Forecasting financial time series with machine learning models and Twitter data

Argimiro Arratia

argimiro@lsi.upc.edu

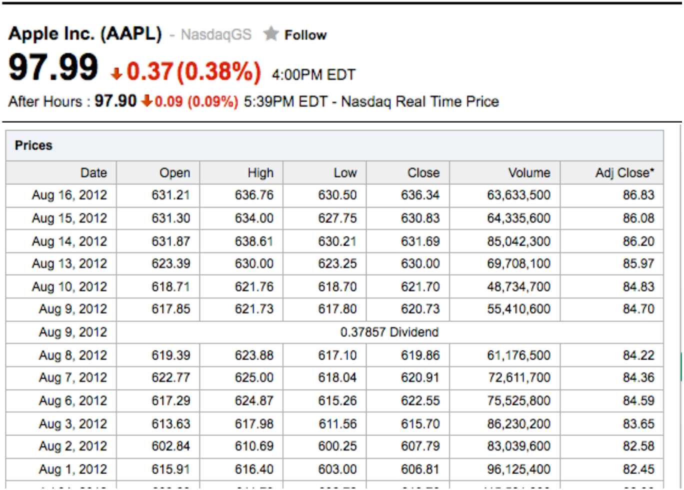
computationalfinance.lsi.upc.edu

Computer Science, UPC, Barcelona

LARCA. Laboratory for Relational Algorithmics, Complexity and Learning

UNIVERSITAT POLITÈCNICA DE CATALUNYA

What should we try to model?



Apple Inc. (AAPL) - NasdaqGS ★ Follow

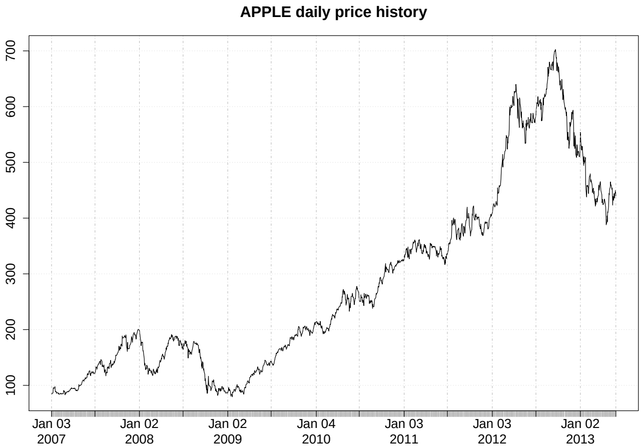
97.99 ↓0.37(0.38%) 4:00PM EDT

After Hours : 97.90 ↓0.09 (0.09%) 5:39PM EDT - Nasdaq Real Time Price

Prices

Date	Open	High	Low	Close	Volume	Adj Close*
Aug 16, 2012	631.21	636.76	630.50	636.34	63,633,500	86.83
Aug 15, 2012	631.30	634.00	627.75	630.83	64,335,600	86.08
Aug 14, 2012	631.87	638.61	630.21	631.69	85,042,300	86.20
Aug 13, 2012	623.39	630.00	623.25	630.00	69,708,100	85.97
Aug 10, 2012	618.71	621.76	618.70	621.70	48,734,700	84.83
Aug 9, 2012	617.85	621.73	617.80	620.73	55,410,600	84.70
Aug 9, 2012	0.37857 Dividend					
Aug 8, 2012	619.39	623.88	617.10	619.86	61,176,500	84.22
Aug 7, 2012	622.77	625.00	618.04	620.91	72,611,700	84.36
Aug 6, 2012	617.29	624.87	615.26	622.55	75,525,800	84.59
Aug 3, 2012	613.63	617.98	611.56	615.70	86,230,200	83.65
Aug 2, 2012	602.84	610.69	600.25	607.79	83,039,600	82.58
Aug 1, 2012	615.91	616.40	603.00	606.81	96,125,400	82.45

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 . . .

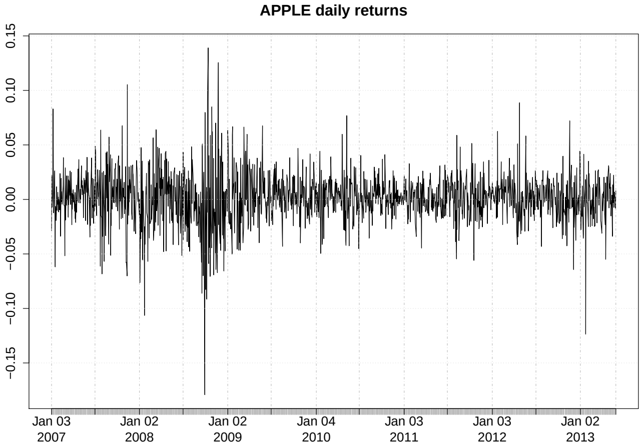
Financial Time Series of Returns

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 τ):

$$R_t = \frac{P_t}{P_{t-\tau}} - 1 = \frac{P_t - P_{t-\tau}}{P_{t-\tau}}$$



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?

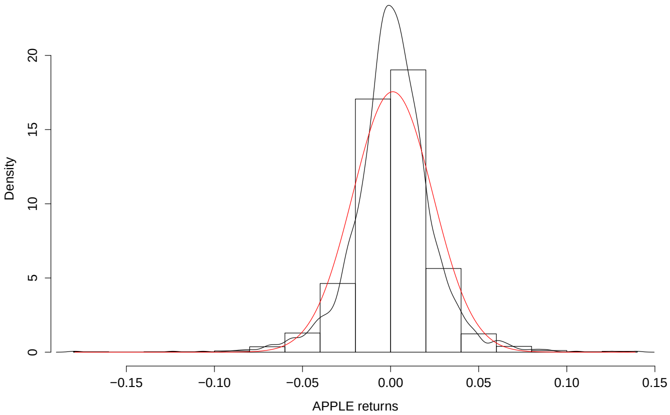
This would be great: a normal dist. has density funct. totally determined by mean μ and variance σ^2

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

R Lab

Exploring the possibility of a normal distribution

R Lab



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 ∞
- 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

multiperiod log return

$$\begin{aligned} r_t[k] &= \ln(1 + R_t[k]) = \ln((1 + R_t)(1 + R_{t-1}) \ldots (1 + R_{t-k+1})) \\ &= \ln(1 + R_t) + \ln(1 + R_{t-1}) + \ldots + \ln(1 + R_{t-k+1}) \\ &= r_t + r_{t-1} + \ldots + r_{t-k+1} \end{aligned}$$

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)

Weak stationarity (at least)

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 $H(\cdot) > 0$.

(linear models) Autoregressive models ARMA(p, q)

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 ϕ_1, . . . , ϕ_p, θ_1, . . . , θ_q are real numbers,

Nonlinear models

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)

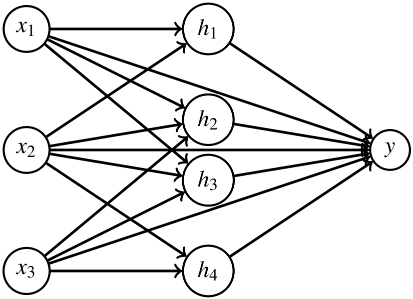


Figure: A 3-4-1 feed forward neural network with one hidden layer

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 $z_j = \sum_{i\to j} \omega_{ij} x_i > \alpha_j$.
The signal is produced by activation function (logistic):

$$\mathcal{L}(z) = \frac{\exp(z)}{1+\exp(z)} = \frac{1}{1+\exp(-z)}$$

Neural Networks (feed-forward)

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:

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

$x_1, \ldots, x_p$ input values, y output, $\mathcal{L}$ activation funct. for the q hidden nodes, f (non-linear) activation funct. for the output; α_j threshold, ω_{ij}, ϕ_i, θ_j weights, a is the bias in the connection. These parameters are chosen so that some forecasting error measure is minimized (more later).

FFNNet Special case: $ARMA(p, q)$

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

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

(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 $\mathcal{G} = \{(\boldsymbol{x}_k, y_k) : k = 1, \ldots, N\}$ by multiple regressions of the form:

$$f(\boldsymbol{x}_k, \boldsymbol{w}) = \sum_{i=1}^{D} w_i \phi_i(\boldsymbol{x}_k) + b$$

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

$$R(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})|_\epsilon + \frac{1}{2}||\boldsymbol{w}||^2$$

with respect to the ϵ-insensitive loss function

$$|y_i - f(\boldsymbol{x}_i, \boldsymbol{w})|_\epsilon = \begin{cases} 0 & \text{if } |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})| < \epsilon \\ |y_i - f(\boldsymbol{x}_i, \boldsymbol{w})| & \text{otherwise} \end{cases}$$

Interpreting the approximations $f(\boldsymbol{x}_k, \boldsymbol{w}) = \sum_{i=1}^{D} w_i \phi_i(\boldsymbol{x}_k) + b$ as hyperplane in D-dimensional feature space defined by $\{\phi_i(\boldsymbol{x})\}$

The Goal: to find a hyperplane $f(\boldsymbol{x}, \boldsymbol{w})$ that minimizes $R(\boldsymbol{w})$

Vapnik (1995) shows

such minimum is attained by functions of the form:

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

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

Kernel trick

One does not need to compute the features $\phi_i(\boldsymbol{x})$ 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): $K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}+1)^d$;
- Gaussian radial basis function (RBF) $K(\boldsymbol{x},\boldsymbol{y}) = \exp(-||\boldsymbol{x}-\boldsymbol{y}||^2/\sigma^2)$, with bandwidth σ^2, and
- sigmoid kernel: $K(\boldsymbol{x},\boldsymbol{y}) = \tanh(\kappa\boldsymbol{x}\cdot\boldsymbol{y}-\sigma)$.

A model typology

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)

Closed-form solutions	Parametric	Semiparametric
YES	Linear regression	Taylor polynomial
NO	ARCH / GARCH	NNet / SVM

Model training and testing

Let $\{(\boldsymbol{x}_t, r_t) : t = 1, \ldots, T\}$ be the available data
r_t is the return of some financial asset and $\boldsymbol{x}_t$ 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 ($\pm 75\%$) TESTING ($\pm 25\%$)

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

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

For SVM: use the risk functional

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

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

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(r_t - pred(r_t))^2, \qquad MAE = \frac{1}{N}\sum_{t=1}^{N}|r_t - pred(r_t)|,$$

$$RMSE = \sqrt{MSE}$$

But some experts recommend to use

Normalized RSME

$$NRSME = \frac{\sqrt{MSE}}{\sqrt{\frac{1}{N}\sum_t (r_t - \widehat{\mu}_r)^2}} = \sqrt{\frac{SE}{(N-1)Var(r_t)}}$$

R Lab for NNet and SVM

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.

Summary of results

Machine	NRMSE $\tau = 1$	NRMSE $\tau = 12$
SVM, Radial $C = 10^{4.5}, \gamma = 10^{-2}$	2.12 (bad)	0.116 (good)
NNet size = 6, decay = 10^{-2}	0.97 (bad)	0.112 (good)

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.

$(*)$ Fama, E., Efficient Capital Markets: A review. *J. of Finance*, 1970

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

Are stocks return predictable?

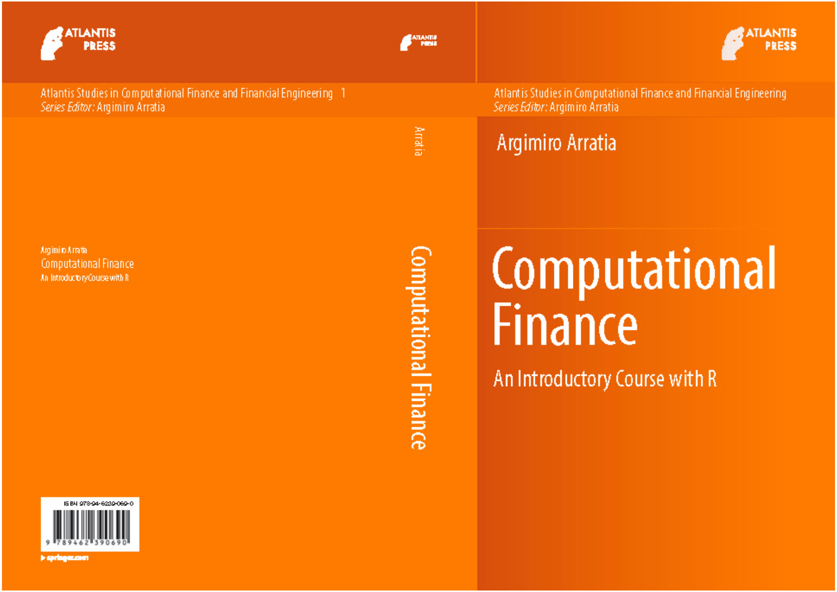
Robert Shiller says PERHAPS

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.

Campbell, J., Shiller, R. Valuation ratios and the long-run stock market outlook. *Advances in Behavioral Finance*, 2, 2005

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

A Break



ATLANTIS PRESS

Atlantis Studies in Computational Finance and Financial Engineering 1
Series Editor: Argimiro Arratia

Argimiro Arratia
Computational Finance
An Introductory Course with R

ISBN 978-94-6239-069-0

9 789462 390690

springer.com

Arratia

Computational Finance

ATLANTIS PRESS

Atlantis Studies in Computational Finance and Financial Engineering
Series Editor: Argimiro Arratia

Argimiro Arratia

Computational Finance

An Introductory Course with R

Forecasting with Twitter (with Marta Arias and Ramón Xuriguera)

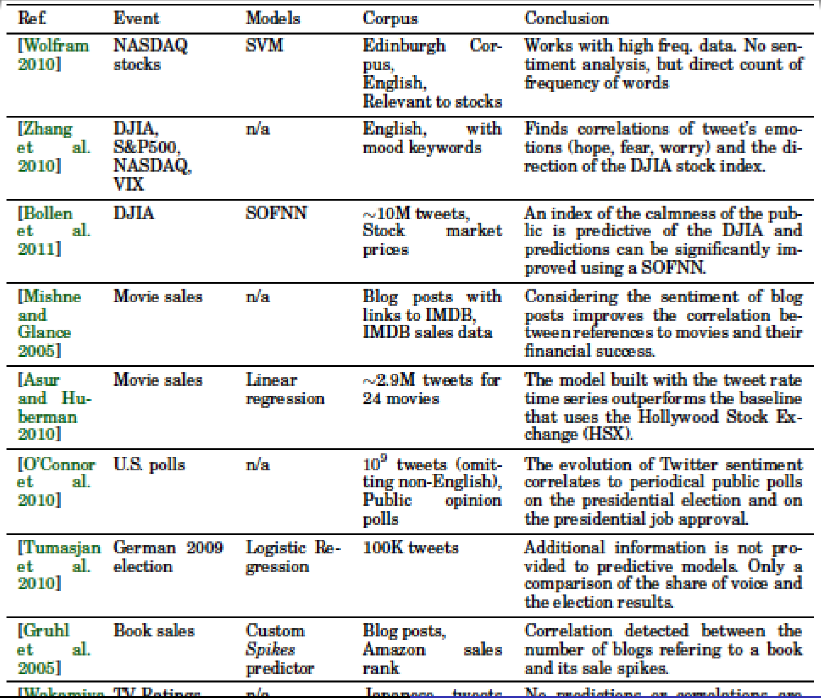
published in: *ACM Transactions on Intelligent Systems and Technology*, 5 (1), 2013.

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.



Ref	Event	Models	Corpus	Conclusion
[Wolfram 2010]	NASDAQ stocks	SVM	Edinburgh Corpus, English, Relevant to stocks	Works with high freq. data. No sentiment analysis, but direct count of frequency of words
[Zhang et al. 2010]	DJIA, S&P500, NASDAQ, VIX	n/a	English, with mood keywords	Finds correlations of tweet's emotions (hope, fear, worry) and the direction of the DJIA stock index.
[Bollen et al. 2011]	DJIA	SOFNN	~10M tweets, Stock market prices	An index of the calmness of the public is predictive of the DJIA and predictions can be significantly improved using a SOFNN.
[Mishne and Glance 2005]	Movie sales	n/a	Blog posts with links to IMDB, IMDB sales data	Considering the sentiment of blog posts improves the correlation between references to movies and their financial success.
[Asur and Huberman 2010]	Movie sales	Linear regression	~2.9M tweets for 24 movies	The model built with the tweet rate time series outperforms the baseline that uses the Hollywood Stock Exchange (HSX).
[O'Connor et al. 2010]	U.S. polls	n/a	10^9 tweets (omitting non-English), Public opinion polls	The evolution of Twitter sentiment correlates to periodical public polls on the presidential election and on the presidential job approval.
[Tumasjan et al. 2010]	German 2009 election	Logistic Regression	100K tweets	Additional information is not provided to predictive models. Only a comparison of the share of voice and the election results.
[Gruhl et al. 2005]	Book sales	Custom *Spikes* predictor	Blog posts, Amazon sales rank	Correlation detected between the number of blogs refering to a book and its sale spikes.
[Wakamiya	TV Ratings	n/a	Japanese tweets	No predictions or correlations are

Some technical difficulties

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

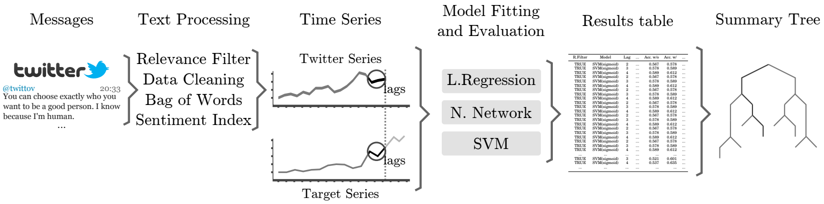
The Twitter Gold Mine

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/

How we did it



Messages

Text Processing

Time Series

Model Fitting and Evaluation

Results table

Summary Tree

@twittov 20:33
You can choose exactly who you want to be a good person. I know because I'm human.
...

Relevance Filter
Data Cleaning
Bag of Words
Sentiment Index

Twitter Series

Target Series

L.Regression

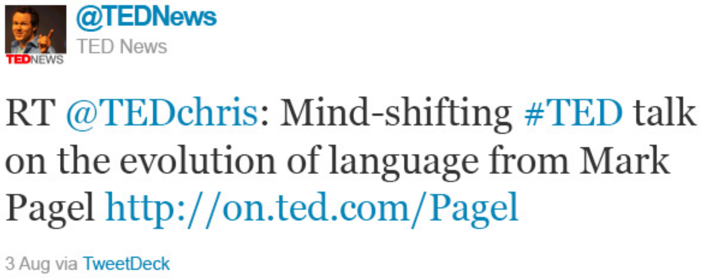
N. Network

SVM

R.Filter	Model	Lag	...	Acc. w/o	Acc. w/	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
...	...	...	...	...	...	...
TRUE	SVM(sigmoid)	3	...	0.521	0.601	...
TRUE	SVM(sigmoid)	4	...	0.537	0.635	...
...	...	...	...	...	...	...

Figure: Overview of data collection, preprocessing, forecasting and final analysis processes.

Twitter



@TEDNews
TED News

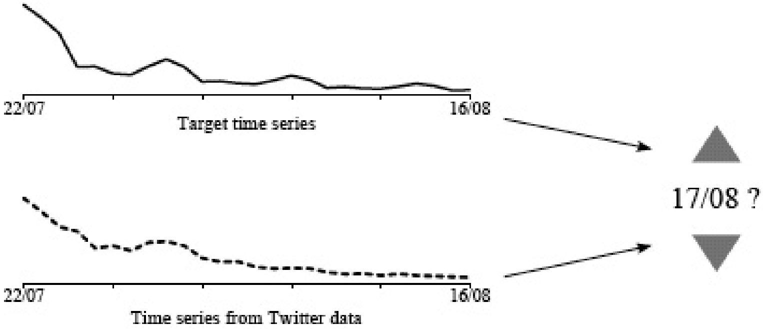
RT @TEDchris: Mind-shifting #TED talk on the evolution of language from Mark Pagel http://on.ted.com/Pagel

3 Aug via TweetDeck

- 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 $\mapsto$ more variance (Volatility)?
- Hypothesis:
  (Sentiment index) negative/positive $\mapsto$ 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.)

Sentiment Analysis

- 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 $\mapsto$ 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

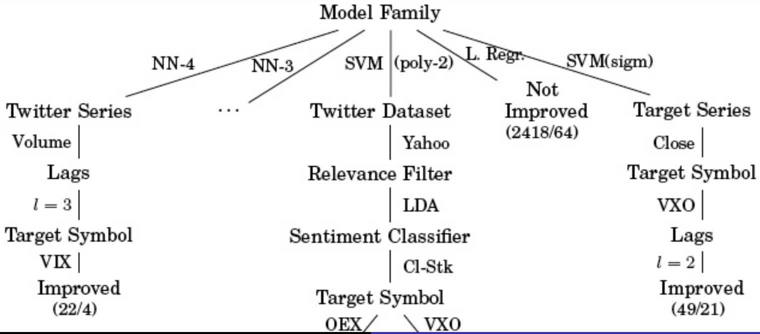
$$V(t_n) = (1-\lambda)\sum_{i=1}^{m} \lambda^{i-1} R_{n-i}^2$$

Experimental set up

Combining all different parameters give us approx. 39000 experiments.

Summary trees

Decision trees built with `REPTree` (Weka), by greedy selection of attributes that give a higher performance gain. The tree give an indication of the attributes that most influent the increase on accuracy.



Model Family

NN-4

NN-3

SVM (poly-2)

L. Regr.

SVM(sigm)

Twitter Series

Volume

Lags

$l = 3$

Target Symbol

VIX

Improved (22/4)

. . .

Twitter Dataset

Yahoo

Relevance Filter

LDA

Sentiment Classifier

Cl-Stk

Target Symbol

OEX

VXO

Not Improved (2418/64)

Target Series

Close

Target Symbol

VXO

Lags

$l = 2$

Improved (49/21)

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 $lags = 3$.

Model family	Successful	Unsuccessful	Success rate
NNets	100	35	74.07%
SVMs	103	121	45.98%

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

Kernel Type	Successful	Unsuccessful	Success rate
Polynomial Kernels	104	192	35.13%
Radial	5	70	0.07%
Sigmoid	68	7	90.67%

WARNING! Don’t play with your home account

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).

THE END

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