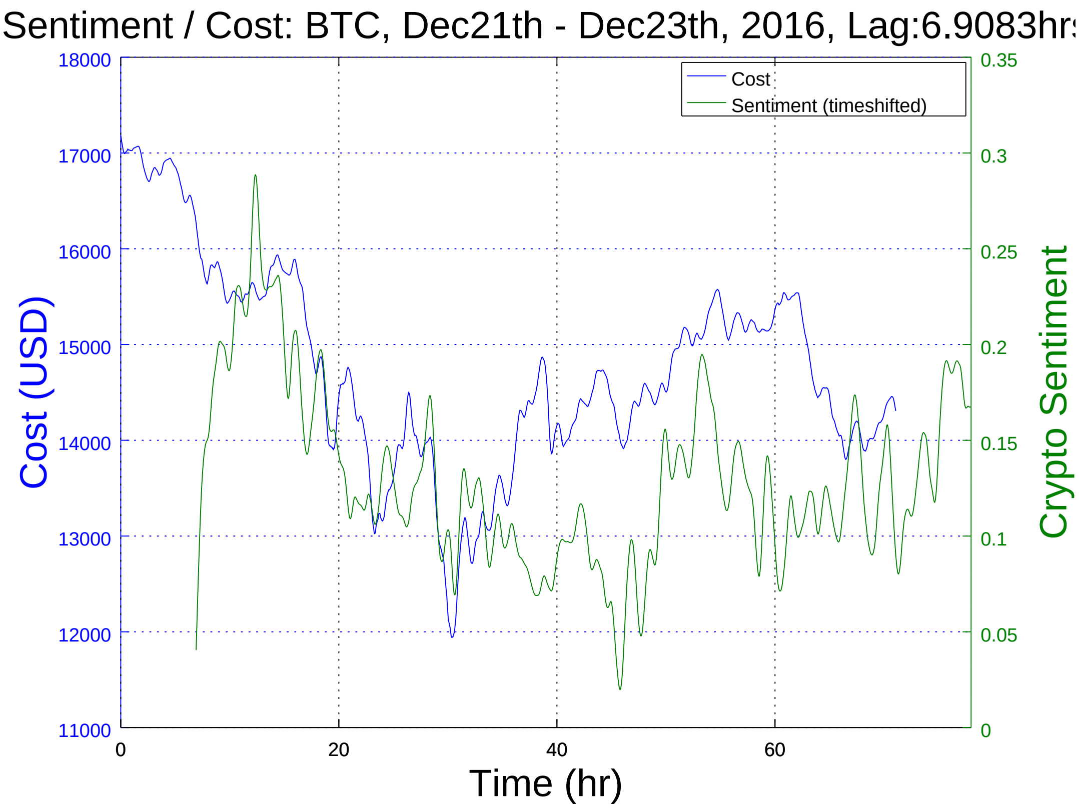

Fig. 9. Bitcoin ’crash’, Dec21 - Dec23, 2017

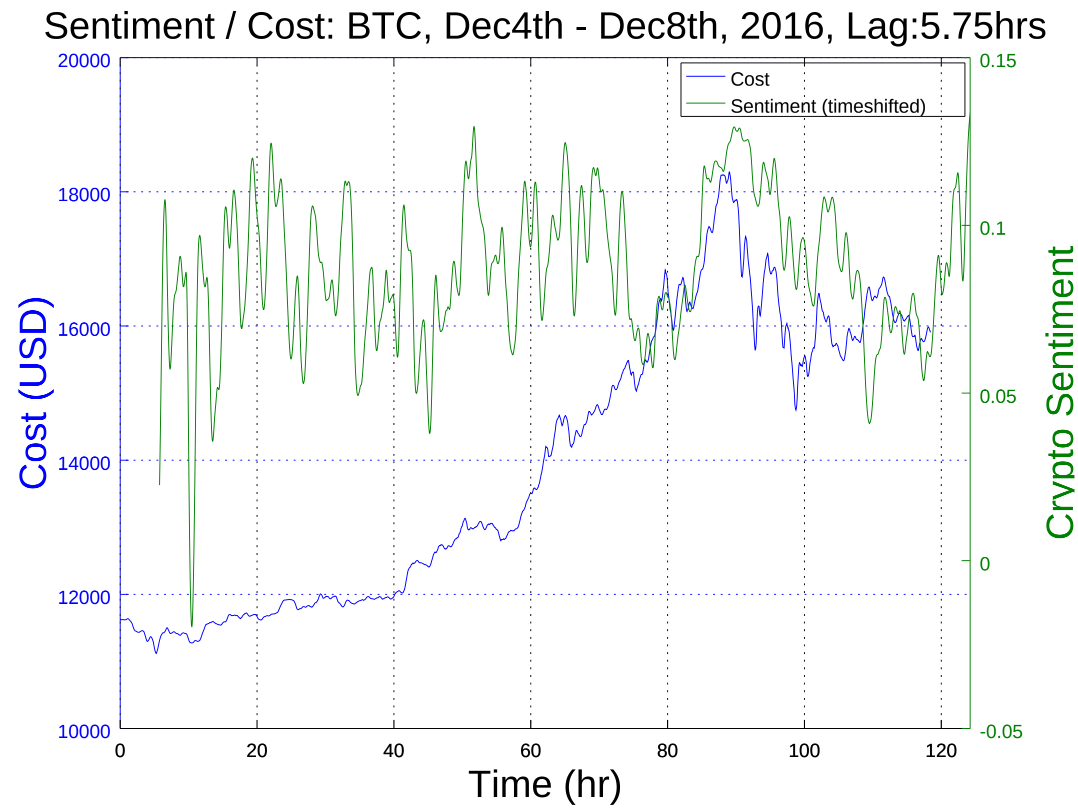

Fig. 10. Four day trend - Dec4 - Dec8, 2017