Supplementary Information for:

When Can Social Media Lead Financial Markets?

Ilya Zheludev[1*], Robert Smith[1], Tomaso Aste[1]

[1] UK PhD Centre in Financial Computing, University College London, Gower Street, London, WC1E 6BT, UK.

[*] Correspondence and requests for materials should be addressed to IZ. (ilya.zheludev.09@ucl.ac.uk)

Data Twitter Collection Framework systems developed for the study

The TCF, used for the collection of Twitter Data for the study is a Java platform residing within the Eclipse Integrated Development Environment. The package was created to simultaneously connect to Twitter's Sample API and to Twitter's Filter API using a multithreaded approach. This permits the user to record both *all* of the data pushed through Twitter's 10% elevated Gardenhose feed, and data which matches a particular set of filters. These filters can either be keywords, e.g. "iPhone", or pairs of longitude and latitude coordinates which bound a geographical area from which Tweets should be recorded. For example, the coordinates '40,-74' and '41,-73' represent the South-West and North-East coordinates which bound New York City. The framework stores the Twitter Data outputs to .txt files, which must then be read and analysed in a separate environment, e.g. in MATLAB.

The TCF resides within a single dedicated server. The TCF's functionality is controlled by a XML file, which is a list of string and/or geographical-location filters which define the criteria by which the TCF filters Twitter's incoming data streams. By using this XML control file, the TCF is able to filter incoming Tweets based on the locations they are sent from, and/or string combinations. String combinations can be in the form of:

1. AND statements, e.g. "$AAPL" AND "apple".
2. OR statements, e.g. "work" OR "play".
3. Combinations: ["$AAPL" AND "apple"] OR ["work" OR "play"].

The TCF requires an internet connection that can support the delivery of 10% of all Tweets in real time, otherwise a backlog occurs and not all Tweets are delivered on time. Twitter's API documentation strongly advises against such situations since a connected system's sustained inability to process all Tweets being fed through will result in a forced disconnection. Such disconnections would require the system to automatically reconnect – a process which must follow strict (and often difficult to implement) guidelines to avoid permanent barring by Twitter's network. To minimise any sources of incompatibility between the TCF and Twitter's API requirements, the platform used Twitter4j, a highly-capable open-source Java Library designed specifically to deal with connections, disconnections and reconnections to Twitter's

API. By implementing this library in the TCF, we are able to guarantee a reliable connection to Twitter's Gardenhose feed thus minimising the accidental and unnecessary omission of any raw Twitter Data.

Provided that the internet connection powering the TCF is of sufficient bandwidth, the platform can comfortably deal with Twitter's 10% Gardenhose feed. Its maximum collection and *sentiment* classification performance has been benchmarked at c.13,000 Tweets per second. At the time of writing, Twitter's 10% Gardenhose feed delivered up to a maximum sustained *volume* of c.460 Tweets per second; this can however peak at c.4000 Tweets per second in exceptional circumstances for periods of up to a few seconds.

The TCF stores each filter's results to a .txt file with a filename prefix as defined within the XML control file. The platform appends this prefix with the date/time-stamp of creation, allowing for easy identification of results files.

Sentiment and message-*volume* analysis tools

A quantitative assessment of the moods of Tweets can be programmatically ascertained by using an automated *sentiment* analysis system. The chosen *sentiment* analysis tool for this study is the research-orientated SentiStrength package as it is particularly suited to the task of meaning-extraction from Tweets. The package is capable of dealing with the colloquial and often grammatically and lexically incorrect language employed by Twitter users. It is also capable of assigning *sentiment* to emoticons; dealing with misspellings; and most importantly dealing with the effects of negation words such as "not" and "never". Furthermore, since the system is based on dictionary-term matching, it is completely transparent, meaning that the process behind the generation of a *sentiment* score for each piece of text can be viewed by the user. The system has been found to outperform baseline competitors in terms of the accuracy of ranking the *sentiment* of social media vernacular found on MySpace pages[1], and more recently in ranking the *sentiments* of YouTube video comments, Tweets, and online posts on the Runner's World forum[2].

The TCF framework was developed in such a manner that incoming Tweets were parsed by SentiStrength at the point of collection. To accomplish this, the SentiStrength package was

configured for multithreaded use. We configured SentiStrength with the lexicon as at 16th October 2012 to its default settings without setting additional parameters. Most significantly, in this configuration the package thus takes into account the negation of text by assigning negative *sentiments* to terms which are preceded by negators such as "not". For each incoming Tweet, the TCF stored the date/time-stamp of creation, and SentiStrength's *sentiment* outputs which consisted of a Positive *Sentiment* (i.e. how positive a string of text is) and a Negative *Sentiment* (i.e. how negative a string of text is). Positive *Sentiments* are ranked on a scale of +1 (least positive) to +5 (most positive); and Negative *Sentiments* are ranked on a scale of +1 (least negative) to +5 (most negative). The subtraction of the Negative *Sentiments* from the Positive *Sentiments* for a given string results in its overall Net *Sentiment*, which is ranked on a scale of -4 (most negative) through 0 (average) to +4 (most positive). The data produced by all three scoring systems were considered in our study.

Data processing frameworks developed for the study

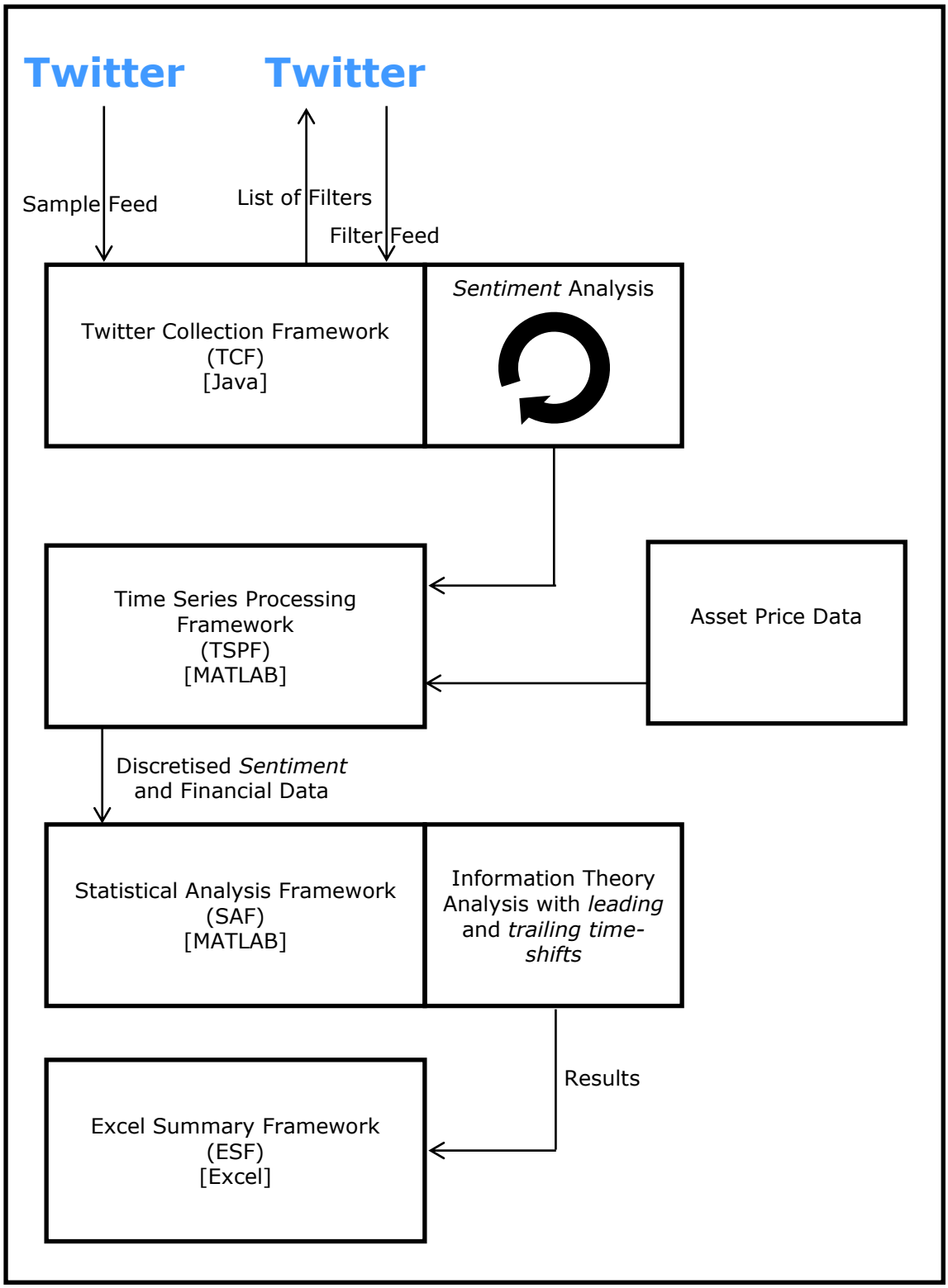
A series of analysis frameworks were designed for use in this study. The TCF's end output is .txt files which contain *sentiment* scores for each message, which require further analysis.

A MATLAB framework was created for post-processing the .txt data dumps created by the TCF. Called the "Time Series Processing Framework" (TSPF), the platform is capable of aggregating/discretising the *sentiment* data produced by the TCF into one-hour non-overlapping windows by way of mean averaging; creating a Net *Sentiment* score for each Tweet (Positive *Sentiment* - Negative *Sentiment*); pulling in financial price data from our data providers; calculating the changes in *sentiments*; calculating the changes in asset prices; and preparing these time-series for further analysis. The TSPF is also capable of processing Tweet message *volumes* rather than Tweet *sentiments*. This was used in our study to evaluate if Tweet *sentiments* showed stronger abilities to lead financial data than Tweet *volumes*.

Finally, another MATLAB framework was created for conducting analyses between the discretised *sentiment* data and the financial data. Called the "Statistical Analysis Framework" (SAF), the platform performed analyses based on Information Theory on the time-series. The platform also instituted a range of *time-shifts* between the *sentiment* data and the financial

data to allow us to assess the degree to which *sentiment* data can lead financial data. Finally, the SAF was also capable of producing a metric for the statistical-significance of our results. This was achieved by permutating the actual *sentiment* data against the financial data 10,000 times, and comparing the resulting *randomised mutual information* values against our *observed mutual information* values. The results produced by the SAF were then fed into a custom-built VBA-driven Microsoft Excel Workbook for final summarisation & graphing purposes, denoted the "Excel Summary Framework" (ESF). The SAF is also capable of processing Tweet message *volumes* rather than Tweet *sentiments*, to help us evaluate if Tweet *sentiments* show stronger abilities to lead financial data than Tweet *volumes*.

An illustration of the frameworks developed for this study, and their mutual interactions is presented in Supplementary Figure 1.



Supplementary Figure 1: Interactions of Frameworks developed for the study.

Time Series Processing Framework (TSPF)

The Time Series Processing Framework (TSPF) was developed iteratively throughout the study. In its final iteration, the TSPF has the following user-controlled options:

1. *Sentiment* data read-in selection. The user selects if he wishes to read in raw *sentiment* data for a particular filter from the Twitter Collection Framework (TCF) for the first time, or if he wishes to open data from the TCF that has been read-in on a previous occasion. Reading data for the first time is more time-consuming as the TSPF has to convert .txt file data to MATLAB's own .m file data line by line, and this takes place at a rate of c. 2500 rows per second. Reading data any subsequent time is near-instantaneous. There is no limitation on the size of files which may be read-in, or the number of files which can be read in simultaneously for a given filter.
2. Financial data read-in selection. The user selects the underlying file which contains the raw price data for a particular filter. Whilst there is no restriction on the granularity of the financial data that can be used, all financial data considered in this study were presented in 5-minute tick intervals.
3. Discretisation-window selection. The user selects the size of the window for which *sentiment* data and financial data are aggregated. This allows for the conversion of raw data, which is continuous, into discretised time frames. There is no restriction on the size of the windows, but for our experiments we used a window-size of 1-hour. For example, if the user selects the window to be 1-hour in size, the system performs the following steps:

    a. Determine the earliest start-dates and end-dates common to the *sentiment* data and the financial data.
    b. Create a time-vector increasing in 1-hour intervals from the start-date to the end-date. Any period at the end of the time-vector which is not 1-hours in size is deleted, and associated *sentiment* data and financial data are also discarded.
    c. For each time-series, whether *sentiment* data or financial data, determine all data-points which fit within each 1-hour interval.

d. For each 1-hour interval, calculate the mean of the aforementioned data-points mentioned in point (c) to produce a mean *sentiment* data value and a mean financial data value for that interval.
e. Calculates the changes in *sentiment* data and changes in financial data between adjacent windows in the time-series. Throughout the investigation, a change-in window of 1 hour was used, giving us hourly financial returns, and values for hourly changes in *sentiment*.

The TSPF is capable of being operated in a batch manner to process the data for a number of user options, without frequent interaction. Thus, it is possible to supply the raw *sentiment* data and financial data for a particular filter, select a range of discretisation options, and leave the TSPF to process all the data in an automated manner.

The TSPF also calculates the Net *Sentiment* for each Tweet, on top of the Positive *Sentiment* and the Negative *Sentiment* which are also explored in our study. The Net *Sentiment* is calculated by subtracting the Negative *Sentiment* (a value on a scale of +1 [least negative] to +5 [most negative]) per Tweet from the Positive *Sentiment* (a value on a scale of +1 [least positive] to +5 [most positive]), resulting in a Net *Sentiment* value on a scale of -4 [most negative] through 0 [neutral] to +4 [most positive].

The TSPF is also able to perform all the aforementioned steps and calculations using Tweet *volumes* instead of Tweet *sentiments*. This functionality was used during our investigation to ascertain the extent to which social media *sentiment* can provide us with stronger powers to lead the financial data, over and above what is possible by considering simply Tweet *volumes*. Upon request, when operating in the Tweet *volume* analysis mode, the TSPF is also able to calculate the absolute financial returns (rather than actual financial returns), by taking the absolute figure of the returns.

Statistical Analysis Framework (SAF)

The Statistical Analysis Framework (SAF) was developed iteratively throughout the study. Its function is to read in data produced by the Time Series Processing Framework (TSPF), clean

the data and perform Information Theory analyses on the data. In its final iteration, the SAF has the following user-controlled options:

a. Data-cleaning selection. This binary switch controls whether the SAF removes erroneous *sentiment* data and financial data produced due to temporary disconnections of the Twitter Collection Framework (TCF) from the 10% Gardenhose feed.
b. Autocorrelation-removal control. In our study, we identified autocorrelations in the social media *sentiment data*, peaking at a lag of 24-hours. The autocorrelative processes were removed by applying a 24-hour backward-looking rolling simple moving average (SMA) to the social media data. For each element in the social media time-series, this was determined by calculating the mean of the preceding twenty-three data points and the element in question. However, for the first twenty-three entries in the social media data time-series – for which there are less than twenty-four preceding elements – we calculate the SMA for each such entry based on the mean of the element itself and all available chronologically-preceding elements, up until the first in the time-series. For example, for element 13 of the social-media time-series series D: $SMA_{i=13} = \frac{D_{13}+D_{12}+\cdots D_1}{13}$, whilst for element 42 of the social-media time-series D: $SMA_{i=42} = \frac{D_{42}+D_{41}+\cdots D_{19}}{24}$.
c. *Time-shift* control. This switch allows the user to select the amount of *time-shift* instituted by the SAF into the data to create a chronological offset between the *sentiment* data and the financial data. The *time-shift* can be either positive or negative, depending on the disired direction of the offset.

Once the above options are configured, the SAF executes the statistical analyses, and stores the results in a series of .txt files which are subsequently read and interpreted by the Excel Summary Workbook (ESW).

Excel Summary Framework (ESF)

The results of the Information Theory Analyses produced by the Statistical Analysis Framework (SAF) are read by the Excel Summary Framework (ESF). The ESF is a collection

of interlinked Microsoft Excel workbooks that aggregate results data from the SAF based on the Twitter Collection Framework filters. Its primary purpose is to amalgamate and condense the large *volumes* of results produced by the SAF into coherent summaries through the use of automated VBA scripts.

Sentiment data and financial data post-processing

Data collection summary and Tweet ranking example

A series of string and geographical filters were set-up to collect and filter relevant messages from Twitter's 10% Gardenhose feed. The data were collected over a 3-month period from 11th December 2012 to 12th March 2013, encompassing periods of both normal market activity, and holiday-period market activity. The following is an example of the raw Tweet collected using the TCF: "Exxon Mobil disappoints, shares down 3.6% premarket. $XOM". SentiStrength, the classifier used in the investigation ranked this Tweet as having a Positive *Sentiment* score of +1 (on a scale of +1 to +5), and a Negative *Sentiment* score of +3 (on a scale of +1 to +5). After subtracting the Negative *Sentiment* score from the Positive *Sentiment* score, this gives a net score of -2 on a scale of -4 (most negative) to +4 (most positive).

Overview of time-series comparison methodology

Before the correlations between the Twitter Data and financial data time series can be established, it is important to consider the chronological frame of reference. Due to the lack of availability of historic Tweets, part of the study had to involve the collection of Twitter Data. This collection period lasted three months, and encompassed both global holiday and high-activity periods, providing a varied source of both Twitter Data *volumes* and financial data *volumes*. Secondly it is important to note that the instantaneous change in the *sentiment* of Twitter Data is unlikely to be coupled with an instantaneous change in financial data, and that for ease of mathematical manipulation, such data had to be discretised. In our study, we discretised our *sentiment data* and financial data into non-overlapping adjacent chronological windows of 1-hour in size. Here, the *sentiment* data and the corresponding financial data for each Twitter Filter were aggregated by way of mean averaging into discretised non-overlapping consecutive windows of 1-hour in size.

It is also important to note that because *sentiment* and price are not static variables, it is not mathematically-sound to compare them like-for-like. For the purposes of our study, it is the change in *sentiment* that was compared to the change in price of financial assets. In each case, the change in *sentiment* and change price is 1 hour. In a similar manner, we calculate the changes in message *volume* in our Tweet *volume* experiments, and compare these either to the returns or the absolute returns of the corresponding financial instruments.

Once the discretisation levels are established, the relationships between the Twitter Data and financial data were then evaluated.

Information Theory

Information Theory refers to a branch of applied mathematics centred on the quantification of information. Based on probability theory and statistics, the construct has found use in applications requiring signal processing and statistical inference in areas such as but not limited to: finance and engineering.

A key measure used in Information Theory is entropy, which quantifies the uncertainty involved in predicting the value of a random variable. It is defined as:

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) log p(x)$$

Where:

- $\mathbb{X}$ is the set of all messages $\{x_i, \ldots, x_n\}$ that X could be.
- $p(x)$ is the probability of some $x \in \mathbb{X}$.
- $I(x)$, the self-information, is the entropy contribution of an individual message.
- $\mathbb{E}_X$ is the expected value.

This quantification of information is applied to the notion of Mutual Information. This is a measure of the amount of information which can be obtained about one random variable by observing another. The Mutual Information of variable X relative to variable Y is given by:

$$I(X;Y) = \mathbb{E}_{X,Y}[SI(x,y)] = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

Where:

- SI, the Specific Mutual Information (a measure of association) is the Point Mutual Information (PMI). The PMI of a pair of outcomes x and y belonging to discrete random variables X and Y is given as:

$$pmi(x;y) = log \frac{p(x)}{p(x)p(y)}$$

The computation of entropy, which is necessary as part of the process for calculating Mutual Information, is based on the probability of the values within the dataset being investigated, as shown in the definition of Information Theory. We estimate such probability distributions using a histogram as this is a computationally-efficient methodology. We select histogram bin size using Sturges' Histogram Rule[3], a well-known method for histogram binning. Sturges' Histogram Rule is defined as:

$$\omega = \frac{r}{1 + log_2(n)}$$

Where:

- r is the range of values within the dataset.
- n is the number of elements in the dataset.
- ω is the ideal bin width to be used for the histogram.

Determining the Mutual Information between social media data and financial data allowed us to quantify the amount of information Twitter Data contains about the future performances of financial instruments, without having to identify the dataset's mean-variance characteristics, or identify their theoretical probability distributions.

Crucially, it is also necessary to determine if *sentiment* data leads financial data, thus giving an indication of whether Social Media Data can be used to lead financial markets. By

quantifying the Mutual Information between the *sentiment* data and the financial data at different *time-shifts*, it is possible to compare how much information Social Media Data contains about the future performance of financial markets. For the notion that Social Media Data can lead financial markets ahead of time to have validity, the quantity of Mutual Information between the *sentiment* data and the financial data must be greater at a chronological *leading time-shift* between the two datasets than at no *time-shift*. Furthermore, the Mutual Information between the *sentiment* data and the financial data at a *leading time-shift* must also exceed what is available between the two time-series or at a chronological *trailing time-shift*. Finally, the Mutual Information between the two time-series must be statistically-significant. To achieve this, we permutate the observed *sentiment* data (for each *sentiment* type: positive, negative, or net) with respect to the financial data 10,000 times and calculate the *randomised mutual information* at each permutation for a given financial-instrument/Twitter-Filter combination for each *leading time-shift* from 0 hours to 24-hours. In such a manner, we are then able to determine the *mean randomised mutual information* for each financial-instrument/Twitter-Filter combination over the 10,000 permutations. Thus, for each *leading time-shift* we are able to calculate the frequency at which the observed mutual information between the *sentiment* data and the financial data exceeds the *mean randomised mutual information* over the 10,000 permutations. With a statistically-significant confidence interval of 99%, we therefore reject those *leading time-shifts* for which the observed mutual information between the *sentiment* data and the financial data is less than the *mean randomised mutual information* 99% of the time.

1. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. & Kappas, A. Sentiment strength detection in short informal text. *J. Am. Soc. Imf. Sci. Technol.* **61**, 2544-2558 (2010).

2. Thelwall, M., Buckley, K. & Paltoglou, G. Sentiment strength detection for the social web. *J. Am. Soc. Imf. Sci. Technol*. **63**, 163-173 (2012).

3. Sturges, H.A. The Choice of a Class Interval. *JASA* **21**, 65-66 (1926).