Modeling of Complex Systems

John Sherwood

May 15, 2007

Contents

1 Abstract

The stock market is an immensely complex system made up of millions of interactions between different investors and affected by every action made by thousands of companies. However, economic theory decrees that the actions of most investors are governed by the actions of a few well informed primary investors and that those other investors primarily follow preexisting trends

set by the well informed. Thus, as the primary catalyst for the well informed should be news reports, press releases, income reports, etc, (barring things like insider trading), it should be possible to predict the broad trends across the market and fairly detailed trends for specific stocks by analyzing the available news on stocks.

2 Introduction

The internet is a vast repository of data, and given such an immense amount of data, it is assured that at least some amount of that data is financial data, such as stock price history, and business news reports. While most attempts to predict stock prices are based on analyzing pure numerical data in attempts to regress relations between price history and future prices, the eventuality of internet news reporting makes it theoretically possible to evaluate the effect of specific pieces of data on stock prices in both the short and long term. Given that a relationship does exist between the price of a stock and related news, keeping track of news on companies as well as stock price histories should allow for predictions of immediate and semi-long term price changes in company stocks.

3 Previous Work

Due to the highly lucrative rewards of success for determining a method with which one may predict future stock prices, there have accordingly been many other attempts with similar projects. The majority have been based on purely numerical regression, attempting to predict future prices entirely from the previous price record. Based on my research, at the time I began this project, no prediction methods attempted to base their prediction of the stock market off of the prediction of human reactions to stock-related news.

4 Methodology

The concept of the program is that humans react in predictable ways to certain events, and that these events - if related to a company - will accordingly effect the stock price. My project stored records on several major stocks, as stocks that are commonly traded should accordingly have more detailed

news and have less overall speculative price fluctuation, and then generated equations correlating news items to price changes in stocks.

4.1 Process Specifics

4.1.1 Data Mining

Through the course of my project, several methods for gathering data were used. The news data was originally was culled from RSS (Really Simple Syndication) feeds from Yahoo! Finance (finance.yahoo.com) using an XML parser I wrote. However, these did not provide the degree of detail required for the rating algorithm (discussed in section 4.2.3, Quantitative Processing). I then turned to Google Finance (finance.google.com/finance), which compiles abstracts of and links to recent news for each company whose stock one researches. My data mining script reads the Google Finance pages for the links, then follows them to whatever website they link to. Due to the variety of layouts and designs of the web pages that were linked to from Google, it was impossible to write a parser that specifically went for the content parts of layouts. Instead, the algorithm takes all plain text portions of the site with continuous content longer than 200 characters. While this does occasionally grab other items such as copyright notices, credentials, etc, these miscellaneous sections do not effect the quantitative score of the news (discussed in in section 4.2.3). Despite the complexities of the news mining algorithms, the design of the price recording algorithm was much simpler. Using CSV files supplied by Yahoo! Finance, an automated algorithm is capable of storing the slightly delayed price of a stock at an arbitrary interval, in this case daily.

4.1.2 Data Storage

Due to the comparatively massive amounts of data being stored (several megabytes of plain text data over a period of months), an efficient data storage and retrieval method was needed, and so I chose to use a MySQL database to store the data. Two tables were used, one for the news data and one for the price records. The table structure is as follows:

mining_2

Field	Type	Null	Key	Default	Extra
newsID	int(11)	NO	PRI	NULL	auto_increment
subject	text	NO			
news	text	NO			
symbol	varchar(10)	NO			
time	datetime	NO			
score	double	NO			
normalized_score	double	NO			

stockData

Field	Type	Null	Key	Default	Extra
symbol	varchar(10)	NO			
price	double	NO			
time	int(11)	NO			

4.1.3 Quantitative Processing

Once data has been collected from the various sources supplied by Google Finance, the data must be changed from a textual format intended to be read by humans into numerical data that can be interpreted by the regression algorithms (discussed in section 4.2.5, Regression). To achieve this, the scoring algorithm looks at each news item and checks for the presence of key terms that it corresponds to a certain positive or negative constant. Thus, randomly included text does not affect the score of a news item as it is highly unlikely that the included text would contain any of the key terms. In order to determine what key terms and their corresponding values, I personally scored a collection of data. Then, I wrote a script to try and determine key terms and values that would match the news items to the score I gave them. Finally, I would have the program score other news items in order to test its accuracy. When I found it worked satisfactorily, the key terms and corresponding scores were as follows:

- deal, .2
- lawsuit, -.15
- infringement, -.3
- new, .2
- error, -.2

- loss, -.3
- killed,-.7
- hiring, .4
- layoff, -.3
- earnings, .2
- increase, .1
- increased profit, .5
- fired, -.1
- scandal, -.4
- merger, .4
- delay, -.2
- sue, -.2
- infringe, -.25
- patent, .1
- correupt, -.5
- favorable, .1
- enhancement, .2
- win, .1

A function in the program was created to score each piece of news, storing the score into the score column of the news table. The scores were then normalized between -1 and 1 based on the lowest and highest scored news, with that entry put into the normalized_score column.

4.1.4 Regression

Once the news items were scored, equations needed to be created to ascertain the affect of news on price over time. I decided to model the effect of a piece of news with score S at time ΔT past the time at which it first occurred with the equation $\Delta price = S * c^{k*\Delta T}$. As both the values of S and ΔT are known, an algorithm is still required to determine the values of c and k. In order to regress the values of c and k, I wrote and used a genetic algorithm that took a start and end time to generate an equation for, then found the target $\Delta price$ and news items from the MySQL tables, and after a set number of generations returned the best equation it had found. The accuracy of an equation was determined by summing the output of the equation for each news item in the interval, and comparing to the expected $\Delta price$.

4.1.5 Program Architecture

Despite that the project itself was primarily a compilation of various scripts servering different purposes, the majority of the code for the project was compiled into several Python modules that were then imported into the scripts. Four primary modules - XML.parser, techlab, regression, and config were written. XML.parser contained - naturally - the XML parsing algorithm as well as a plethora of useful functions for the mining algorithms. The 'techlab' module itself contains most of the functions used to score news, set up data regression, and so on. The regression module contains the regression methods, and finally the config module supplies the common connection used by all scripts to connect to the MySQL server.

5 Results

Using my database and equation regression algorithm, I tested the validity of the resultant equations by applying them to other time intervals to see how accurate they were. While all equations were very precise for times immediately adjacent to the interval they had been generated with data from, there was a steep decline in the accuracy of the equations as the interval they were applied to grew more distant from the interval they originated from.

6 Analysis

As there was a definate correlation to the proximity of the timeframe that equations were used to compared to the timeframe that the equations were generated in and there accuracy, there are several possible conclusions. The first, and most likely, is that my equations were not complex enough to take into account unrealized factors, and thus were only primitive enough to determine prices with more data than that which they were originally supplied with. The second is that the way that the market reacts to news is itself timeframe specific, and that the time at which a piece of news is released changes its overall effect on the price of a stock. The last possibility is that while my reasoning and methodology is sound, my execution was not sophisticated enough to find and analyze all the news items that were effecting stock prices. While the existence of multiple feasible reasons for the less than complete success of my project makes any definite conclusions difficult to draw, it still appears that my work has provided a suitable framework for future, more sophisticated attempts.