Sentiment analysis and stock evolution forecasting

Nicolas Lassaux

December 11th 2015

Contents

1 Problem description

1.1 Formalisation of a stock Time Series

We are going to consider a stock Time Series as the daily evolution of the stock's price from the opening of the stock market until its closing.

For example, o_i and c_i are respectively the stock's opening and closing values for day i. We will consider the TS : $c_0 - o_0$, $c_1 - o_1$, ..., c_n - o_n, the list of daily variations. As $o_i = c_{i-1}$, because no price evolution occurs when stock markets are closed, we can use $e_i = c_i - c_{i-1}$.

1.2 Background information

1.2.1 Random Walk and stocks value evolution

Scientists and economists have two opposed major theories about stocks behaviour. One part of them affirm and try to prove that stock's values follow a Random Walk behaviour, that means, for all $i \neq j$, that e_i is independent from e_j. Others refute this idea and assign to stock values a non-random distribution. Plenty of papers support one of these two hypothesis[1].

A funny application of this question is the Turing Random Walk Test: it consists in showing a stock's value plot and a generated Random Walk plot to a human and let him guess which one is the stock value plot.[2].

We will not take position in favor or against one of these hypothesis. In this experiment we will consider the TS forecasting from past values to not be practicable. Even if we could do it, the forecasting would not be simple, if not, some people would be way too rich thanks to the discovered method. We will avoid the problem and use attributes that could follow a Random Walk behaviour too to forecast next values.

1.2.2 Sentiment analysis

People's sentiments can be used to understand global opinion about a brand, an event, etc. A sentiment analysis function f is function defined as:

- s is a sentence
- $x, y \in R^+$ are respectively positive and negative sentiment scores
- $f(s) = (x, y)$

[1] Use of Random Number Generators tests on stock evolution to bring forward the non-random behaviour: http://www.turingfinance.com/hacking-the-random-walk-hypothesis

[2] http://www.argentumlux.org/documents/random.pdf

For example, f("What a great day") $= (2.5, 0)$ and f("I'm not sure you have had a great idea") $= (1, 1.5)$ are possible outputs of a such function.

To compute sentiment scores of a sentence, a basic solution is to attribute a sentiment score to all words and to sum them for all words of a sentence. We could have f("good") $= (1, 0)$ and $f(s) = \sum_{i \in s' words} f(i)$.

All $f(i)$ values are found in datasets like the ones proposed by SentiWordNet[3], that computes a word's sentiments with machine learning algorithms and offer various different vocabularies sentiments. It is the one used by most scientists. Without explaining deeply, I've used a more complex version of this basic algorithm. Words can have different sentiment scores in function of their use in a sentence. For example "good" is more positive as adjective than as noun. The algorithm has to main steps:

- Compute words use probability by regarding the whole sentence (e.g "good" in a particular sentence, P("good is a noon") $= 0.12$ and P("good is an adj")$= 0.88$.
- For each possible function of each word, we look for sentiment scores of the word considering its function.
- For each word, we weight the sum of all possible scores with function probabilities.
- We sum all sentences' words sentiment scores.

We can observe that a higher sentence's size is likely to increase sentiment scores. We will have to normalize our scores.

[3] http://sentiwordnet.isti.cnr.it/

1.3 Project objectives

The main idea of this project is come from the next idea: if we cannot forecast a stock value with past values, with high-speed Internet and high-speed information transfer, maybe social networks can bring forward some warning signs of a stock value evolution. We would like to use sentiment analysis to catch these warning signs.

Different models can be tried to gain information on stock values:

- Linear Regressions between daily sentiment scores the number of tweets and daily total stock evolution.
- Logistic Regression, to try to forecast if the stock will in general increase or decrease during the next seance. Indeed, we try to forecast $P(e_i > 0)$, because it would mean we can earn money by buying now. We could count processing taxes with some transformations: $P(e_i - t > 0)$ or $P(e_i \times (1 - t) > 0)$ with t a constant price or a percentage.

One difficulty will be to find the good lag. In effect the information is not processed instantly by investors, and they can wait before taking action.

After trying to manually create a lag to perform a Linear Regression based on last q values of the other TS, we used a Var(q) model. It will be explained in the part treating the chosen model and the methodology.

Ideally, we could take a stock's history and simulate on a period some stock purchases and sales only based on our estimations and observe our gains (or losses).

1.4 Data gathering and processing

1.4.1 Social Network sentiment - Twitter

We have coded our own Python script to gather tweets. It is a Jupyter notebook (you can explore both the code and its outputs) available on my GitHub account[4].

We will not explain deeply how we have used the API, just for the big picture, we have collected all tweets about stocks with the search function[5], by specifying a day and moving this day. The request mention the stock's abbreviation, like $GOOG for Google, $GPRO for GoPro or $NFLX for Netflix.

We can notice two things :

- We have used $ words instead of # words like #GOOG, #GPRO. Twitter has introduced the $ mentioning especially to be used in tweets related to stock markets.
- The search API does not return all tweets, but only a part of them. The ones returned are the most shared (retweeted) or that belong to the most known authors. It can be seen as bias source but also as a noise filter, since it's more unlucky we receive some spam or some false information.

We have gathered data for Google, Netflix and Apple, as they are quite popular.

For each, after processing, available data are:

- Date
- Num (number of gathered tweets)
- Pos (Total positive score of all tweets of the day related to the stock)
- Neg (Total negative score of all tweets of the day related to the stock)

We added two values: NormPos and NormNeg.

$$\text{NormPos} = \text{Pos}/\text{Num}$$

$$\text{NormNeg} = \text{Neg}/\text{Num}$$

We create one CSV file for each stock, so that we can use them with R.

[4] https://github.com/nico401/stocks-sentiment-analysis/blob/master/exploration.ipynb
[5] https://dev.twitter.com/rest/reference/get/search/tweets

1.4.2 Financial stock historical - Yahoo Finance

Yahoo proposes to download from its financial API the history of a stock's values[6]. We have chosen to download it directly in CSV format, which is easily readable in R and Python. Below the list of all gathered attributes:

- Date
- Open (stock value on session opening)
- High (value max of the day)
- Low (value min of the day)
- Close (stock value on session closing)
- Volume (Total exchanged stocks)
- Adj Closed (To adjust the Close value if exceptional events occur, almost always equal to Close value)

We had to compute for each day the value *Diff = Close - Open* and add it to the CSV file.

[6]The API documentation: https://code.google.com/p/yahoo-finance-managed/wiki/csvHistQuotesDownload

1.5 Data description

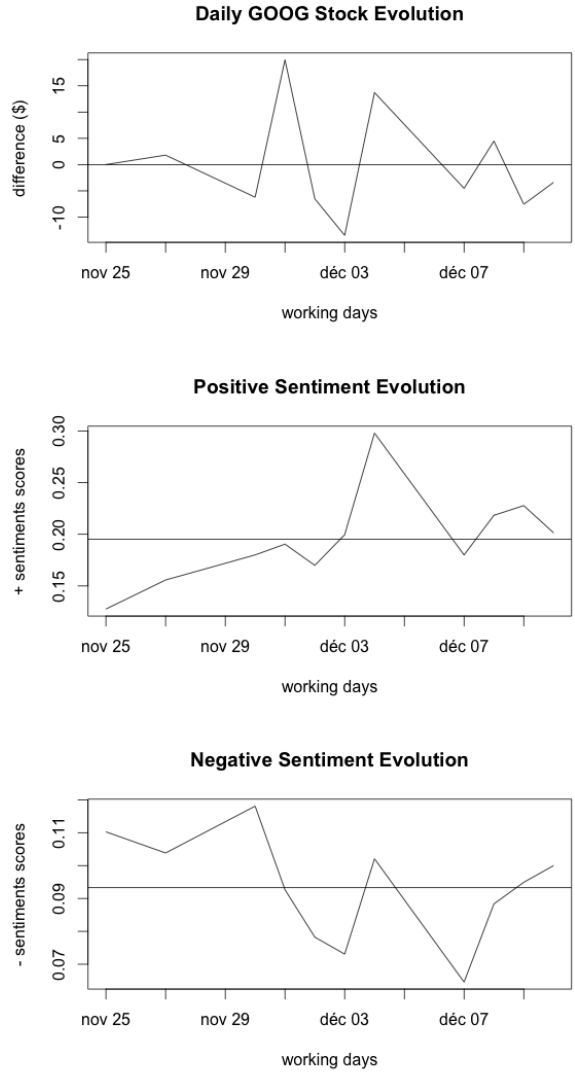

Figure 1: Time evolutions of the google stock and google sentiments

As you can see on Figure 1, we can observe some strong similarities between the three graphs. Unfortunately, the lag between graphs seems to be nonexistent. It would signify we can't really forecast, and we do not know which TS is the cause or the consequence.

	Median	Mean	Variance
Stock value	-3.3900	-0.1582	95.9797
Positive score	0.1905	0.1953	0.0019
Negative score	0.0949	0.0933	0.0003

Table 1: Statistic values

1.6 Model building

The pitfall of our dataset is the low size of the data gathered from Twitter. We have few dimensions, with $k = 2$ or $k = 3$ if we use the number of tweets, the number of days with Twitter data is only $n = 11$.

1.6.1 Logistic and Linear Regressions

First, we can construct a model without any lag. It could show a potential cause-effect between our measures. We must pay attention to the term "potential" because in reality only an experiment could prove a causality effect.

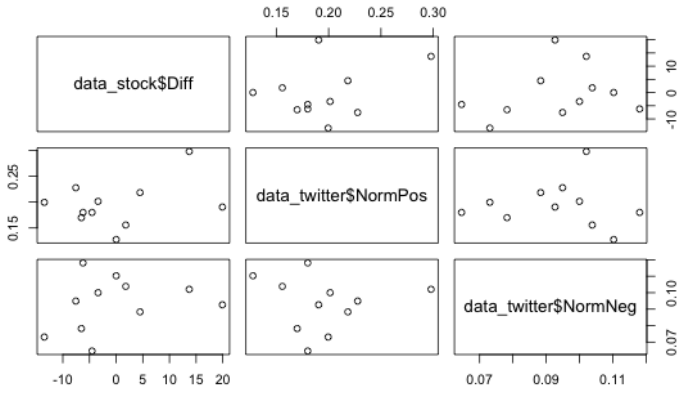

Figure 2: Scatterplot with lag = 0

We can quickly be pessimistic when we observe the scatterplot of Figure 2.

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -27.378191  36.136299  -0.758    0.473
data_twitter$NormPos  69.648512  80.648825   0.864    0.416
data_twitter$NormNeg 158.410159 228.362862   0.694    0.510
data_twitter$Num      -0.002966   0.030716  -0.097    0.926
```

```
Residual standard error: 10.64 on 7 degrees of freedom
Multiple R-squared:  0.1749,Adjusted R-squared:  -0.1787
F-statistic: 0.4946 on 3 and 7 DF,  p-value: 0.6973
```

The Linear Regression with all attributes, the result is very bad. At least we can eliminate the number of tweets with it, because of its p-value of 0.926. Its AIC equals 88.25887. With a negative R-squared, which we can consider as a 0, the whole model can be considered unusable.

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -29.88      23.60  -1.266    0.241
data_twitter$NormPos   72.15      71.50   1.009    0.342
data_twitter$NormNeg  167.53     194.63   0.861    0.414

Residual standard error: 9.956 on 8 degrees of freedom
Multiple R-squared:  0.1738,Adjusted R-squared:  -0.03274
F-statistic: 0.8415 on 2 and 8 DF,  p-value: 0.4659
```

By removing the number of tweets, we reach a AIC of 86.27351. It's a better model but it's still unusable, with a adj-R-squared = 0. It seems obvious we can't use simples Linear regression or Logistic Regression. The best model with the best AIC is the one without any independent variables.

The first idea has been to repeat these models by manually adding a lag (shifting all rows of one vector by q) with R or Python. The solution after some researches seemed to be to use another model, the VAR(q) model. It already uses a lag between some TS and it could both modeling the effect from stock markets to social networks and from social networks to stock markets.

1.6.2 Vector Autoregression model

The VAR(q) model, Vector Autoregression with q lags, can be described by:

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-q} + e_t,$$

We would like to use a VAR(q). We first need to choose the good lag value. We can compute the AIC value for multiple lags and choose the smallest one. Because of the small size of our data, we can't try with large lags. It would have been better to use AICc criterion to take in account the sample size, but used the R package, do not propose this value to choose.

```
  q           1             2
AIC(q) -1.047336e+01 -4.968817e+01
```

```
HQ(q)  -1.104084e+01 -5.068126e+01
SC(q)  -1.021040e+01 -4.922798e+01
FPE(q)  3.453649e-05  1.268420e-21
```

The chosen q is $q = 2$ with the AIC criteron.

```
Estimation results for equation Diff:
=====================================
Diff = Diff.l1 + Pos.l1 + Diff.l2 + Pos.l2 + const

         Estimate Std. Error t value Pr(>|t|)
Diff.l1   -0.8365     0.3647  -2.293   0.0836 .
Pos.l1   -86.4909    69.6332  -1.242   0.2821
Diff.l2   -0.3967     0.2806  -1.414   0.2303
Pos.l2  -117.3645    93.3792  -1.257   0.2772
const     42.8309    21.0192   2.038   0.1112
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 7.492 on 4 degrees of freedom
Multiple R-Squared: 0.7454,Adjusted R-squared: 0.4909
F-statistic: 2.928 on 4 and 4 DF,  p-value: 0.1614
```

The above result about Diff forecasting is the best we can have, and doesn't use the NormNeg value. We can also observe on Figure 3 the lack of spikes after lag=0. We do not have correlation after the normal first spike, its good.

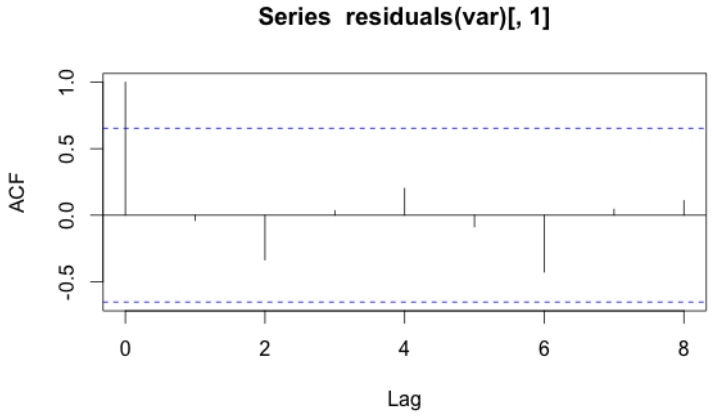

Figure 3: ACF of residuals for Diff

The predict graph drawn on Figure 4 is quite good. It shows us a fitting and a plot of residuals. It is interesting to observe that we could at least forecast if,

for a day i, e_i is going to be positive or negative.

If we set some security conditions, it seems that we could have some earnings greater than 0. Of course, it still need some statistical arguments. A simulation could also be interesting.

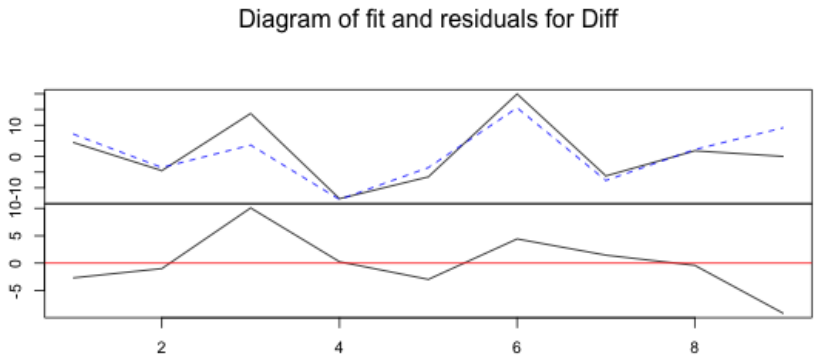

Figure 4: ACF of residuals for Diff

1.7 Conclusion

There are several conclusions to be drawn from this experiment. First, we can't skip the small dataset size. It can be very interesting to enlarge it.

Classical Linear Regression and Logistic Regression are not appropriated to this project. We have better results with a VAR(q) model. In our case its better to look up to two days before. It can be explained, if, for example, investors wait to see others' reaction after an event before acting, introducing a lag. Nevertheless, with a reduced time scale, to an hour for example, maybe we could find even more information.

The sentiment analysis uses a classical vocabulary. We could improve it by using a specialized vocabulary. E.g, does "it will break the resistance" expresses a good or a bad sentiment ? A classical vocabulary will likely tell us it's a bad thing, although it a very good information.

A outlier detection is very important. In the past, some exceptional news have made some stock values to go down for bad reasons[7].

[7] President Obama announced dead and its effect on DOW market. http://www.theguardian.com/business/2013/apr/23/ap-tweet-hack-wall-street-freefall