Discriminative Aggregation of Social Network Sentiment Data for Cryptocurrency Price Prediction

Emile Clastres
Stanford University
clastres@stanford.edu

Boris Beltinoff
Stanford University
borisoff@stanford.edu

Abstract

1 Introduction

Cryptocurrencies such as Bitcoin, Ethereum and Litecoin have risen to a prominent role in the space of speculative trading. What sets them apart from other instruments is that they are traded 24/7 around the globe, and also have a high volatility, which makes them especially attractive. Yet their fundamental value is difficult to gauge, and they are often seen as purely speculative assets.

Moreover, the emergence of purely speculative cryptocurrency Dogecoin in 2013 was based solely on a meme[1]. The cute dog of Shiba Inu variety was also the base for another meme coin, SHIBA INU, released in 2020; this coin could be considered even more of a meme currency since its inception is linked to Dogecoin, which is already a satirical, albeit functional cryptocurrency.

It is therefore not surprising that online sentiment has been a topic of interest to predict their future price movements. In recent years, researchers have shown that social media sentiment, in particular Twitter, is a good predictor of future returns for alt-coins (1). The very speculative and sometimes purely meme-based nature of many cryptocurrencies might even make crowd sentiment the main price driving force. Similarly to the way bond investors decipher the Fed's every word, it is logical to conclude that social media accounts with the most influence would be influencing the cryptocurrencies the most. A quintessential manifestation of these influences is Elon Musk, whose

[1] https://en.wikipedia.org/wiki/Doge_(meme)

MS&E448: Big Financial Data for Algorithmic Trading, Spring 2021 , Stanford University, CA.

tweets about Dogecoin caused it to jump in price[2], and tweets about Bitcoin first helped the currency[3], then caused it to plummet[4].

Interestingly, all previous research has merely evaluated the predictive power of the aggregate sentiment over all messages posted on social platforms. Yet, social media are characterized by a rich temporal graph structure which could and should be leveraged to get most out of the posted messages. For instance, a bot could post volumes of misleading sentiments, multiple subgroups could each have conflicting independent opinions, or some individuals may have very good predictions and/or may influence the markets directly.

In this project, we lay a framework to exploit the contextual-structural information of online platforms which subsumes the vanilla method of obtaining messages, extracting their sentiment and aggregating it. In essence, we need to define a notion of importance for each message, a measure of contextual independence between messages and a way to extract sentiment signal out of text. Further, we conduct an experiment on Twitter data which demonstrates that correlation gains of around 20% can be made using a parameter-less discriminative aggregation approach.

2 Discriminative Aggregation

In this section, we first formalize the framework of Discriminative Aggregation and then show how the traditional approach is a special case of it.

2.1 Proposed Framework

Let $\mathcal{M}$ be the set of collected messages. Each such message is a pair $m = (\tau, \kappa)$ where τ is the text (content) of the message and κ is its context.

Let $T(m)$ be the time at which m was posted, and let $\Omega = \{\omega_1,\ ..., \omega_n\}$ be the asset universe.

Given all data in $\mathcal{M}$, one has to define :

- $I : \mathcal{M} \mapsto \mathbb{R}^+$ importance function
- $s : \mathcal{M} \mapsto \mathbb{R}^k$ sentiment extraction function
- $a_i : \mathcal{M} \mapsto \{0, 1\}$ attribution function for asset i
- $C : \mathcal{M} \mapsto \{1,\ ..., K\}$ cluster membership function

Additionally, denote $\mathcal{M}^{(i)} = \mathcal{M} \cap a_i^{-1}(1)$

Denote $\forall j \in \{1,\ ..., K\}, \forall i \in \{1,\ ..., n\},\ C_i^j = \{m \in \mathcal{M} : C(m) = j \wedge a_i(m)\}$

The resulting $n \times K \times k$ dimensional signal vector is then defined as, for each timestep t:

$$y_{i,t}^{(j)} = \frac{1}{|C_i^j|} \sum_{\substack{m \in C_i^j \\ t-1 \leq T(m) \leq t}} I(m)s(m)$$

The vanilla sentiment analysis framework is a special case where context is ignored. It can be defined as:

- $I(m) = 1$
- $s : \mathcal{M} \mapsto \mathbb{R}$ sentiment extraction function (usually vocabulary based)
- $a_i(\tau, \kappa)$ is true if asset i is mentioned in τ
- $C(m) = 0$

[2]https://www.msn.com/en-us/autos/news/elon-musk-tweets-about-dogecoin-and-price-immediately-jumps/ar-BB1gI47z

[3]https://www.cnbc.com/2021/05/24/bitcoin-price-soars-after-elon-musk-tweet-on-sustainability-efforts.html

[4]https://www.cnn.com/2021/06/04/investing/elon-musk-bitcoin-breakup/index.html

3 Methodology

In this section, we present the Discriminative Aggregation scheme we used in our experiment on Twitter data. It is simple by design and parameter-less. The first subsection presents the specificity of Twitter data and how context was defined. The next one provides our definitions of I, s, a_i and C.

3.1 Twitter data

Twitter is a social media with a two-fold graph-like structure. Only registered users can post tweets which may or may not contain media. They form a network where each user is uniquely identified by its id. Users follow each other, thus forming a directed graph. They each have a number of followers and a number of favorites, which measure their centrality in the network.

Tweets can be posts, retweets or comments to other tweets. Each tweet is identified by a unique id, and it is possible to retrieve the number of retweets, of favorites, the language and many more tweet attributes. Tweets which are comments to other tweets are flagged as such and the id of the commented tweet is available.

We chose to simply use the user id of each tweet as context. Therefore, we have $\kappa = u$, and we denote $U(m)$ the user corresponding to message m. Downstream, we cluster users rather than tweets, and each tweet is assigned to its poster's cluster. Thus, we obtain communities of people giving their opinion on cryptocurrencies.

3.2 Topic-based Discriminative Aggregation

Our DA scheme is defined as follows:

- $I(m) = 1$
- $S : \mathcal{M} \mapsto \mathbb{R}^2$ sentiment and volume
- $a_i(\tau, \kappa)$ is true if asset i is mentioned in τ
- $C(m) =$ topic user clustering

Topic user clustering is a clustering on users where the proportion vector of each user's past tweets about each coin is used as coordinates for a Nearest Neighbor clustering procedure (by default 8 clusters). Each message is then assigned its user's cluster.

Volume for asset i is defined for each user as the proportion of its tweets in a certain time period which are related to asset i. More formally,

$$v_{i,t}(m) = \frac{1}{N} \sum_{\substack{m' \in \mathcal{M} \\ U(m')=u \\ t-1 \leq T(m') \leq t}} a_i(m')$$

where N is the number of terms in the sum, and $u = U(m)$

Sentiment s for τ is defined as the normalized sum of word sentiment for words in τ, using a dictionary, averaged over all tweets a user posted over a certain period of time.

$$s_{i,t}(m) = \frac{1}{N} \sum_{\substack{m' \in \mathcal{M}^{(j)} \\ U(m')=u \\ t-1 \leq T(m') \leq t}} a_i(m')$$

where N is the number of terms in the sum, and $u = U(m)$

Superscript (j) refers to the restriction of the quantity to messages posted by users in cluster j.

The resulting signal S is the vector $[s_i, v_i]$.

Below is the dictionary used for sentiment analysis[5]:

[5] http://stanford.edu/class/msande448/2019/Final_presentations/gr4.pdf

positive_words = ['conviction','bold','up', 'buy', 'bullish', 'bull', 'free money', 'long', 'rise', 'boom', 'bid', 'support', 'grow', 'get', 'make', 'earn', 'investment', 'invest', 'investing', 'invested', 'buying', 'bought', 'pump', 'like', 'skyrocket', 'enthusiastic', 'optimistic']

negative_words = ['scam', 'capitulation', 'down', 'fork', 'sell', 'short', 'bear', 'bearish', 'bubble', 'stop', 'crash', 'clamp', 'shut', 'freeze', 'fall', 'bust', 'trash', 'forbid', 'oppose','dash', 'sold', 'selling', 'collapse', 'plummet', 'plunge']

The downstream signal extractor is simply the nested average of all components related to asset i:

$$\hat{y}_{i,t} = \mathbf{MEAN}_m^{(j)} \frac{1}{2}(s_{i,t}^{(j)}(m), v_{i,t}^{(j)}(m))$$

In what follows, the vanilla baseline will be the same quantity, with no clustering ($K = 1$)

4 Experiment & Technical Limitations

4.1 Data collection

In order to conduct our analysis, we gathered 1.5 million tweets using a Twitter scraper called TweetScraper[6], which was running from an Ubuntu virtual machine. This scraper is not based on Twitter's APIs, which frees the scraper from the API limitations, such as number and frequency of requests, necessity of credentials, or direct restrictions on searches.

Instead, TweetScraper uses Twitter's built-in search function, which it queries presumably using simple http requests. The resulting search queries and their results are indistinguishable from the native search results in every practical way. Thus, the only limitations on search queries were those we imposed ourselves, e.g. temporal, filtering by minimum number of likes, comments, or replies, and limitations of our file systems.

We ran over 40 queries for top 20 cryptocurrencies by market capitalization[7] for one year April 2020-2021: one query using the full name of the coin, and a query using the corresponding ticker. The number of results varied from about 500 (shiba+inu+crypto) to over 446k (bitcoin) tweets. Each scraped tweet produced a JSON file, along with corresponding user information, also in JSON format. We then repackaged each query's JSON files into one JSON file per query in order to enable the VM's filing system to move the data out for analysis.

Due to overlap of search results, overall we complied 770k unique tweets, and 127k unique users.We further filtered for excessive hashtags, English as the language, and limited to only the cryptocurrencies for which we had hourly pricing data. The best pricing data source that we've found was `https://www.cryptodatadownload.com/`, which only contained 9 pairs we were interested in. Their data was still not exempt of missing entries, and we've filled forward prices where needed (at most three times in a row). The resulting dataframe still had "holes" - hours for which we don't have any data. We decided not to manage that, and there are cases where next-hour returns can be 4-hours-later returns instead, somewhat reducing signal quality.

4.2 Cluster analysis

Characteristically, the clusters, combined due to their mentions of various currencies, look and feel different. We aggregated their texts into one word cloud per cluster in order to glance what their content look like. They do capture different groups of users.

A wordcloud of all tweets gathered is presented in Figure 1. It features many coin symbols and some usual crypto-related lexicon such as DeFi, NFT, fees, trading.

Below in Figure 2 we have Cluster 3, which seems to be composed of hype following users, with an emphasis on the doge meme and its eponymous coin, as well as Elon Musk. We had to add a key word $crypto$ in order to avoid encountering pure meme tweets. Similarly, we've had to deal with ambiguous coins such as LINK, VET, ATOM, UNI, THETA etc. which can cause word searches to return unrelated tweets. Where possible we've filtered on other non-ambiguous keywords.

[6]https://github.com/jonbakerfish/TweetScraper

[7]https://coinmarketcap.com/

Figure 1: Wordcloud of all messages

Figure 2: Wordcloud of cluster 3's messages

These different user clusters indeed present different sentiment dynamics. Below in Figures 3 through 5 are featured graphs showing cluster average sentiment correlation to returns at different horizons for each coin. Visibly, different groups yield better signal for different coins and at different time scales. This fact highlights the opportunities discriminative aggregation could bring.

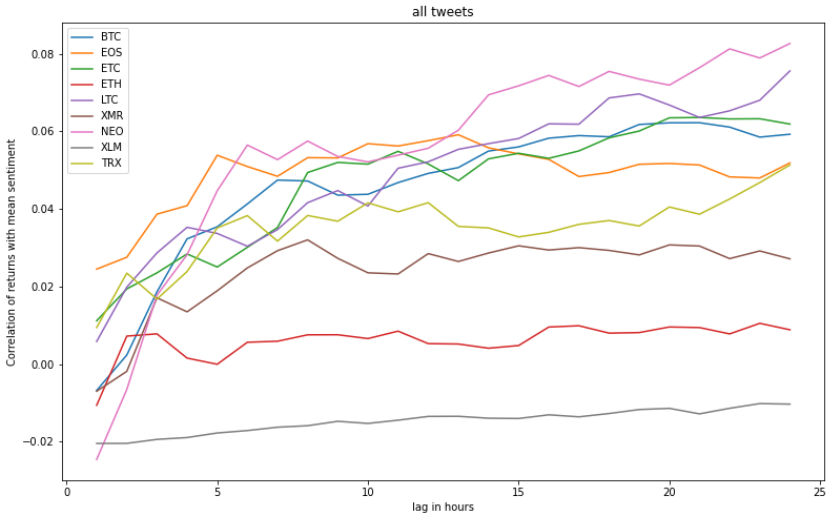

Figure 3: Correlation between sentiment and coin returns at different timescales (all tweets)

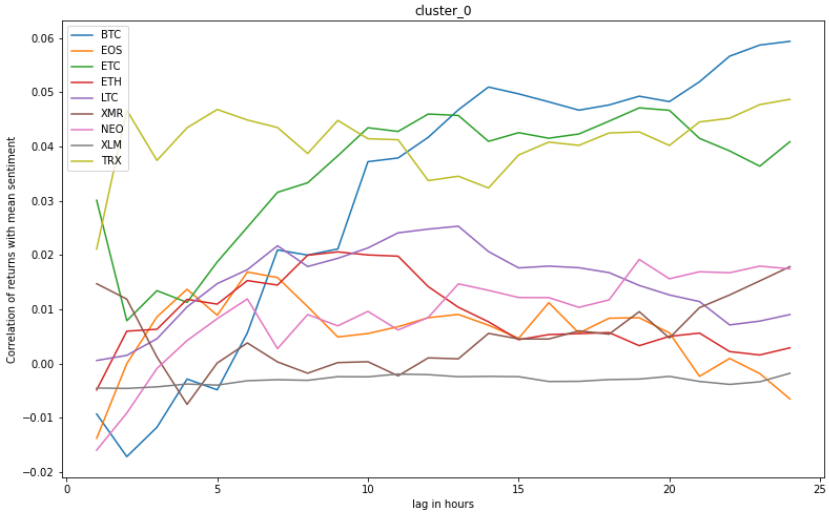

Figure 4: Correlation between sentiment and coin returns at different timescales (Cluster 0)

4.3 Information gains & limitations

To evaluate the signal quality, we compare its correlation to 18-hours returns on the 9 cryptocurrencies we have the price in USD for. on average, we get a 20% increase in correlation strength by clustering users clusters (Table 1). In essence, we are giving equal weight to each group of users rather than to each tweet, and this distinction increases signal quality to the point for ETH that we excavate signal where mean sentiment was uncorrelated to prices.

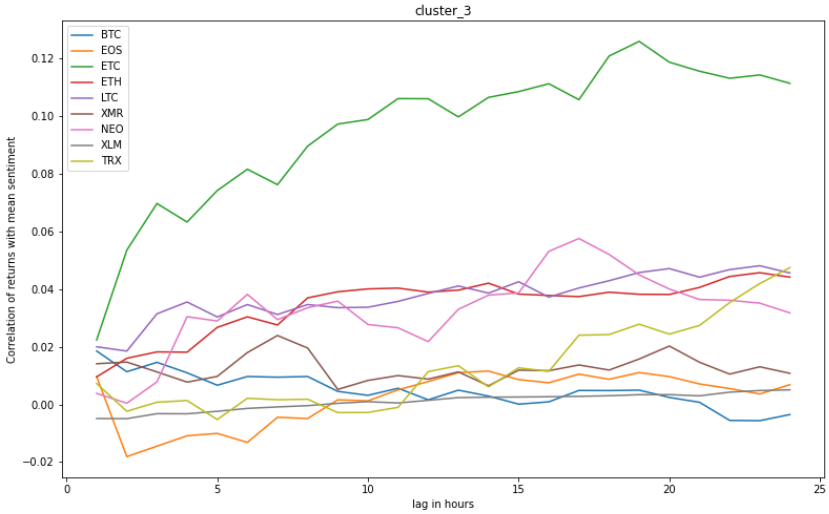

Figure 5: Correlation between sentiment and coin returns at different timescales (Cluster 3)

Table 1: Signal correlation to 18-hour returns

	Discriminative	Vanilla
BTC	**0.0570**	0.0556
EOS	**0.0235**	0.0203
ETC	**0.0262**	0.0249
ETH	**0.0424**	-0.003
LTC	**0.0787**	0.0608
XMR	-0.007	**0.0047**
NEO	**0.0196**	0.0158
XLM	**0.0494**	0.0377
TRX	**0.0170**	0.0105

However, due to the poor quality of the Twitter scraped tweets and the prices data, we can't yet use more complex models. Indeed, the tweet search is non-exhaustive and a lot of noise needs to be filtered out. Additionally, joining hourly prices resulted in many missing time steps, which we didn't handle at all. Hence, sometimes the next returns refers to several hours ahead instead of the very next, and statistical models tend not to respond well. Further, our sentiment definition results in very sparse sentiment, and models tend not to be stable. this is illustrated by the fact that even using a linear regression on $[s_i, v_i]$ to predict prices severely under-performed out-of-sample. As a consequence, it is difficult for now to envision more complex aggregation schemes or even actual strategies. The data is not available on-line and is tedious to scrape. One would need professional grade access to data to actually build ambitious models on top of Twitter data.

5 Conclusion/Future work

We have proposed Discriminative Aggregation as a generalization of traditional sentiment analysis for price prediction, leveraging the rich contextual data of social networks to augment the quality of sentiment signal. Our experiment shows that even a simple parameter-less approach yields consequential improvements of signal quality. Yet, open data access is limited and ambitious systems have yet to be built. We believe that DA is a fertile framework that will allow investors to handle social data more finely than in the past, without too much complexity. Recent advances in Natural Language Processing, Temporal Graph Analysis and Geometric Deep Learning can and should be

included as modules of future DA frameworks, paving the way to new advances in the handling of complex relational textual data.

References

[1] T. Li, A. Chamrajnagar, X. Fong, N. Rizik, and F. Fu, "Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model," *Frontiers in Physics*, vol. 7, p. 98, 07 2019.