Market Predictions using Sentiment Analysis and State-Space Models

Jeffrey Wong

December 10, 2010

1 Abstract

Traders on Wall Street are constantly working and gathering information about a variety of assets. They meet with clients from private companies, read the news, and monitor the global economy. The more information a trader has, the more likely he is to turn a profit. These traders comprise a large portion of market activity, and hence drive stock market movements.

In order to determine whether or not an asset is worth purchasing, we wish to know whether it is overvalued or undervalued. This can be estimated by comparing the current stock price to the current value the traders assign to the asset. If the current price is above this value, we can expect that fast-moving traders will sell the asset and the price will fall, and vice versa if the current price is below the traders' value.

In this study we try to determine a model to describe how a trader may value an asset. First, we assume that the trader has some initial value in mind for the asset. That value changes over time as the trader acquires more information. We assume that the trader acquires information by reading the news, and hence we will want to quantify how the traders' opinion changes with the addition of new news articles. The model we use in this study is a state-space model, where the observable values are stock prices, and the hidden state is the value assigned by the traders. The state equation includes exogenous inputs which represent shocks to the system as traders read the news. We model these exogenous inputs through an aggregate sentiment score, which we define as the feelings conveyed by finance authors for a particular asset. The underlying process, which represents the value to the traders, can then be extracted using a Kalman Filter.

After fitting a regression model to determine the evolution matrix of the state equation, we find that the sum of squared errors is unacceptably high, and therefore conclude that modeling stock prices in terms of a trader's opinion, and a trader's opinion in terms of sentiment scores, is inappropriate.

2 Background

Financial movements in the markets have historically been an overwhelming and unpredictable force. With millions of people on the trading floor, the markets move, respond to actions, and adapt quickly. Recently however, traders have been taking advantage of high powered technology and advanced statistical algorithms to assist them in their day to day trading; information is almost instantly available. This field of algorithmic trading has exploded in the past 10 years, with many statisticians trying to predict how certain parts of the market will respond to changes in another. While other researchers have developed dozens of new mathematical models to attempt to minimize risk and maximize return, we will focus on the human factor.

In this study, we view the financial markets as a large game with many players. The markets are like an auction house, and players sell and bid on certain assets. The value of an asset is largely determined on its current finances and its potential to grow; unfortunately, this kind of information is not available on demand. The more information the players have, the better they can judge the asset and determine its current value and the potential for that value to increase. The opinion of a trader can change when he learns that a company has released a new line of products, or when a company changes its business strategy that increases its profit margins. The news is a crucial source of such information, and the addition of new information is a primary factor in driving movements in the market. Therefore we are interested in studying to what degree can news articles affect the markets.

3 Project Goals

A trader's opinion has a direct impact on how the markets trade and how different assets are valued on a given day. Traders read news articles every day in order to keep up with the global economy, and their opinion is influenced by the sentiment of the articles' authors. Using financial news and business journals, we would like to measure the sentiment an author has for a particular asset or situation, and then measure its impact on a trader's opinion. By doing so, we hope to make predictions on the magnitude and direction of the market whenever new news is released.

Here, we will propose a two stage prediction process:

1) Just as the Google Bot crawls the web to index websites, we will produce a spider that will crawl business articles and financials websites to generate a document matrix that can be analyzed in a learning algorithm. From the output, we will create an aggregate sentiment score for the day, which represents how positive or how negative people feel about a particular asset.

2) The aggregate sentiment score will be used in a regression model to predict the value of an asset

4 Models

Below, we state a series of hypotheses that we would like to test in this study:

- A stock's price is a function of a trader's opinion
- A trader's opinion is influenced by the news articles he reads
- Specifically, a trader's opinion is an autoregressive time series with the news causing exogenous shocks to the system

Let y_t be the percent change in stock price on day t, and let x_t be a trader's opinion also on day t. Define a variable w_t to be the aggregate sentiment toward a stock on day t. We may consider the following model:

$$\begin{aligned} x_t &= \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_k x_{t-k} + \phi_{k+1} w_{t-1} + u_t \quad (1) \\ y_t &= f(x_t) + v_t \quad (2) \end{aligned}$$

where u_t and v_t are random noise. (1) describes the opinion of a trader on day t as some linear combination of his opinion from 1 day ago, 2 days ago, up to k days ago, and what he sees on the news on day t. (2) describes the stock price as some linear function f with input parameter x_t, the opinion of the trader that day.

We will make the intuitive claim that the x_t time series is a hidden process, i.e. we cannot explicitly measure a trader's opinion on a given day. This leads us to believe that a state-space model, where x_t is the unobservable state equation, would be an appropriate model. Rewriting, we claim

$$\begin{aligned} x_t &= \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_k x_{t-k} + \phi_{k+1} w_{t-1} + u_t \\ &= \begin{bmatrix} \phi_1 & \phi_2 & \ldots & \phi_k \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-k} \end{bmatrix} + \phi_{k+1} w_{t-1} + u_t \\ y_t &= A_t x_t + v_t \end{aligned}$$

Thus, we can write the state-space model as

$$\begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-k+1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \ldots & \phi_{k-1} & \phi_k \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-k} \end{bmatrix} + \begin{bmatrix} \phi_{k+1} \end{bmatrix} \begin{bmatrix} w_{t-1} \end{bmatrix} + \begin{bmatrix} u_t \\ \vdots \\ 0 \end{bmatrix} \quad (3)$$

$$y_t = \begin{bmatrix} A_t & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-k+1} \end{bmatrix} + v_t \tag{4}$$

The significance of this is in the 1 step ahead forecast.

$$E[y_{t+1}|y_t] = E[A_{t+1}x_{t+1}|A_t x_t] \tag{5}$$
$$= A_{t+1}[\phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_k x_{t-k} + \phi_{k+1} w_{t-1}] \tag{6}$$

Note that x_{t+1} is a function of lagged values from the past. The prediction would be easy if these values of x_t were observable. The best way to make the 1 step ahead forecast will be to use the Kalman Filter to smooth the stock prices and uncover the underlying process x_t, then plug the smoothed values into (6).

5 Learning Algorithm and Methodology

Note, in this study we will train two learning algorithms: (1) to run sentiment analysis on news articles and determine an aggregate score for a certain asset at time t, and (2) the state space model which will fit the data to a time series that can be forecasted for time $t + 1$.

5.1 Sentiment Analysis

For (1), we will construct a spider that can crawl links on financial websites and download and parse the content of news articles. Every day for two months we read and label the articles ourselves. To build up our vocabulary, we will create a frequency table from a sample of 160 news articles; we must be careful to sample from news articles that have positive connotations as well as negative connotations. It will also be helpful to sample articles that are covering different market sectors. In this study, we use a vocabulary of roughly 9500 words.

When crawling the web, we will generate a feature vector $x^{(i)}$ for each article, where $x_j^{(i)}$ is the amount of times the $j-th$ word in the vocabulary appeared in the $i-th$ article. Later, we expand our vocabulary to contain a bag of words instead of single tokens. Building these feature vectors for multiple articles gives us a document matrix which we can use for training.

For this project, we are concerned with whether or not authors are writing positively, neutrally, or negatively about an asset or situation. Like a spam classification problem, we may choose to use a linear SVM where the labels are either 1, 0, or -1 for positive writing, neutral writing, or negative writing. Using the vocabulary above, we notice that positive articles tend to use words such as *earnings, growth, buy, surge, upgrade, acquire, launch*, and *call*. Negative articles tend to use words such as *charges, infringement, reject, struggle, downgrade*, and *plunge*. On a naive algorithm with cross-validation, our learning algorithm produces the following confusion matrix:

```
Accuracy = 45.8647% (61/133)

C =

     6    12    20
     4    10     6
    20    11    44
```

Based on the above confusion matrix, we note that most of the error comes from predicting positive articles as negative ones, and predicting negative articles as positive ones. This is a common natural language processing problem, which we will discuss below.

5.1.1 Natural Language Processing

Sentinment analysis is very difficult to do because it is based on a fixed vocabulary, and the algorithm cannot detect the surrounding syntax. It is particularly hard to understand the presence of negations, such as:

. . . had a great performance . . . vs. *. . . did not have a great performance . . .*

While we see the word "great" as having positive connotation, our algorithm needs to be aware of the word "not" which reverses its meaning. Unfortunately, these negatives do not necessarily occur right before the word that they modify.

In order to combat this difficulty, we will double the length of our feature vector. Suppose the initial size of the feature vector was n. We will allocate a vector of size $2n$, where the first n values represent our vocabulary, and the latter n values represent their negations. Upon downloading content, the algorithm will cache the location of all negation words in the article. As the spider looks for word w from our vocabulary, it will check if any of the cached negation words are also near w. If w is near a negation word, the algorithm increments the value at $x^{(i)}_{j+n}$ instead of $x^{(i)}_j$. In this study we consider the following negations: no, not, unable, rarely, never, without, barely, hardly.

Other natural language processing problems are apparent in this study. For example, suppose an article compares two companies, Google and IBM. The author may write positively about Google and negatively about IBM, in which case the algorithm will not be able to classify the article as a whole. In general, it is difficult to match an adjective to the noun that it is describing, and likewise with adverbs and verbs. In this case, we keep a lexicon of company names. When performing sentiment analysis on company i, we will look for a word w from our vocabulary and also use an extra byte to store the distance from w to the nearest occurence of the company name i. If this distance is large, we hope that this will nullify the presence of word w.

Applying such techniques yields an improved confusion matrix:

```
Accuracy = 55.6391% (74/133)
```

```
C =

    16     11     16
     5     12      8
     9     10     46
```

The above confusion matrix is a slight improvement, but our learning algorithm continues to report false negatives and false positives. This is likely due to negation detection, and the inability to distinguish which words modify which nouns.

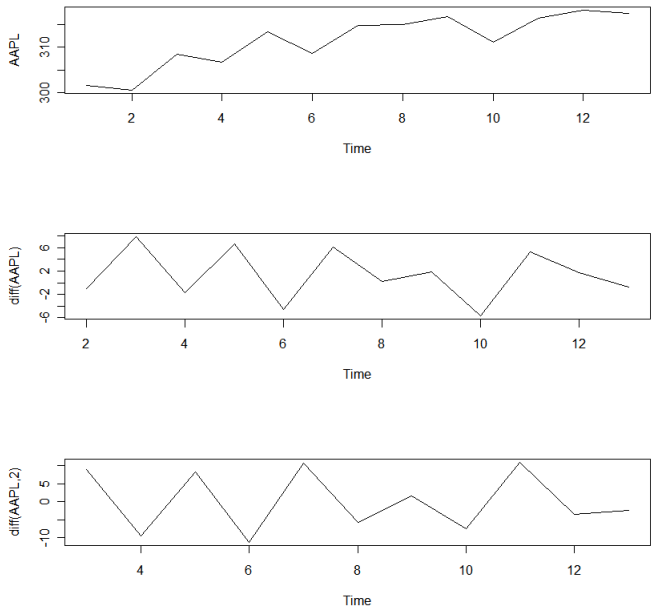
5.2 State Space Model

Here we will use the aggregate sentiment score found from the previous part to construct a time series model. Note that our state space model describes y_t as some linear function of x_t, and x_t is some autoregressive process with exogenous inputs.

First, we can simplify our model by determining the order of the autoregressive component. If x_t is AR(p), then the $p-th$ difference of the x_t time series should be stationary. For brevity, we will show the analysis on one company, Apple.

Below, we plot the closing prices of Apple and find that taking the first difference makes the series stationary. Thus we estimate

$$x_t = \phi_1 x_{t-1} + \phi_2 w_{t-1}$$



The state equation becomes:

$$\begin{bmatrix} x_t \\ w_t \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 0 & \frac{w_t}{w_{t-1}} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ w_{t-1} \end{bmatrix} \tag{7}$$

We would like to learn the parameters A, ϕ_1, ϕ_2 to fit our state space model. Let y_t be the value of a stock at time t, and let $\hat{y}_t$ be the value forecasted by our model. If our goal is to have accurate 1 step ahead forecasts, then we may consider fitting A, ϕ_1, ϕ_2 such that

$$y_{t+1} \approx A\phi_1 x_t + A\phi_2 w_t$$

Without loss of generality, we claim $A \approx 1$, and make any necessary adjustments to ϕ_1 and ϕ_2. The above is akin to a least squares regression problem. However, we should note that changing ϕ_1 and ϕ_2 changes the evolution matrix, which changes the smoothed values x_t. Instead of writing x_t as a recursive process, we will write it in closed form with respect to its initial value x_1. Recall that w_t is an exogenous input that is already known.

Lemma: If x_t is AR(1) with exogenous inputs and initial value x_1, and has the evolution equation $x_t = \phi_1 x_{t-1} + \phi_2 w_{t-1}$ then we can write $x_t = \phi_1^{t-1} x_1 + \phi_1^{t-2}\phi_2 w_1 + \ldots + \phi_1^1 \phi_2 w_{t-2} + \phi_2 w_{t-1}$.

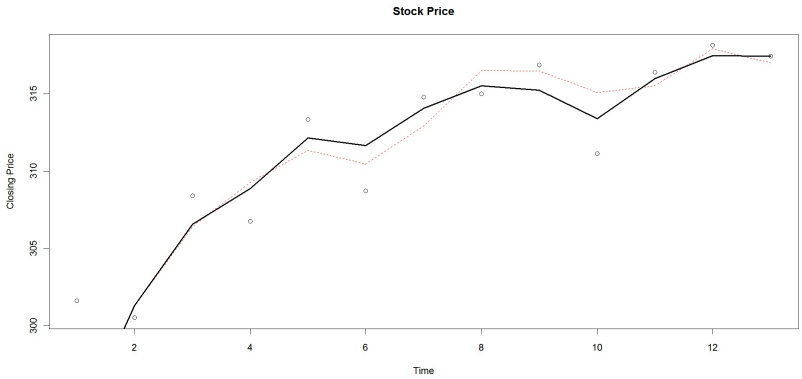
Proof: We illustrate the recursive process here:

$$\begin{aligned}
x_2 &= \phi_1 x_1 + \phi_2 w_1 && (8)\\
x_3 &= \phi_1 x_2 + \phi_2 w_2 && (9)\\
&= \phi_1(\phi_1 x_1 + \phi_2 w_1) + \phi_2 w_2 && (10)\\
&= \phi_1^2 x_1 + \phi_1\phi_2 w_1 + \phi_2 w_2 && (11)\\
x_4 &= \phi_1^3 x_1 + \phi_1^2\phi_2 w_1 + \phi_1\phi_2 w_2 + \phi_2 w_3 && (12)\\
&\vdots \\
x_t &= \phi_1^{t-1} x_1 + \phi_1^{t-2}\phi_2 w_1 + \ldots + \phi_1^1\phi_2 w_{t-2} + \phi_2 w_{t-1} && (13)
\end{aligned}$$

Then, we will fit ϕ_1, ϕ_2 that satisfy

$$\arg\min_{\phi_1,\phi_2} (y_{t+1} - \phi_1 x_t - \phi_2 w_t)^2$$

Running this on the AAPL data series, we find that $\phi_1 \approx 1.0005$, and $\phi_2 \approx 0.9$. Below, we show the Kalman Filter applied to the AAPL series with these parameters; the underlying process x_t is drawn through its points.



Unfortunately, these parameters yield a sum of squared errors value of 418 for the rpediction, and hence $R^2 = 0.67$. We repeat this on a portfolio of 23 stocks and find that the R^2 value hovers around 0.73. Hence, we must deduce that this model is inappropriate for the data.

6 Conclusion

In this study we tried to model a stock price as a function of traders' opinions, and traders' opinions as an autoregressive time series that had exogenous inputs from the opinions of authors writing financial articles. We implemented a two stage learning process where we tried to analyze the sentiment of an author using a linear SVM, and then used an aggregate sentiment score to fit a state-space model. The end goal was to uncover an underlying process that drove stock price movements by applying a Kalman Filter on the observed stock prices. During this study, we encountered many natural language processing problems that prevented our spider from accurately labeling a piece of financial text. Furthermore, through regression diagnostics, we have determined that the claimed state-space model is not suitable for prediction.

6.1 Further Work

- Improve the accuracy of the spider, particularly in negation detection and the ability to recognize authors who are comparing two or more assets.

- Collect more data - this study monitored stock prices during a very volatile period from Sept '10 - Nov '10.

- Expand the size of the state-space model to include more predictor variables. May be interesting to switch to GAM (Generalized Additive Modeling).

- Study interaction terms - how does sentiment on stock i affect a trader's opinion on stock j, and hence how does news about stock i affect the stock price on j? This is particularly interesting when news is released on competing companies.