Twitter Sentiment Analysis for Bitcoin Price Prediction

Abhineet Gupta *(Author)*
Dept. of Computer Engineering
San Jose State University
San Jose, USA
abhineet.gupta@sjsu.edu

Jigar Soni*(Author)*
Dept. of Computer Engineering
San Jose State University
San Jose, USA
jigarvijaykumar.soni@sjsu.edu

Nikita Sengupta *(Author)*
Dept. of Computer Engineering
San Jose State University
San Jose, USA
nikita.sengupta@sjsu.edu

Vineet Zunjarwad *(Author)*
Dept. of Computer Engineering
San Jose State University
San Jose, USA
vineet.zunjarwad@sjsu.edu

***Abstract* — Previous research has shown that prediction of market movement of financial commodities can be done successfully using Twitter data. This paper aims at providing insight into the aspect of using Twitter data pertaining to Bitcoin and machine learning to develop a predictive trading strategy.**

***Index Terms* —Twitter, Bitcoin, Prediction, Sentiment**

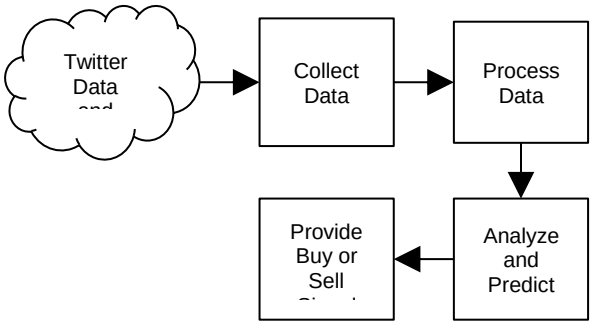
I. Introduction

It's 2017, almost 2.5 million terabytes of information are generated in a day. Each day, 500 million tweets and 1.8 billion pieces of information is shared on Facebook. This information ranges from what someone had in their breakfast to some random funny incident on the street. When it comes to news which is quickly and concisely disseminated, Twitter has become the best choice. Public confidence is of utmost importance for a financial commodity. Using the open APIs provided by Twitter, the shared information on social media can provide a ton of metadata which further provides an insight into what people are thinking. Just like regular currency, Bitcoin (BTC), the decentralized cryptographic currency, is also affected by public opinions; whether those opinions are based on facts, or otherwise. Since its introduction in 2009, Bitcoin rapidly garnered interest as an alternative to regular currencies. Our team has been working to develop a stable trading strategy for Bitcoins since its exchange rates are notorious for being volatile. Our model for calculating and suggesting the best time to trade takes into consideration not just the Bitcoin price trends but also the Twitter sentiment observed during that time. The fact that human opinion is also a part of the equation while coming up with a trading strategy is what makes this project unique. This strategy can be further extended and applied to many other crypto currencies and stocks as well.

II. Problem Statement

While comparing Bitcoin prices with people's opinion on Twitter:
Can we find a relation between Bitcoin price and Twitter sentiment?
Can a Logistic Regression model based on sentiment be used to form a predictive trading strategy?

III. Flow Chart



IV. Data Preparation

To create training and testing data, we have collected data directly from twitter website, as twitter API has limitation of time constraint i.e. we cannot get data more than one week before. So, we have used the python module *Beautiful Soup* to get data from twitter's HTML page. Further, for getting more tweets, we have used *a JSON provider*, using which, we enabled infinite scrolling and collected tweets which contain word "bitcoin". For prediction, we needed real time data and for that we have used an open source python library - Tweepy which also collects tweets based on word bitcoin. Tweets are being collected every minute. All the collected tweets being saved in text file with username, text and time of creation. Further, the prices of bitcoin are being collected every hour and being saved in text file.

The overall data we have collected was limited to 1 million tweets for prediction and after training the model we have collected tweets from tweepy for 45 days. Throughout the process, we needed stable systems and large storage to store the data. For that, we have used services from AWS. That is, for collecting tweets and processing the data operations, we have created EC2 instances and stored the collected data on S3 buckets.

V. Method

For predicting the bitcoin price movement with help of tweets, we have utilized sentiment analysis and text classification algorithms. For text classification, we have implemented logistic regression and support vector machines in Scikit Python library. For sentiment analysis, we have utilized same implementation of the above algorithms.

Logistic Regression

A. *Logistic Regression*

Logistic regression is a Discriminative Learning algorithm. This model creates and examines two classes and determines the separation. This is how it is different from other generative algorithms. Thus, we have used it for text classification. It is using likelihood function as described below:

$$L(\theta) = \prod_{i=1}^{m} (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}}$$

In this equation, $x^{(i)}$ is feature vector. The exponent, $y^{(i)}$, is the state of feature vector i. The index, i, maps the feature vector to the observations in the training set of size m. The function is the sigmoid function below:

$$g(z) = \frac{1}{1+\exp^{-z}}$$

To determine an update rule for the parameters, theta, the log likelihood function can be differentiated to derive the stochastic gradient ascent formula.

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

After reaching convergence, the parameters, are utilized in the sigmoid function to classify the state of currency market.

B. *Training and Testing*

Our training and testing data was a single CSV file containing following headers: *date, sentiment, volume and change.* We have collected all the data and divided it by date it was generated, aggregated the sentiment values based on number of tweets i.e. volume, and determined its feature by the change it has made on bitcoin currency market. The training-testing split was 70-30 percent. That is, 70 percent of the data was allocated for training the LR model and 30 percent was left for prediction.

VI. Sentiment Analysis

A large part of our project research was attributed to this domain with the motive of classifying natural language as either positive or negative sentiment. Our model was capable of both detecting the sentiment intensity and polarity in text. We combined the lexicon based approach and polarity classification which gave us high accuracy in predicting change in trends of public sentiment. For performance testing we applied our calculator model to around 10000 tweets and compared the performance with the existing semantic analysis tools. Our model performed exceptionally well in clean tweets but suffered a hit in performance when tweets consisted too many symbols instead of text. To overcome this, we shifted our focus in reducing noise in the twitter dataset.

Sentiment Analysis Process

Performing sentiment analysis on twitter dataset comprises of three phases: cleaning bot generated redundant data, sentiment analysis of individual tweets and aggregating individual tweet sentiment score into combined score for each day.

Reducing Noise and Cleaning Data Set

While tweets are collected in real time, initial cleaning phase included removal of excess white space and converting text to lowercase.

a) *Removing Duplicates:* Presence of twitter bots complicated our work as they instantaneously disseminated tweets containing keywords which altered words in our training data. To overcome this removal of duplicates was necessary.

b) Filtering: Remove all non-alphabetic words. Invalid English and stop words not having membership in words corpus of the Natural Language Toolkit are also removed. Cleaning phase comprises of data filtering and dropping duplicates.

Aggregating Sentiment Scores

We grouped the individual sentiment scores into time series and for each group we calculated the mean sentiment. The final output was a sentiment vector ordered in time based on past interval length having values mentioned as below: -

{"formatted_text": "two calling fraud corruption", "text": "RT @PhilCrypto77: Two weeks after calling Bitcoin a fraud he's arrested for corruption... https://t.co/6Yn5kFbVVO", "created_at": "2017-11-05T09:10:00", "sentiment": {"neg": 0.559, "neu": 0.441, "pos": 0.0, "compound": -0.5859}, "author": "madman"}

	Initial Tweet Size	**Size after Applying Filter**	**Reduction Percentage**
100K tweets	*100000*	*64231*	*35.7%*
All tweets	*982456*	*653129*	*32.9%*

VII. Deriving Prediction Model

Setting the threshold value

In real time scenarios the sentiment changes very rapidly. To make our prediction model stable a threshold value was required on our sentiment change vector. If the change occurred greater than the threshold value, then only the reading was taken into consideration. The static thresholds are step of 0.05% in-between values of sentiment change vector.

The main challenge while designing the prediction model was to identify the correlation between Twitter sentiment change and Bitcoin price change. We correlated them by taking two temporal aspects into account- frequency length and prediction shift. For short term predictions we have used time series ranging from 30 minutes to 4 hours in length.

Peeking into the Future

Each time series is evaluated over two different shifts forward: 1, 2 where a shift indicates the event predictions for that time in future, e.g. if a negative event occurs in a test with 60 minutes frequency at 18:30, with shift value as 1, the prediction timestamp is set for the bitcoin price change at 19:30. So this timestamp is identified as a next positive change point of bitcoin price. Important point to note here is we only identify the possible bitcoin price change time periods in near future. In the subsequent sections of this paper the term *predicted bitcoin values* refers to this possible price change timestamp and not the absolute bitcoin price value.

Initial Predictions

Periods sentiment score is done by calculating the difference of sentiment scores between neighboring periods. If the sentiment score has increased i.e. positive rate of change, it is considered as “increased positive sentiment” about Bitcoin. Such events predict an increase in price during future periods. Same methodology is applied for negative sentiment rate. This rate value is deemed to be compared against a static threshold value.

To measure the performance of our prediction model we used historical data to compare with our identified price change points. The Error Matrix defined below identifies the four major attributes which are then used to calculate accuracy parameters of our prediction model.

Real Time Historical Data	**Predicted Price (Fluctuation)**	
	Increase:	Decrease:
Increase:	True Positive (TP)	False Negative (FN)
Decrease:	False Positive (FP)	True Negative (TN)

Parameters for Analyzing Prediction Model:

Accuracy: (SUM(TP) + SUM(TN)) / Total Price Values

Sensitivity: SUM(TP) / (SUM (FN) + SUM (TP))

Positive Predictive Value: SUM(TP)/ (SUM (TP) +SUM(FP))

Following table contains the accuracy of the prediction model over the entire dataset, with prediction accuracy depending on threshold for 1 hour and 4-hour intervals.

Freq. Shift	**Accuracy**	**Sensitivity**	**Positive Predictive Value**	**Threshold**
1 hr	0.8254	0.8123	0.787256	2.25
4 hr	0.6521	0.6312	0.666666	0.75

VIII. Plotting Buy Sell Signal Graph

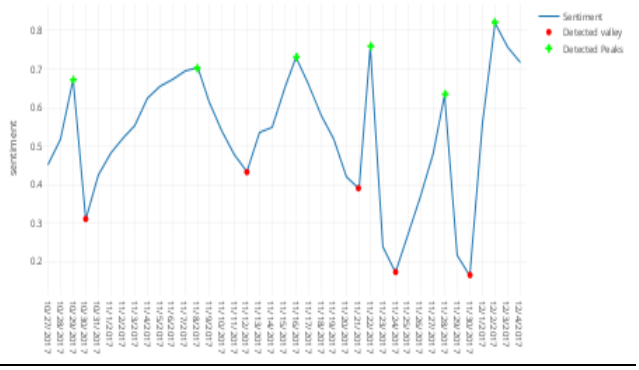
This is the last phase of our project wherein we merge the calculated sentiment vectors and predicted price vectors into one comma separated value file where all components are calculated for a single day. This action is performed by the features module with predicted bitcoin prices, sentiment values as components. This CSV file serves as the input for plotting signal graph.

```
date,sentiment,volume,change
2017110,0.1302217391304347,16288.04830538,-1.7532099
2017111,0.13157192982456145,14672.50624723,-3.99315514
2017112,0.16266134020618553,13981.73283021,-2.29401068
2017113,0.1560315533980582,13375.03001194,-3.40750119
2017114,0.1783660714285714,13296.76313311,0.62668133
2017115,0.18755823529411764,13296.88323793,1.80901457
2017116,0.18632055555555563,12760.56926263,0.89511309
2017117,0.15472786885245904,10882.86542861,42.97384177
2017118,0.1713967289719625,10721.84227939,0.2295919
2017119,0.2151302857142856,10211.39848619,1.2900118
2017120,0.15087617187499994,28860.08719244,0.51804454
2017121,0.16367594339622632,29169.20318198,2.90967485
2017122,0.13212600896860985,29925.18810524,7.48914056
2017123,0.16925650406504059,30991.16393417,0.47514162
2017124,0.1499326771653543,31611.01158756,3.01620353
2017125,0.14012610837438433,31671.54529875,-2.84901515
2017126,0.12460248756218904,32071.64393585,3.4905312
2017127,0.154690366972477,32495.99587765,1.90102839
```

Signal Graph depends on the predicted price and tweet sentiment graphs plotted over time. The final graph gives the end user a signal whether to buy, sell or retain his bitcoin stocks.

+1 value depicts buy signal. 0 value depicts Retain Signal. -1 value depicts sell signal.

Graph showing variation in sentiment for Bitcoin over time:



Graph showing with signals whether to buy or sell BitCoin:

Below signal changes were monitored for 1-hour interval

Graph Axis	**Sentiment Graph Change**	**Predicted Price Graph Change**	**Signal**
Valley	+ve >Threshold	+ve>Threshold	Buy
Peak	-ve < Threshold	-ve<Threshold	Sell
Neutral	Change within Threshold Range	Change within Threshold Range	Retain

IX. Weaknesses

Static Threshold

Currently, we have used static threshold for prediction model which has some limitations because of the static nature. One alternative possible solution was to check for patterns during different periods of the day rather sampling a single time i.e identify thresholds at noon, evening and set them dynamically according to historical data.

Lack of Data

Although we found a way to fetch around 1 million old tweets, bypassing the restriction imposed by twitter API. Even this amount of data was not sufficient to predict data with highest accuracy.

Uncertainty in nature of Tweets

Our model predicts if the price will fall or rise i.e. predict only positive or negative change but does not quantify an absolute value. Although it's nearly impossible to predict exact value but still dependency on higher fluctuations could indicate higher uncertainty in our model.

X. Conclusion

We studied that performing sentiment analysis on twitter data can serve as a predictive basis to give end user a signal to buy or sell his bitcoin stocks. A prediction model was presented, based on the changes in the intensity of sentiment vector from one interval to next. We tested and experimented our prediction model on different time intervals ranging from 30 minutes to 4 hours on a historical data set.

Our prediction model achieved maximum accuracy under 1-hour interval and gave us an 0.8123 (81%) accuracy value. Logistic regression model that we have used has given Mean Square Error of 33 on 70-30 split over 653K records. Further improvements would begin with changing the static threshold value to a dynamic one which will take dynamic nature of cryptocurrency terms into account.

XI. Acknowledgment

We would like to thank Prof. Rakesh Ranjan for his comments and insights which helped us to think from the perspective of solving a problem rather than just a technical solution.

XII. References

[1] Dr Tamer Samee, "Can Twitter sentiment analysis predict BitCoin price fluctuation?", published June 2017.

[2] Stuart Colianni, Stephanie Rosales, and Michael Signorotti, "Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis", Dec 12, 2015.

[3] Sam Couch, on his blog,"Understanding Cryptocurrencies with Sentiment Analysis".

[4] Jermain Kaminski, "Nowcasting the Bitcoin Market with Twitter Signals", Cornell University, submitted.

[5] Jaroslav Bukovina and Matúš Martiček "Sentiment and Bitcoin Volatility", Mendel University in Brno.