Forecasting financial time series with machine learning models and Twitter data

Argimiro Arratia
argimiro@lsi.upc.edu
computationalfinance.lsi.upc.edu

Computer Science, UPC, Barcelona

LARCA. Laboratory for Relational Algorithmics, Complexity and Learning
UNIVERSITAT POLITÈCNICA DE CATALUNYA
What should we try to model?

The price of an asset as a function of time constitutes a financial time series.
BUT the price of a reasonable asset shows exponential behavior in time ...
AND an exponentially–increasing t.s. is hard to manipulate (most statistical tools, e.g. correlation, regression, work best with linear functions):
The mean value of an exp–increasing t.s. has no meaning. The derivative of an exponential function is exponential, so day-to-day changes in price have same unfortunate properties.

A BETTER way to represent the data is as a

simple return (of period $\tau$):

\[
R_t = \frac{P_t}{P_{t-\tau}} - 1 = \frac{P_t - P_{t-\tau}}{P_{t-\tau}}
\]
The return is a complete and scale–free summary of the investment opportunity. ( Negative return = the asset declined in value; positive return = increased; zero return = unchanged )
What is the distribution of stock returns?

In practice (see previous figure) the daily return of a stock presents:

- small (sample) mean or expected value
- small skewness, hence some gross symmetry
- more smaller (in value) returns than larger returns

...this suggests a distribution centered around the mean, symmetrical, and tails going out to infinity in both directions (bell-shaped curve)

...a normal distribution?

This would be great: a normal dist. has density funct. totally determined by mean $\mu$ and variance $\sigma^2$

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Argimiro Arratia
Forecasting stock markets
Exploring the possibility of a normal distribution
APPLE returns

Density

-0.15 -0.10 -0.05 0.00 0.05 0.10 0.15

0 5 10 15 20

Argimiro Arratia

Forecasting stock markets
Drawbacks to the assumption of normality

- return of a stock has lower bound -1 and no upper bound
  \[ \therefore \] hard to believe in symmetry with tails going out to \( \infty \)
- returns aggregate multiplicative:
  Holding the asset for \( k \) periods between dates \( t - k \) and \( t \) gives a

\[
1 + R_t[k] = \frac{P_t}{P_{t-k}} = \prod_{j=0}^{k-1} \frac{P_{t-j}}{P_{t-j-1}} = \prod_{j=0}^{k-1} (1 + R_{t-j})
\]

Thus a multiperiod return is the same as multiplying successive simple returns over the period.

But the product of normal variables is not necessarily normal
Assume log–normal distribution

log return

\[ r_t = \ln(1 + R_t) = \ln \left( \frac{P_t}{P_{t-1}} \right) = \ln P_t - \ln P_{t-1} \]

By considering logarithmic returns we turn products of returns into sums

multi-period log return

\[ r_t[k] = \ln(1 + R_t[k]) = \ln((1 + R_t)(1 + R_{t-1}) \ldots (1 + R_{t-k+1})) \]
\[ = \ln(1 + R_t) + \ln(1 + R_{t-1}) + \ldots + \ln(1 + R_{t-k+1}) \]
\[ = r_t + r_{t-1} + \ldots + r_{t-k+1} \]

The sum of log-normal variables is log-normal, and log returns have lower bound 0

However in practice there still is some degree of skewness and extreme values (hence kurtosis)
Some statistical property must be assumed or else there is no possibility of modeling at all.

- Stationarity
- Weak stationarity (more likely)
General model for a financial time series

Let \( \{r_t : t = 1, \ldots, T\} \) be observations of a time series (e.g. returns)

\[
r_t = \mu_t + a_t
\]

where

- \( F_{t-1} \) = information set available at time \( t - 1 \)
- (Conditional mean) \( \mu_t = E(r_t|F_{t-1}) := G(F_{t-1}) \)
- \( a_t \) is stochastic shock or innovation, assumed to have zero conditional mean, and hence
- (Conditional variance) \( \sigma_t^2 = Var(r_t|F_{t-1}) = E(a_t^2|F_{t-1}) := H(F_{t-1}) \)

where \( G \) and \( H \) are well-defined functions with \( H(\cdot) > 0 \).
Consider

- $F_{t-1} = \{r_{t-1}, r_{t-2}, \ldots, r_{t-p}\}$ ($p$ lags)
- $G$ a linear function of $F_{t-1}$, so

$$\mu_t = \phi_1 r_{t-1} + \ldots + \phi_p r_{t-p}$$

- $H$ is constant ($H(F_{t-1}) = \sigma^2$)

We get AR($p$) model: $r_t = \phi_1 r_{t-1} + \ldots + \phi_p r_{t-p} + a_t$

This generalizes to AutoRegressive and Moving Average of order $p$, $q$, ARMA($p$, $q$):

$$r_t = \phi_1 r_{t-1} + \ldots + \phi_p r_{t-p} + a_t + \theta_1 a_{t-1} + \ldots + \theta_q a_{t-q}$$

where $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ are real numbers,
We are concerned with nonlinear models, where $G$ or $H$ are non-linear functions
We look at
- Neural Networks (NNet)
- Support Vector Machines (SVM)
Neural Networks (feed-forward)

**Figure:** A 3-4-1 feed forward neural network with one hidden layer

$x_1, x_2, x_3$ input nodes; $y$ output node;
$h_1, \ldots, h_4$ hidden nodes (neurons) in hidden layer;
$h_j$ goes active and transmit a signal to $y$ if $z_j = \sum_{i \rightarrow j} \omega_{ij}x_i > \alpha_j$.

The signal is produced by activation function (logistic):

$$L(z) = \frac{\exp(z)}{1 + \exp(z)} = \frac{1}{1 + \exp(-z)}$$
Composing a linear combination of possible values of signals from the hidden layer with activation function $f$ for the output node, and direct connections input to output:

$$y = f \left( a + \sum_{i=1}^{p} \phi_i x_i + \sum_{j=1}^{q} \theta_j \mathcal{L} \left( \alpha_j + \sum_{i \rightarrow j} \omega_{ij} x_i \right) \right)$$

$x_1, \ldots, x_p$ input values, $y$ output, $\mathcal{L}$ activation funct. for the $q$ hidden nodes, $f$ (non-linear) activation funct. for the output; $\alpha_j$ threshold, $\omega_{ij}, \phi_i, \theta_j$ weights, $a$ is the bias in the connection. These parameters are chosen so that some forecasting error measure is minimized (more later).
If $f$ is linear the FFNNet becomes an autoregressive moving average model for a given time series $\{r_t\}$:

Take $x_1 = r_{t-1}, \ldots, x_p = r_{t-p}$ (i.e., $p$ different lags of the series), the output $y = r_t$ as the time series value to forecast, and $f$ the identity. Then

$$r_t = a + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j \mathcal{L} \left( \alpha_j + \sum_{i \rightarrow j} \omega_{ij} r_{t-i} \right)$$

(here the moving average part is being modelled by a nonlinear function on the input lags).

Hence, feed forward neural network is a generalization of ARMA$(p, q)$ model.
Support Vector Machines (for regression)

An SVM approximates dataset \( \mathcal{G} = \{(x_k, y_k) : k = 1, \ldots, N\} \) by multiple regressions of the form:

\[
f(x_k, w) = \sum_{i=1}^{D} w_i \phi_i(x_k) + b
\]

where \( \{\phi_i(x_k)\}_{i=1}^{D} \) are the features of inputs, \( w = \{w_i\}_{i=1}^{D} \) and \( b \) are coefficients estimated from data by minimizing the risk functional:

\[
R(w) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i, w)|_{\epsilon} + \frac{1}{2} ||w||^2
\]

with respect to the \( \epsilon \)-insensitive loss function

\[
|y_i - f(x_i, w)|_{\epsilon} = \begin{cases} 
0 & \text{if } |y_i - f(x_i, w)| < \epsilon \\
|y_i - f(x_i, w)| & \text{otherwise}
\end{cases}
\]
Interpreting the approximations $f(x_k, w) = \sum_{i=1}^{D} w_i \phi_i(x_k) + b$ as hyperplane in $D$-dimensional feature space defined by $\{\phi_i(x)\}$

**The Goal:** to find a hyperplane $f(x, w)$ that minimizes $R(w)$

Vapnik (1995) shows such minimum is attained by functions of the form:

$$f(x, \alpha, \alpha') = \sum_{i=1}^{N} (\alpha'_i - \alpha_i) K(x, x_i) + b$$

where $K(x, y) = \sum_{i=1}^{D} \phi_i(x)\phi_i(y)$ is the kernel function, and coefficients $\alpha_i, \alpha'_i$ are Lagrange multipliers obtained by maximizing certain quadratic form
Kernel trick

One does not need to compute the features $\phi_i(x)$ and their inner product, since kernel can be computed alternatively through analytical functions not involving them.

Common choices for kernel

- polynomial kernel (with degree $d$): $K(x, y) = (x \cdot y + 1)^d$;
- Gaussian radial basis function (RBF)
  $K(x, y) = \exp(-||x - y||^2/\sigma^2)$, with bandwidth $\sigma^2$, and
- sigmoid kernel: $K(x, y) = \tanh(\kappa x \cdot y - \sigma)$. 
The quality of being semiparametric or parametric, and the existence of closed form solutions provides two important dimensions to characterize the forecasting models (McNelis, 2005)

<table>
<thead>
<tr>
<th>Closed-form solutions</th>
<th>Parametric</th>
<th>Semiparametric</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Linear regression</td>
<td>Taylor polynomial</td>
</tr>
<tr>
<td>NO</td>
<td>ARCH / GARCH</td>
<td>NNet / SVM</td>
</tr>
</tbody>
</table>
Let \( \{(x_t, r_t) : t = 1, \ldots, T\} \) be the available data. 

\( r_t \) is the return of some financial asset and \( x_t \) is vector of inputs or features:

- lags of the series (its past behavior);
- volume,
- variance (volatility)
- any fundamental indicator of the series (e.g. Price-to-Earnings, Dividend-to-Price)

Model fitting for a NNet or SVM requires division of the data into

**TRAINING** (±75%)  **TESTING** (±25%)
Training

In this step build a few models by choosing the parameters (e.g., weights, the thresholds and connection bias) so that some forecasting error measure is minimized

For NNet: use the mean squared error

$$MSE(w) = \frac{1}{N} \sum_{t=1}^{N} (r_t - modelFit(w, x_t))^2$$

For SVM: use the risk functional

$$R(w) = \frac{1}{N} \sum_{t=1}^{N} |r_t - modelFit(x_t, w)|_\epsilon + \frac{1}{2} ||w||^2$$
Testing

The best fitted model build in training step is tested on the subsample of data reserved for testing to predict some values and compare estimations with actual sample values. Usual measures of forecasting accuracy

\[ MSE = \frac{1}{N} \sum_{t=1}^{N} (r_t - \text{pred}(r_t))^2, \quad MAE = \frac{1}{N} \sum_{t=1}^{N} |r_t - \text{pred}(r_t)|, \]

\[ RMSE = \sqrt{MSE} \]

But some experts recommend to use

**Normalized RSME**

\[ NRSME = \frac{\sqrt{MSE}}{\sqrt{\frac{1}{N} \sum_{t} (r_t - \hat{\mu}_r)^2}} = \sqrt{\frac{SE}{(N - 1) \text{Var}(r_t)}} \]
Packages: **e1071** for SVM; **nnet** for neural networks; **kernlab** for both; **caret** for data handling functions

File: **sp500m** contains monthly readings of the price of S & P 500 Composite Index from January 1900 to January 2012

Run two experiments:
Target: the log return of S&P 500 for periods $\tau = 1$ (monthly) and $\tau = 12$ (yearly)
Features: lags 1, 2, 3, 5 of the series

We build only one SVM and one Nnet (due to time constrains) but you should improve the code with an iterative scheme to tune models (caret package has functionalities for doing this tuning)

Parameters to tune: (SVM) gamma, cost, kernel (Nnet) size of hidden layer, decay factor.
<table>
<thead>
<tr>
<th>Machine</th>
<th>NRMSE $\tau = 1$</th>
<th>NRMSE $\tau = 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM, Radial</td>
<td>2.12 (bad)</td>
<td>0.116 (good)</td>
</tr>
<tr>
<td>$C = 10^{4.5}, \gamma = 10^{-2}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNet</td>
<td>0.97 (bad)</td>
<td>0.112 (good)</td>
</tr>
<tr>
<td>size = 6, decay = $10^{-2}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Are stocks return predictable?

Eugene Fama says NO

And has been saying so since 1970 (∗), supporting the

**Efficient Market Hypothesis**

The price of a security at any time fully reflects all available information; hence, it is impossible for investors to make a profit above the average market returns.


and got the Nobel Prize in Economics in 2013 for that!

Argimiro Arratia

Forecasting stock markets
Are stocks return predictable?

Robert Shiller says **PERHAPS**

At least the direction future long-term returns will take could be known from measures of the structure of speculative bubbles. And Shiller and coauthors have developed various indicators to explain and model the *Irrational Exuberance* of Mr. Market.


and got the Nobel Prize in Economics in 2013 for that!
<table>
<thead>
<tr>
<th>Title</th>
<th>Computational Finance: An Introductory Course with R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Argimiro Arratia</td>
</tr>
<tr>
<td>Series</td>
<td>Atlantis Studies in Computational Finance and Financial Engineering 1</td>
</tr>
<tr>
<td>Editor</td>
<td>Argimiro Arratia</td>
</tr>
</tbody>
</table>

**ISBN**: 978-3-030-60593-7

**Publisher**: Atlantis Press

**Series Editor**: Argimiro Arratia
Motivations

Does a public Sentiment Indicator extracted from daily Twitter messages can indeed improve the forecasting of social, economic, or commercial indicators, based on time series models?

Answer:

Experiment with all possible machine learning models reinforced with a Sentiment Index built from Twitter data, and compare their performance when trained with and without the Twitter index.
<table>
<thead>
<tr>
<th>Ref</th>
<th>Event</th>
<th>Models</th>
<th>Corpus</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Wolfram 2010]</td>
<td>NASDAQ stocks</td>
<td>SVM</td>
<td>Edinburgh Corpus, English, Relevant to stocks</td>
<td>Works with high freq. data. No sentiment analysis, but direct count of frequency of words</td>
</tr>
<tr>
<td>[Zhang et al. 2010]</td>
<td>DJIA, S&amp;P500, NASDAQ, VIX</td>
<td>n/a</td>
<td>English, with mood keywords</td>
<td>Finds correlations of tweet’s emotions (hope, fear, worry) and the direction of the DJIA stock index.</td>
</tr>
<tr>
<td>[Bollen et al. 2011]</td>
<td>DJIA</td>
<td>SOFNN</td>
<td>~10M tweets, Stock market prices</td>
<td>An index of the calmness of the public is predictive of the DJIA and predictions can be significantly improved using a SOFNN.</td>
</tr>
<tr>
<td>[Mishne and Glance 2005]</td>
<td>Movie sales</td>
<td>n/a</td>
<td>Blog posts with links to IMDB, IMDB sales data</td>
<td>Considering the sentiment of blog posts improves the correlation between references to movies and their financial success.</td>
</tr>
<tr>
<td>[Asur and Huberman 2010]</td>
<td>Movie sales</td>
<td>Linear regression</td>
<td>~2.9M tweets for 24 movies</td>
<td>The model built with the tweet rate time series outperforms the baseline that uses the Hollywood Stock Exchange (HSX).</td>
</tr>
<tr>
<td>[O'Connor et al. 2010]</td>
<td>U.S. polls</td>
<td>n/a</td>
<td>10^9 tweets (omitting non-English), Public opinion polls</td>
<td>The evolution of Twitter sentiment correlates to periodical public polls on the presidential election and on the presidential job approval.</td>
</tr>
<tr>
<td>[Tumasjan et al. 2010]</td>
<td>German 2009 election</td>
<td>Logistic Regression</td>
<td>100K tweets</td>
<td>Additional information is not provided to predictive models. Only a comparison of the share of voice and the election results.</td>
</tr>
<tr>
<td>[Wakamiya 2011]</td>
<td>TV Ratings</td>
<td>n/a</td>
<td>Japanese tweets</td>
<td>No predictions or correlations are detected.</td>
</tr>
</tbody>
</table>
There are no public data sets available and Twitter have imposed several restrictions on retrieving on-line posted tweets. As of April 2010 Twitter forbids 3rd parties to redistribute tweets. Hence we have to create our own data set, with limitations. Begun on 22 March 2011. Use a Streaming API:

- One HTTP connection is kept alive to retrieve tweets as they are posted
- Filter the stream by keyword or user
The Twitter Gold Mine

You can BUY data from official Twitter reseller. Very, very expensive!

The Twitter-Hedge fund

July 2011: Derwent Capital, a UK based hedge fund, in partnership with Bollen et al. began trading a $40 Million Hedge Fund using the Twitter Predictor.

http://www.derwentcapitalmarkets.com/
http://mashable.com/2011/05/17/twitter-based-hedge-fund/
Figure: Overview of data collection, preprocessing, forecasting and final analysis processes.
Online service that allows to build social networks based on microblogging

Messages or tweets up to 140 characters

Special or reserved symbols: @ for replying; # (hashtags) for topics assignment or keywords; RT (retweet) for sharing with followers; URLs.
Twitter social indicators implemented

- Our **Twitter** based indicators: **Volume** and **Sentiment**
- Hypothesis:
  - (Volume) More messages $\leftrightarrow$ more variance (Volatility)?
- Hypothesis:
  - (Sentiment index) negative/positive $\leftrightarrow$ decrease/increase benefits (returns)?
Sentiment Analysis

Goal: Determine whether a message contains positive or negative impressions on a given subject

- Subjectivity Recognition
- Polarity Detection

- Begin with creating a labelled corpora for supervised training a sentiment classifier

- We apply a recent pictorial tagging idea [Bifet et al, 2010]: Create a dataset of tweets that are automatically labelled positive if contains a smiley of the form: :-), ;-D or negative if contains :-(. (Formally, consider all regular expressions from [:=8][-]?[ )D] for positive, and from [:=8][-]?( for negative.)
Expand the training datasets by Feature Lists of words: classify by frequent words with at least 5 occurrences, and topic (e.g. a stock’s ticker).

Clean the data previously:
- Remove duplicates (mostly in retweets);
- Remove stopwords (e.g. pronouns, prepositions, many verbs);
- Stemming (reduce words to their roots by removing suffixes);
- Negation handling (use tagging: not good → NOT_good);

Relevance filtering: text containing some of the keywords is no guarantee of its relevance for the subject we want to classify. E.g.

“I really love eating an apple”

“Apple stock soared above $404 today”
Sentiment Classifiers (SC)

- We only focus on binary classification (positive/negative)
- 3 classes of data sets to train different SCs: English, Multi-language, Stock. Variations of SC obtained by training on these data sets with different preprocessing, feature words to represent docs, etc.
- Best scoring SC extracted: $C-En$, $C-Ml$, $C-Stk$.
- Accuracy: 76.49% for $C-En$, 79.5% for $C-Ml$, 76% for $C-Stk$
Sentiment Index (or Twitter series)

**Sentiment:** A time series consisting of the daily percentage of positive tweets (over the total number of tweets posted) for each top-scoring SC.

**Volume:** time series of daily number of tweets concerning a subject.
Financial Time Series

Goal
To predict stock’s return or volatility

Focus on following companies and indices: AAPL, MSFT, GOOG, YHOO S&P100 (OEX), S&P100’s implicit volatility (VXO), S&P500 (GSPC), S&P500 implicit volatility (VIX).

Returns
- Computed from Adjusted Close
- Log–normally distributed
- Log returns

Volatility
- Computed from log returns
- Exponential Weighted Moving Average

\[ V(t_n) = (1 - \lambda) \sum_{i=1}^{m} \lambda^{i-1} R_{n-i}^2 \]
Combining all different parameters gives us approximately 39000 experiments.

**Summary trees**

Decision trees built with **REPTree** (Weka), by greedy selection of attributes that give a higher performance gain. The tree gives an indication of the attributes that most influence the increase on accuracy.
Experimental Results (General)

- Big failure of linear models (only improve 2.5% of the time: 64 improved/ 2418 not improved)
- Nonlinear models with either Twitter series improve as predictors of the trend of volatility indices (VXO, VIX) and historic volatilities of stocks.
- Predicting the trend of benefits is more dependent on the parameters, the input data and Twitter classifier. Best case is SVM with poly-2 kernel for predicting OEX with sentiment index obtained from C-Stk classifier.
### Experimental Results (Specific)

Success rates by model family for predicting the VIX index using Tweet Volume and $lags = 3$.

<table>
<thead>
<tr>
<th>Model family</th>
<th>Successful</th>
<th>Unsuccessful</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNets</td>
<td>100</td>
<td>35</td>
<td>74.07%</td>
</tr>
<tr>
<td>SVMs</td>
<td>103</td>
<td>121</td>
<td>45.98%</td>
</tr>
</tbody>
</table>

Success rates of SVM by kernel type for predicting the VXO index when $lags = 2$, plus either Twitter series.

<table>
<thead>
<tr>
<th>Kernel Type</th>
<th>Successful</th>
<th>Unsuccessful</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial Kernels</td>
<td>104</td>
<td>192</td>
<td>35.13%</td>
</tr>
<tr>
<td>Radial</td>
<td>5</td>
<td>70</td>
<td>0.07%</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>68</td>
<td>7</td>
<td>90.67%</td>
</tr>
</tbody>
</table>
The Twitter-hedge fund crash

October 2011: Derwent Capital closes shop with reported returns of 1.86% after a month of trading with Bollen et al Twitter predictor, and is for sale today: http://www.derwentcapitalmarkets.com/auction/

The only machine model used in Bollen et al is a “Fuzzy Logic Neural Network” and they only care about price direction disregarding volatility. We were not able to test such machine (the program for it was of course not available), but in general Neural Network score low in our tests for predicting direction of returns for stocks (and the results of such experiments are very parameter-dependent).
Computational Finance

http://computationalfinance.lsi.upc.edu/