

# Research into the Modeling of Complex Systems

## Predicting the Stock Market

John Sherwood  
Period 2

### 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.

### Methodology

In order to gain the information necessary to analyze stocks, an XML parser is used to read financial news based websites (Such as Google Finance and Reuters). The information garnered by these data miners is stored into a MySQL database so that it is accessible to other aspects of the program. The news data is then analyzed with a program that takes the qualitative news rendered in text and assigns a signed floating point number to it so that the regressor has a qualitative value to base its analysis on . This, along with the stock price data which is taken from CSV (comma-seperated-value) files provided by Yahoo! Finance, is fed into an equation refiner, which generates equations describing the effect of news over time on stock prices using a genetic algorithm to refine the equations. The equations are then used to project the future price of stocks.

```
def regress(price_data, news_data, generations=100, children=4):
    start_price = price_data[min(price_data.keys())]
    end_price = price_data[max(price_data.keys())]
    end_time = max(price_data.keys())
    equations = [equation() for x in range(children)]
    target = end_price/start_price
    mutations = .25
    for generation in range(generations):
        output = [0 for x in range(children)]
        for i in range(children):
            for time in news_data.keys():
                if time > end_time: break
                output[i] += equations[i].calc(news_data[time], end_time-
time)
            output[i] = (abs(1-output[i]/target), equations[i])
        output.sort()
        keep = children//2
        generate = children - keep
        equations = [output[k][1] for k in range(keep)]
        mutamt = output[keep-1][0]/keep
        for i in range(generate):
            b = random.choice(equations).decay_rate
            c = random.choice(equations).mult_const
            if random.random() < mutations:
                if random.random() < .5: b = b * (1+mutamt)
                else: b = b * (1-mutamt)
            else:
                if random.random() < .5: c = c * (1+mutamt)
                else: c = c * (1-mutamt)
            equations.append(equation(b,c))
    return output[0]
```

### Results and Findings

So far, with previous tests involving AI bots that managed portfolios, the stock market can be predicted – to a degree – using very basic algorithms and low quality news, but with higher sophistication, results should be greatly improved. The overall success of the predictor hinges primarily on quality of data and data analysis, which are very error prone, but still feasibly capable of being parsed with some good degree of accuracy.

**A part of the equation regression system**