Analysis of Hidden Markov Models and Support Vector Machines in Financial Applications

Satish Rao
Jerry Hong

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-63
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-63.html

May 12, 2010

Acknowledgement

We thank Simlio LLC and all its team members for providing many useful market tools to allow us to graph and display market trends with ease. We also thank Professor Peter Bartlett for providing many useful lectures about statistical graphical models, such as exponential families and maximum likelihood, and for helping us get started on this topic. Furthermore, Professor Satish Rao played a crucial role in helping us determine what feature sets to try as well as suggesting different trading algorithms. Satish's ideas were instrumental in developing the manipulation detection section of this paper.

Analysis of Hidden Markov Models and Support Vector Machines in Financial Applications

Jerry Hong
University of California, Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720-1776
jerricality@gmail.com

ABSTRACT

This paper presents two approaches in helping investors make better decisions. First, we discuss conventional methods, such as using the Efficient Market Hypothesis and technical indicators, for forecasting stock prices and movements. We will show that these methods are inadequate, and thus, we need to rethink the issue. Afterwards, we will discuss using artificial intelligence, such as Hidden Markov Models and Support Vector Machines, to help investors gather and compute enormous amount of data that will enable them to make informed decisions. We will leverage the Simlio[*] engine to train both the HMM and SVM on past datasets and use it to predict future stock movements. The results are encouraging and they warrant future research on using AI for market forecasts.

[*]Simlio LLC is a startup co-founded by Jerry Hong. It is currently a stock research platform on the web that enables users to draw graphs at ease as well as perform intensive formula calculations to see how well an idea would profit over time.

1. INTRODUCTION

In much of traditional finance theories and modeling, we are under the assumption that there exists symmetric information among the agents. We generalize the market as a perfectly competitive world where individuals are price-takers and present an over-simplistic representation of the financial market. The conventional theories that we focus in this paper include EMH (Efficient Market Hypothesis), and technical indicators, such as the SMA (Simple Moving Average) or MACD (Moving Average Convergence / Divergence) [1]. These three tools are used to help model the world of finance and assist investors in predicting future events in the market. For instance, technical indicators are often used by stock traders for predicting future prices using historical trends.

These conventional tools offered much insight into the workings of the financial market. However, they provide only a macro-simplification that does not always reflect how the real market works. There are definitely limitations to these tools that prevent them from modeling the market in a more focused, "micro" manner. One of the major issues is that many conventional finance theories only take in so many factors. This limited scope prevents us to accurately model the real market that has countably infinite number of patterns. We need a model that can constantly adapt to the dynamic nature of the market. Technical indicators can only help an investor so much before the different combinations and patterns causes the investor to question whether any formula actually works consistently.

This is where AI models such as HMMs and SVMs come into play. Using these tools, we can achieve a more realistic micro-representation of the market while overcoming the limitations of the earlier

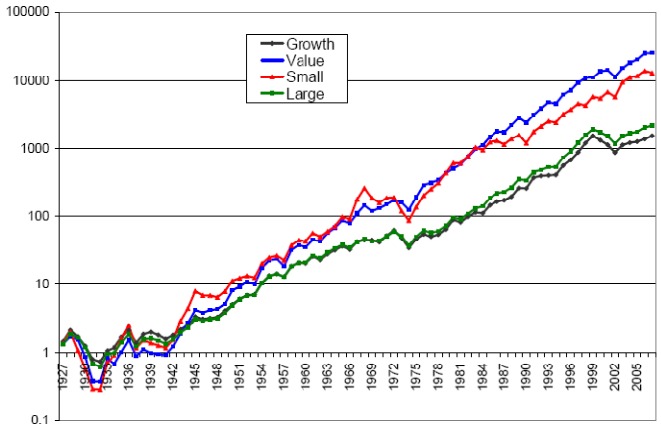
techniques. They allow us to model many different historical patterns and predict how they change through time. In this paper, we will first review conventional models and explore their benefits and problems. Focusing on their shortcomings, we will see how HMMs and SVMs can potentially overcome these weaknesses while maintaining the advantages of classical finance theory[2].

1.1 Introduction to Conventional Approaches

1.1.1 Efficient Market Hypothesis

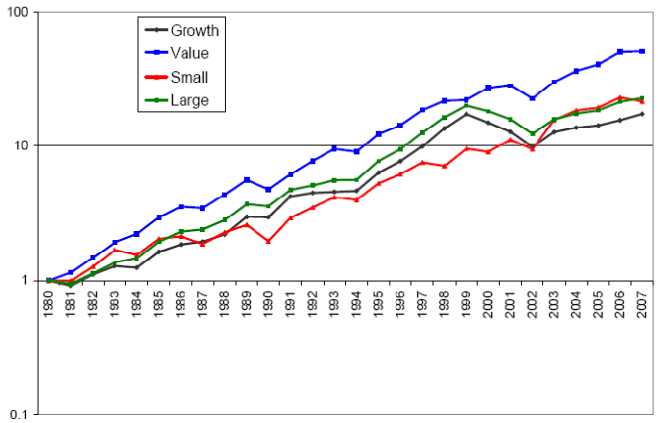
The EMH comes in three different forms: weak, semi-strong, strong. All of them claim in some way that the financial markets are "information efficient" and thus using past/historical data will not help predict future prices because the current prices already take that information into account. The semi-strong and strong forms take a further step and claim that it is futile even if one knows public information and private information respectively.

This theory has been rather controversial for the last few decades because there have been some examples in history of where it holds and others where it seems to suggest that the EMH can't certainly be true. One example of that supports the EMH is the public announcement of the value and small stocks vs. growth and large stocks (See Figures). Before the announcement, this idea has been used by many investment firms to help them profit above the norm for some number of years. Essentially, it is better to buy small/value stocks than large/growth because they have more potential and they are undervalued. So before the 1980s, if you bought these types of stocks, you were more likely to receive a higher profit. However, after 1980, a PhD student discovered the trend that small stocks are being undervalued and many public investors don't pay much attention to it. He decided to publish his findings and as a result, this turned into public information. As one can see from the diagrams, it is no longer feasible to just buy small stocks to profit greatly, for the EMH has taken that factor into account and that information is no longer usable to gain an edge in the market.



http://www.econ.berkeley.edu/~szeidl/ec136/lecture16.pdf

Figure 1: Value and Size Effect from 1927-2005



http://www.econ.berkeley.edu/~szeidl/ec136/lecture16.pdf

Figure 2: Value and Size Effect after 1980

Fortunately for investors, there are many shortcomings to this theory. For one, the empirical evidence for whether this theory holds or not has been mixed. For instance, many papers are published showing that low P/E ratio stocks seem to provide greater returns[3]. Another example is that the "loser" stocks today are usually much more undervalued than the "winner" stocks, so they get less attention. Historically, the "loser" stocks yield higher average returns than the "winner" stocks at the time; hence, this becomes an endless cycle. If the EMH were to hold, then these instances should have happened and there will be no point of trying to beat the market[4].

However, it is very difficult to determine whether or not the EMH holds. Thus, we need new methods

to help decide whether it is able to predict the market. We need a tool that is very versatile that can take in multiple factors into account, such as combinations of historical data, news, people's behaviors, etc. One potential way of doing so could be to use HMMs and SVMs.

1.1.2 Technical Indicators

Another conventional method in financial economics is to use technical indicators to predict stock movements and prices. There are numerous indicators that investors use and some of the most commonly used ones are the EMA (exponential moving average), MACD (Moving Average Convergence / Divergence), Bollinger Band, etc. We will point out one of most common indicators that many investors use to help them pinpoint relative resistance points.

Looking at Figure 3, we used a stock investment tool, called Simlio, to draw out the EMA for 50 and 200 days on the stock Apple from September 2008 to February 2009. Investors pay very close attention to the indicators pointed out by the red circles on the figure. When the EMA 50 crosses below the EMA 200, it usually signals a downtrend in the future. As we see above, the Apple stock fell from $140, at the time the signal got triggered, to around $80. Thus, the investors who heeded the signal and got out of the market or shorted the stocks were much better off than those who stayed in the market.

Another signal that investors pay close attention to is the EMA 50 when the prices are below it. This line usually symbolizes a resistance point and that unless a stock has regained its momentum and health again, the prices will not go above this line. As we see from December 2008 to February 2009, Apple's stock prices hit the resistance line again and again, but it wasn't able to break the barrier caused by the EMA 50.

Courtesy of Simlio (www.simlio.com)

Figure 3: EMA 50(Green) and EMA 200(Blue) for Apple

Courtesy of Simlio (www.simlio.com)

Figure 4: MACD (26, 12, 9) for Apple

Another indicator that many investors use is the MACD [5]. As a matter of fact, the MACD used to work very well when hedge funds kept this concept private. However, when the idea of this indicator was published, it started working less efficiently, probably due to the EMH. Nevertheless, it is still very useful in predicting up and down trends. Looking at Apple's stock prices again, we label four of the many instances of when the MACD lines crossed each other. In the diagram above, when the orange line goes below the green line, it signals a downtrend in the near future. When the orange line crosses and goes above the green line, it signals an uptrend in the near future. Although this indicator is not perfect, the four cases we showed above do a very good job in predicting the future at the given times. Again, investors who used this indicator may be at a better position to trade than those who do not do research.

Although technical indicators appear to be very helpful, there are many shortcomings. First, most indicators only analyze historical prices and do not take any other factor into account. Thus, it is limited to what kind of information it can take in as an input. This becomes problematic because it does not help explain why some stocks are in a downtrend/uptrend. For instance, when the mortgage crisis hit the economy around a year ago, there are certain companies that investors should have been wary about. For instance, financial companies that made huge margin bets on mortgage deals (such as Lehman Brothers) or even commercial banks that generously gave very low interests on loans (such as Washington Mutual) were definitely ones that investors should have thought twice about before purchasing any of their stocks. Investors that did their news research and pieced the information together either shorted these stocks or avoided them. Those who purchased stocks suffered greatly as both of their stocks dropped to essentially 0. Solely using technical indicators would not have told any investors to back away from these stocks, but utilizing other factors and information in the economy should have been more than sufficient to cause the investors to be suspicious.

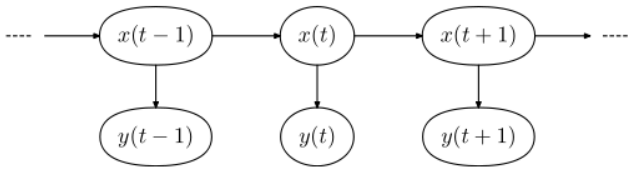
This is where statistical learning theories, such as HMMs and SVMs, can make a significant difference. They allow us to model the market based on many different information. For instance,

it can model how historical prices, fundamentals, and current news affect stock prices. They eliminate the weaknesses of being limited to just analyzing historical prices and they can be exploited to take other factors in the economy into account. The section below will discuss the true potential of applying HMMs and SVMs in forecasting the market.

1.2 Introduction to Modern Approaches

1.2.1 Hidden Markov Models

The Hidden Markov Model (HMM) is a statistical model that is often used in pattern recognition applications, such as speech, handwriting, bioinformatics, etc[6]. The user first needs to decide on how many hidden states are possible for each unobserved state. Moreover, the initial starting probability of each of the hidden states must be specified. Afterwards, the HMM model needs to be trained on a set of data where we have a set of possible observation emissions for each unobserved state.



Courtesy of Wikipedia

Figure 5: States of HMM

We followed similar notations from Hassan and Nath's paper[7].

N = # of states in the model
M = # of distinct observation symbols per state
T = length of observation sequence
O = observation sequence
Q = state sequence $q_1, q_2, \ldots, q_T$ in the Markov model
A = $\{a_{ij}\}$ (transition matrix) where a_{ij} represent the transition probability from state i to state j
B = $\{b_j(O_t)\}$ (observation emission matrix), where $b_j(O_t)$ represent the probability of observing O_t at state j
$\pi = \{\pi_i\}$ the prior probability
$\lambda = (A,B, \pi)$ (the overall HMM model)

Using HMMs, we can somewhat accurately answer the following three questions[7]:

1. Given the model λ, what is $P(O \mid \lambda)$ where $O=O_1, O_2, \ldots, O_T$?
2. Given the observation sequence O and a model λ, what is the best/most likely state sequence $q_1, q_2, \ldots, q_T$?
3. Given the observation sequence O and a space of models found by varying the model parameters, what is the best model?

We will use the forward-backward algorithm to solve $P(O \mid \lambda)$ and use Viterbi algorithm to answer #2. As for #3, we will look into Baum-Welch algorithm to train the HMM for the best parameters and test it on a dataset.

The Baum-Welch algorithm is a special case of EM (Expectation-Maximization) algorithm[9], allowing us to find the best parameters for the model λ. The EM algorithm is an iterative method used to find the maximum likelihood estimates of parameters when there is hidden data[10]. There are two steps in each iteration of the EM algorithm: the E step and the M step. We use conditional expectation to best estimate the missing data using the given observed features and most updated model. In the M step, we maximize the likelihood function assuming that we have the missing data. One great property of the EM algorithm is that it is guaranteed to converge because we increase the likelihood at each iteration.

Here is a quick derivation of the EM algorithm[10].

$$L(\theta) = \ln P(X|\theta)$$

$L(\theta)$ is the log likelihood function of θ and X is a random vector. Our goal is to find θ that maximizes $P(X|\ \theta)$. At each step during the iteration, we want to make an improvement in maximizing $L(\theta)$. Recall that ln(x) is a strictly increasing function. Thus, at each

iteration, we want our new L(θ) to be greater than the old one:

$$L(\Theta) > L(\Theta_{current})$$
$$L(\Theta) - L(\Theta_{current}) = \ln P(X \mid \Theta) - \ln P(X \mid \Theta_{current})$$

Now to make things interesting, we will introduce a hidden random vector Z, whose given realization will be noted as z. P(X| θ) is now:

$$P(X \mid \Theta) = \sum_z P(X \mid z, \Theta)\, P(z \mid \Theta)$$

$$L(\Theta) - L(\Theta_{current})$$
$$\ln\left(\sum_z P(X \mid z, \Theta)\, P(z \mid \Theta)\right) - \ln P(X \mid \Theta_{current})$$
$$\ln\left(\sum_z P(X \mid z, \Theta)\, P(z \mid \Theta) * \frac{P(z \mid X, \Theta_{current})}{P(z \mid X, \Theta_{current})}\right) - \ln P(X \mid \Theta_{current})$$
$$\ln\left(\sum_z P(z \mid X, \Theta_{current})\, \frac{P(X \mid z, \Theta)\, P(z \mid \Theta)}{P(z \mid X, \Theta_{current})}\right) - \ln P(X \mid \Theta_{current})$$
$$\geq \sum_z P(z \mid X, \Theta_{current})\, \frac{P(X \mid z, \Theta)\, P(z \mid \Theta)}{P(z \mid X, \Theta_{current})} - \ln P(X \mid \Theta_{current})$$
$$= \sum_z P(z \mid X, \Theta_{current})\, \frac{P(X \mid z, \Theta)\, P(z \mid \Theta)}{P(z \mid X, \Theta_{current})} = \Delta(\Theta \mid \Theta_{current})$$

Thus, we now have:

$$L(\Theta) \geq L(\Theta_{current}) + \Delta(\Theta \mid \Theta_{current})$$
$$l(\Theta \mid \Theta_{current}) \triangleq L(\Theta_{current}) + \Delta(\Theta \mid \Theta_{current}),$$
$$\text{where } L(\Theta) \geq l(\Theta \mid \Theta_{current})$$

Our objective is to figure out what values of θ would maximize L(θ). In order to do that, we could try to maximize $l(\theta \mid \theta_{current})$ instead. So,

$$\Theta_{current+1} = \arg\max_\Theta \{ l(\Theta \mid \Theta_{current}) \}$$
$$\arg\max_\Theta \left\{ L(\Theta_{current}) + \sum_z P(z \mid X, \Theta_{current})\, \frac{P(X \mid z, \Theta)\, P(z \mid \Theta)}{P(z \mid X, \Theta_{current})} \right\}$$
$$\arg\max_\Theta \left\{ \sum_z P(z \mid X, \Theta_{current})\, P(X \mid z, \Theta)\, P(z \mid \Theta) \right\}$$
$$\arg\max_\Theta \left\{ \sum_z P(z \mid X, \Theta_{current})\, \ln \frac{P(X, z, \Theta)\, P(z \mid \Theta)}{P(z, \Theta)\, P(\Theta)} \right\}$$
$$\arg\max_\Theta \{ E_{z \mid X, \Theta_{current}} \{ \ln P(X, z \mid \Theta) \} \}$$

The E step is used to determine:

$$E_{z \mid X, \Theta_{current}} \{ \ln P(X, z \mid \Theta) \}$$

The M step is used to maximize the above expression with respect to θ. We continue the EM iterations until we maximized our log likelihood. Note that a special feature about this algorithm is that convergence is guaranteed since we are using log functions.

The Baum-Welch algorithm is a particular version of the generalized EM algorithm. It uses the forward-backward algorithm with an additional two auxiliary variables[11]. Using the notation noted above in Hassan and Nath's paper, the Baum-Welch re-estimation formulas aim at adjusting the parameters of the model λ = (A,B, π), such that we achieve the maximum value of P(O | λ). Before we use the Baum-Welch algorithm to improve our parameters, an initial HMM must be constructed. We could potentially use the K-means algorithm, or we could find some other way of guessing the initial values[12]. Let's note a couple of new notations:

$$\gamma_t(i) = P(i_t = i \mid O, \lambda)$$
$$\frac{P(i_t = i, O \mid \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\, \beta_t(i)}{P(O \mid \lambda)}$$

$\gamma_t(i)$ is the probability that we are in state i at time t given by the observation sequences O and model λ. Furthermore,

$$\epsilon_t(i, j) = P(i_t = i, i_{t+1} = j \mid O, \lambda)$$
$$\frac{P(i_t = i, i_{t+1} = j, O \mid \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O \mid \lambda)}$$

$\varepsilon_t(i,j)$ is defined as the probability of being in state i at time t and transitioning to state j at time t+1, given the observation sequences O and model λ. The simplified equation can be easily deciphered as:

$\alpha_t(i)$ = observation sequences $O_1,\ldots,O_t$

a_{ij} = transitioning from state i to j

$b_j(O_{t+1})$ = seeing observation O_{t+1} at state j

$\beta_{t+1}(j)$ = remaining observation sequences O_{t+2} to O_T

We also note that:

1. $\sum_{t=1}^{T-1} \gamma_t(i)$ = Expected number of transitions from state i
2. $\sum_{t=1}^{T-1} \varepsilon_t(i,j)$ = Expected number of transitions from state i to state j

Then, the Baum-Welch alpha-beta recursion update equations are as follows:

$$\hat{\pi}_i = \gamma_t(i),\ 1 \le i \le N$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\hat{b}_j(k) = \frac{\sum_{t=1 \text{ and } o_t=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

After finding the best parameters for the model λ, we will use the Viterbi algorithm to find the most likely sequence given some set of observations. This will consequently also give us the current most likely state and we can leverage that to predict the future state, which in our case would either be the future price or price movement. To demonstrate this concept with equations[13]:

$$Q^* = q^*_{1:T} = \arg\max_{q_{1:T}} P(q_{1:T} \mid o_{1:T}) = \arg\max_{q_{1:T}} P(q_{1:T}, o_{1:T})$$

Let $m_t[q_t]$ store the most likely state at time t:

$$m_t[q_t] = \max_{q_{1:t-1}} P(q_{1:t-1}, q_t, o_{1:T})$$

$$\max_{q_{1:t-1}} P(q_{1:t-1}, o_{1:t-1})\, P(q_t \mid q_{t-1})\, P(o_t \mid q_t)$$

$$P(o_t \mid q_t) \max_{q_{1:t-1}} P(q_t \mid q_{t-1}) \max_{q_{1:t-2}} P(q_{1:t-1}, o_{1:t-1})$$

$$P(o_t \mid q_t) \max_{q_{1:t-1}} P(q_t \mid q_{t-1})\, m_{t-1}[q_{t-1}]$$

Now that we have methods to train the HMM model for the best parameters and predict the most likely current state, we will be able to apply this modern approach to model financial time series: stocks. We can vary our inputs by supplying different combinations of technical indicators and news. The HMM will try to train itself using historical prices and data. The hidden states can either be discrete or continuous. For discrete states, we could potentially want HMM to predict how much a particular stock would move tomorrow: big drop, low drop, neutral, low increase, or high increase. If we use a continuous hidden state, we would most likely be predicting tomorrow's closing price. We will be using HMMs to learn and locate patterns from the past and apply it to today's stock price behavior, which essentially answers the question about what tomorrow's predicted price is: $P(q_{t+1}|q_t)$.

However, there are issues with using HMMs to model financial time series data[8]. HMMs have much success in models that are not sensitive to the concept of time, such as language and video processing. However, once the sequence of observed data becomes important, there is an extra factor, the time dimension, the HMMs cannot really account for. Moreover, financial data has an even more unique property in the sense that current data is more important than past data because old patterns and trends may no longer apply anymore. Thus, we will also look into a different machine learning approach and compare the two results.

1.2.2 Support Vector Machines

The support vector machine (SVM) is a data classification technique that has been recently shown to outperform other machine learning techniques when applied to stock market forecasting[8].

Similarly to HMMs, given a set of training examples, SVMs will try to build a model. Each training data instance is marked as belonging to one of two categories. The SVM will attempt to separate the data instances into those two categories with a p-1 dimensional hyperplane, where p is the size of each data instance. This model can then be used on a new data instance to predict which category it would fall onto. The maximum margin hyperplane can be represented as[14]:

$$y(x) = b + \sum \alpha_i y_i K(x(i), x)$$

Vector x is a test example and y_i is the class value of the training example x(i). In the equation, the parameters of the hyperplane are b and α_i. b is a real constant, and α_i are non-negative real constants. The function K(x(i), x) is a kernel function and SVMs are powerful in the sense that one can substitute different kernel functions. The four basic kernel functions are[15]:

1. Linear: $K(x_i,x_j)=x_i^T x_j$
2. Polynomial: $K(x_i,x_j)=(\gamma x_i^T x_j+r)^d$, $\gamma>0$
3. Radial(RBF): $K(x_i,x_j)=\exp(-\gamma||x_i - x_j||^2),\gamma>0$
4. Sigmoid: $K(x_i,x_j)=(\gamma x_i^T x_j+r)$

The classifier can be constructed as follows[17]:

$$w^T \varphi(x(i)) + b \geq 1, \text{ if } y_i = 1$$
$$w^T \varphi(x(i)) + b \leq 1, \text{ if } y_i = -1$$
$$y_i[w^T \varphi(x(i)) + b] \geq 1, \; i = 1, \ldots, N$$

where φ(-) is a nonlinear function that maps the given inputs into some higher dimensional space. In case we cannot find the separating hyperplane in this space, we introduce additional variables: ξ_i, where i=1, ..., N. After this, we will attempt to solve this minimization problem:

$$\min_{w,b,\xi_i} J(w, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$
$$\text{s.t. } y_i[w^T \varphi(x(i)) + b] \geq 1 - \xi_i$$
$$\xi_i \geq 0, \; i = 1, \ldots, N$$

The solution to the above model will be the optimal separating hyperplane.

For this paper, we focused primarily on the radial kernel because K. Kim's paper has shown that by tweaking the RBF kernel slightly, the SVM can give superior results versus using other types of kernels when forecasting stock market prices[14]. The RBF kernel is capable of handling relations between class labels and attributes that are nonlinear by nonlinearly mapping samples into higher dimensional space if required, thus giving us the extra buffer just in case our feature set cannot be separated in a linear fashion.

Moreover, SVMs are very promising because it reduces the danger of overfitting. With other machine learning techniques, such a neural networks and possibly even HMM, we risk training the model too well and it starts to overfit our training data. When that happens, we get significantly poorer results when we start providing testing data to the trained model. Furthermore, since we are using the RBF kernel, we only have to tune 2 parameters – C and γ. That is relatively little work compared to other machine learning methods.

Thus, given the advantages of using SVMs, we will try to train it on a few years worth of data and use it to forecast stock prices/movements. We will also be comparing the results with our HMM experiments too.

2. APPLYING MODERN APPROACHES AND ITS RESULTS

Forecasting financial time series is difficult because there is no single fixed model that explains the changes in prices all the time. In order to account for the dynamic pattern changes, we need to use a tool that can adapt to new situations. The Hidden Markov Model (HMM) with a Gaussian mixture at each state has the potential in tackling such a problem[8]. We will use a Markov chain to represent stock movement. Using this model, we are able to make predictions to answer questions, such as "What is the probability of seeing a big price drop tomorrow given today's state and observations." We will show that the results are somewhat promising; however, due to the limitations of HMMs mentioned above, we looked into SVMs in hopes of it giving us better results.

2.1.1 Hidden Markov Model with K-means

In this experiment, we used the Simlio engine to provide us the data we need to pass into the HMM. We initially used lagged profits as the sole observation in forecasting future price movements. We decided to run 11 experiments (10 stocks and 1 index), and in each experiment, we used the k-means learner algorithm in Jahmm's library to cluster the data points into one of the 5 hidden states: big price movement up, small price movement up, no movement, small price movement down, big price movement down[18].

Courtesy of Simlio (www.simlio.com)

Figure 6: Pot (Sept 09 – Nov 09)

We trained the HMM on daily lag profits from the beginning of 2004 to mid September 2009. We then tested and trained the data on the last 30 days to calculate how accurate our HMM model is in forecasting price movements.
Here are the following results:

ticker	accuracy prediction
pot	0.5172
aapl	0.5172
gs	0.5517
mos	0.5517
ibm	0.5172
msft	0.7586
gg	0.5862
bac	0.5862
goog	0.5517
c	0.5172
sp_500	0.5172

Table 1: Results from HMM using K-means learner

(Trained from 01/2004-09/2009 / Tested from 09/2009 – 11/2009)

Although the results are very promising, we must note a major issue with this approach. Since we use the k-means learner algorithm on the lagged profits, it just so happened that the 5 hidden states it chose where the 5 that we wanted. If we were to add additional observation features in, such as EMA, MACD, etc, we would not be able to control what the 5 states are. Because of this issue, we started using the Baum-Welch algorithm.

2.1.2 Hidden Markov Model with Baum-Welch and Viterbi algorithms

In order to leverage the full power of the Simlio engine and HMMs to potentially develop more meaningful predictions, we decided to move away from the k-means learner. In our second experiment, we trained the HMM with the Baum-Welch algorithm and then used the Viterbi algorithm to detect our current most likely state in

order to check how accurate our model is. Again, we used our lagged profit as the sole input.

For this experiment, we mimic the idea of a "dynamic training pool" that is presented in Y. Zhang's paper[8]. Essentially, our training window will always be of the same constant size and we would shift the training pool across time. We chose this approach because we want to train the new test data instances one at a time after we have done predicting whether it was accurate for that day. This way, we will include our most recent data in the training pool, and it will hopefully always detect new patterns and put the same emphasis on it, since the training size is constant.

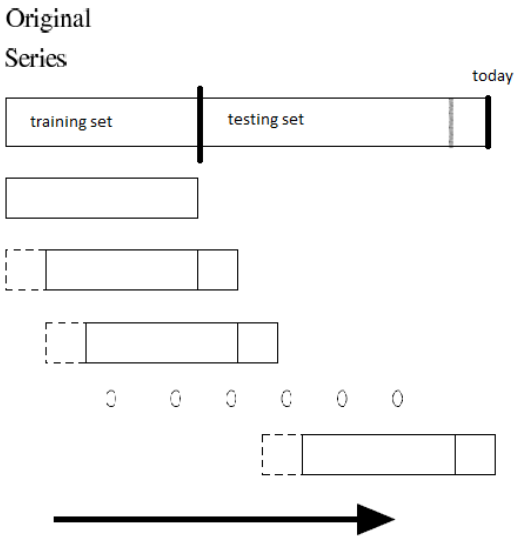


Figure 7: Dynamic Training Pool

After training the HMM on data from 01/01/2004 to 180 days ago, we started testing the model on the most recent 180 days once at a time. After every test, we popped off the oldest training data instance and we appended the recent tested test instance, keeping the training pool the same size. Then we retrain and test again. We do so until we finished testing the prediction accuracy on the last 180 days.

Table 2 summarizes the result of our experiment. As we can see, the average accuracy prediction is roughly around 53%. There are several factors that could account for why the accuracy is so low:

1. We only used lagged profits as our inputs
2. It has no concept of recent news and how it could potentially affect the market
3. Trends of stocks may go beyond just one day. A stock's movement up or down yesterday is probably not an indicator of what it would do tomorrow

ticker	accuracy prediction
pot	0.5167
aapl	0.5611
gs	0.5
mos	0.55
ibm	0.5222
msft	0.5778
gg	0.5556
bac	0.5167
goog	0.5517
c	0.5222
sp_500	0.5172

Table 2: Results from HMM using Baum-Welch and Viterbi algorithms

Tested and retrained for last 180 days

Given that our results are mediocre, we decided to look into combining the k-means learner with Baum-Welch and Viterbi.

2.1.3 Hidden Markov Model with K-means, Baum-Welch, and Viterbi algorithms

One issue that we ran into for the second experiment is that the Baum-Welch algorithm required us to set an initially matrix. Using this matrix, the BW algorithm will iterate through the EM iterations to find the best parameters for the model. However, that initial matrix is not always trivial to setup and giving the wrong initial matrix could mean that the Baum-Welch algorithm will find the wrong local maximums. Thus, we decided to utilize the k-means learner to initialize the matrix for us before we run the Baum-Welch algorithm.

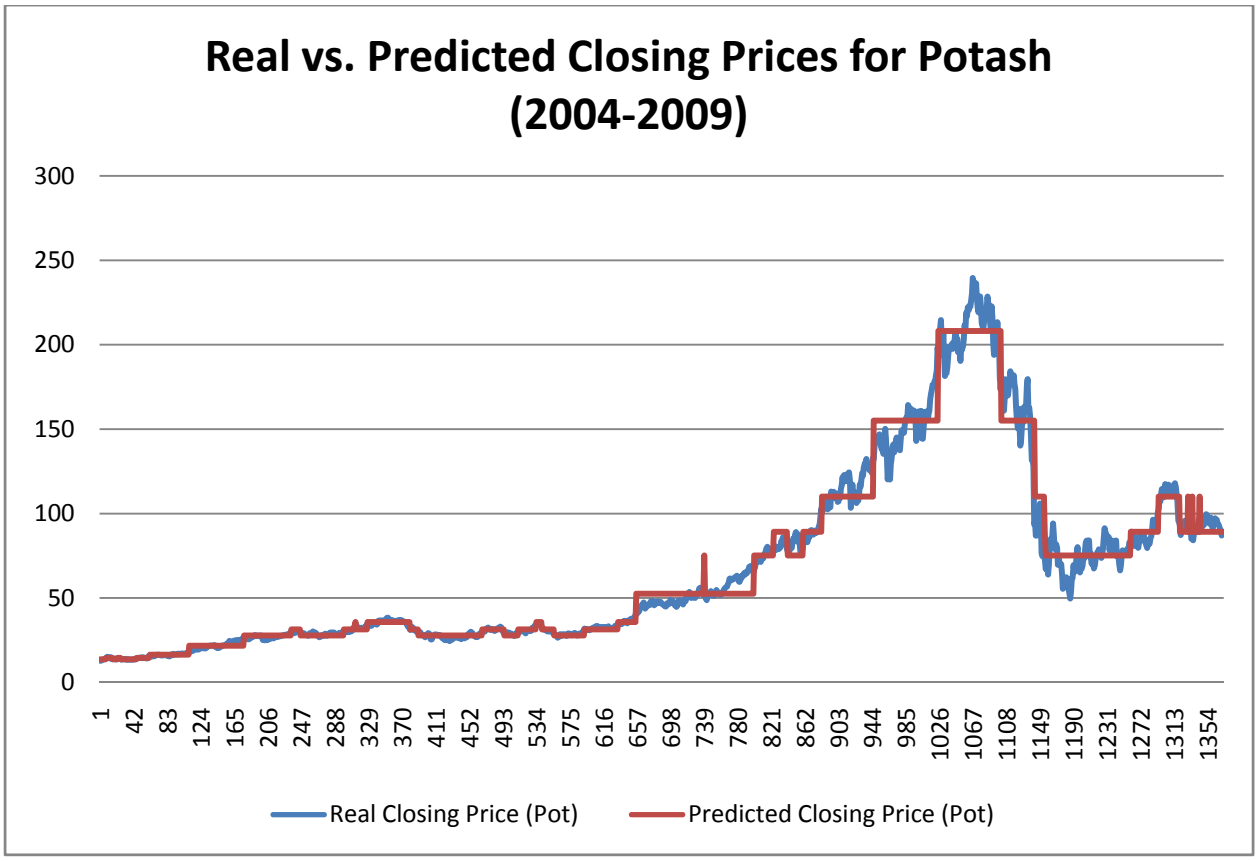


Figure 8a: HMM's Predicted Closing Prices vs. Real Prices for Potash
(01/01/2004 – 10/31/2009)

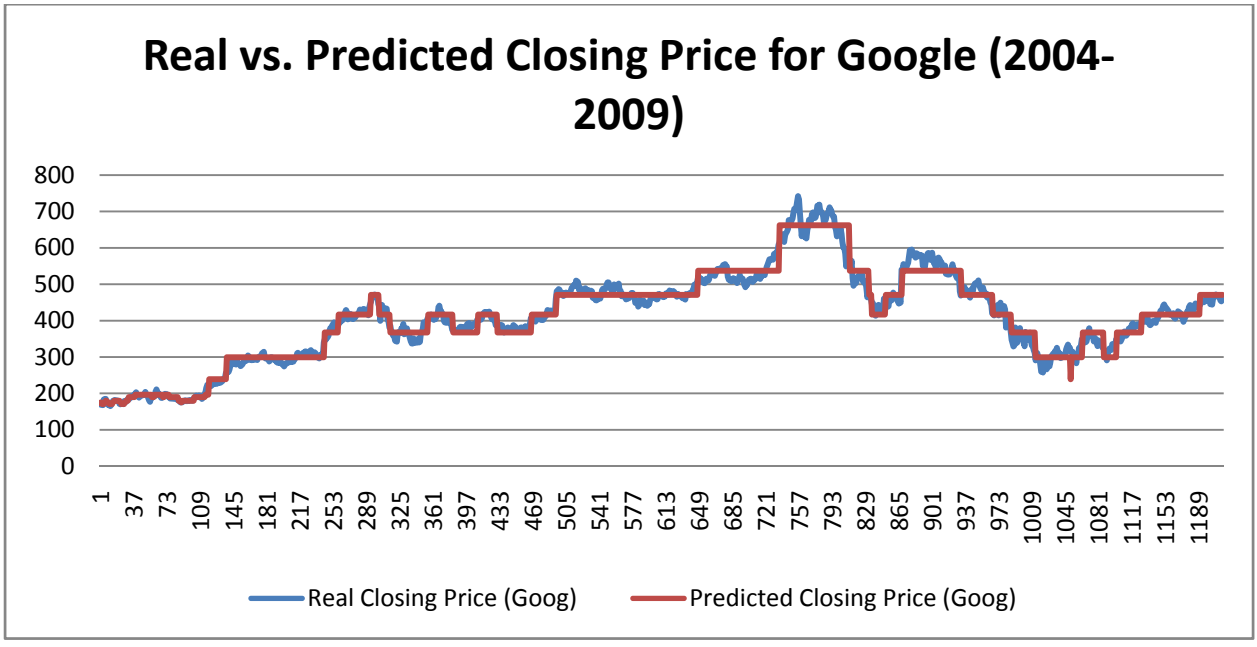


Figure 8b: HMM's Predicted Closing Prices vs. Real Prices for Google
(01/01/2004 – 10/31/2009)

Using Viterbi's algorithm on the data, we were able to demonstrate what the HMM's predictions are for the prices from 2004-2009. We drew out the predictions on the figures above.

Figure 8a is for a stock called Potash and Figure 8b is for Google. For both the stocks, we let the K-means algorithm find around 10+ states. Afterwards, we used the Baum-Welch algorithm to train on the HLOC (high, low, open and close prices), assuming Gaussian distribution, and figure out what the best parameters are for the model. Lastly, we decided to validate the accuracy on the training data by using Viterbi's algorithm at each step. From the graphs above, HMM was able to decently predict the stock movement. This warrants more research in this field, and this makes the statement that the market is always efficient somewhat suspect.

2.1.4 Support Vector Machines with Technical Indicators

Although our HMM results were relatively impressive, we have done more research and researchers have suggested the SVMs outperforms HMM and other machine learning methods by a significant margin. We decided to use libraries JavaML[19] and LibSVM[20] to investigate that claim.

In order to train the SVM with a radial kernel, we first researched on what parameters to use. Based on K. Kim's paper, the best C is 78 and gamma is 25[14]. With these two parameters, Kyoung-jae was able to build a model that predicts test data with 57.83% accuracy on average. In order to replicate similar results, we leveraged the Simlio engine to give us the prices and technical indicators of any stock we want.

In this experiment, we decided to train the SVM on the following features:

1. EMA7
2. EMA50
3. EMA200
4. MACD
5. RSI
6. ADX
7. Lag profits
8. High
9. Low
10. Closing price > EMA200

We trained the SVM from 01/01/2004. While training, we classify each instance as a buy or sell signal. We stopped training on X days before 11/1/2009, where X is varied from 30 to 180 (1 month to 6 months). We then test our trained model on our test data, which is the X number of days we left out of our training. Initially, we did not retrain after testing each test data instance. We will show the results of several experiments. We tried using the linear kernel first, and then we changed it to the RBF kernel. We also tried tweaking the parameters in other ways besides what was suggested by Kyoung-jae.

C	1
gamma	0
kernel	linear

ticker	accuracy prediction
pot	0.5333
aapl	0.5
gs	0.5333
mos	0.5667
ibm	0.5
msft	0.5333
gg	0.6333
bac	0.6333
goog	0.5667
c	0.5
sp_500	0.5667

Table 3: Results from SVM with Linear kernel

Test data: Last 30 days

C	100
gamma	78
kernel	rbf

ticker	accuracy prediction
pot	0.5667
aapl	0.5333
gs	0.6
mos	0.5
ibm	0.6333
msft	0.8333
gg	0.5333
bac	0.5667
goog	0.5333
c	0.5333
sp_500	0.5333

Table 4: Results from SVM with RBF kernel 1

Test data: Last 30 days

C	78
gamma	100
kernel	rbf

ticker	accuracy prediction
pot	0.5667
aapl	0.5333
gs	0.6
mos	0.6
ibm	0.5667
msft	0.8
gg	0.5333
bac	0.5667
goog	0.5333
c	0.5
sp_500	0.5333

Table 5: Results from SVM with RBF kernel 2

Test data: Last 30 days

C	78
gamma	25
kernel	Rbf

ticker	accuracy prediction
pot	0.5333
aapl	0.5
gs	0.6333
mos	0.5333
ibm	0.6333
msft	0.6333
gg	0.6333
bac	0.5667
goog	0.6667
c	0.5333
sp_500	0.5333

Table 6: Results from SVM with RBF kernel 3

Test data: Last 30 days

Table 3 shows the results of using a plain linear kernel. With that result, we can already see that SVMs may have more potential than HMMs.

When we switched our kernel to RBF, we started getting even better consistent results. For instance Table 4 and 5 differ because the parameters are slightly changed. It turns out the Table 4 seems to be better than 5 on average. For Microsoft, it even has an 83.33% prediction accuracy rate in the last 30 days when the SVM predicted whether or not to buy/sell the next day.

We use Kyoung-jae's suggested parameters in Table 6, and we get a rather consistent prediction average of around 60% for most stocks. This is extremely promising, but we wanted to make sure that this was not just a coincidence since we only tested the model on 30 days. Taking Professor Satish's advice, we decided to look into the situation more, and we started testing on 180 days / 6 months worth of data.

Recall that for the experiments that we did above, we did not retrain our model after running through the tests each day for 30 days. This was probably fine for one month because it is short. However, since we are now testing on 180 days worth of day, new patterns and trends could definitely arise during that period and it might significant to retrain the model to take them into account when we test the later half of the testing data. We ran both experiments – one without retraining and one with retraining.

Comparing the two results on the right, we see that in general, the one without retraining performs around 50%, almost like a coin toss. The experiment with retraining performs better in almost all cases. More importantly, the prediction accuracy for the S&P500 is as high as 60%, which definitely warrants further research.

2.1.5 Support Vector Machines with Technical Indicators and News

We were able to add a module to the Simlio engine that crawls the web and parses out news articles of any stock for every day. We started to incorporate news and add it as an additional feature to the SVM. However, due to time constraints, we decided to keep it simple and just use the number of news articles as a feature of the SVM. Our hunch was that if something important happens to a company (whether bad or good), there will be a lot of news about it the previous day. Hence, knowing this, perhaps the SVM can detect when there could be a relatively big jump in prices the next day.

Based on our results from the experiment, it seems that there could be some potential to our idea. In most cases, the retrained version outperforms the other by a few percentages, which could be enough to be significant.

C	78
gamma	25
kernel	rbf

ticker	accuracy prediction
pot	0.5028
aapl	0.5251
gs	0.5196
mos	0.5251
ibm	0.5307
msft	0.5363
gg	0.5698
bac	0.5363
goog	0.5978
c	0.5195
sp_500	0.5028

Table 7: Results from SVM with RBF kernel 3
Test data: Last 180 days. No retraining

C	78
gamma	25
kernel	rbf

ticker	accuracy prediction
pot	0.5667
aapl	0.6
gs	0.5667
mos	0.5333
ibm	0.5333
msft	0.6666
gg	0.6
bac	0.5
goog	0.5
c	0.5
sp_500	0.6

Table 8: Results from SVM with RBF kernel 3
Test data: Last 180 days. With retraining

C	78
gamma	25
kernel	rbf

ticker	No News	News
pot	0.5333	0.6
aapl	0.5	0.5333
gs	0.6333	0.6667
mos	0.5333	0.5333
ibm	0.6333	0.5
msft	0.6333	0.6667
gg	0.6333	0.5333
bac	0.5667	0.5
goog	0.6667	0.6
c	0.5333	0.6333
sp_500	0.5333	0.6667

Table 9: Results from SVM using both no news and with news (RBF kernel 3)

Test data: Last 30 days. Without retraining

C	78
gamma	100
kernel	rbf

ticker	No News	News
pot	0.5667	0.7
aapl	0.5333	0.6667
gs	0.6	0.5333
mos	0.6	0.6
ibm	0.5667	0.5333
msft	0.8	0.6333
gg	0.5333	0.6333
bac	0.5667	0.6333
goog	0.5333	0.5667
c	0.5	0.5667
sp_500	0.5333	0.7

Table 10: Results from SVM using both no news and with news with (RBF kernel 2)

Test data: Last 30 days. Without retraining

Comparing the results of using news as a feature for the SVM on Table 9 and 10, we see that using news does change our prediction accuracy. In most cases, it made the SVM accuracy even higher. Using the parameters that Kyoung-jae suggested, we were able to increase the accuracy by around 3% for most stocks, and notice that we have increased the accuracy of predicting the S&P500 index to 67%!

We did some further test by tweaking more parameters. We realized that if we deviate from Kyoung-jae's parameters and use C=78 and gamma=100, we would further increase the average prediction accuracy. Using these parameters, the SVM has a 70% of accurately predicting whether to buy or sell the S&P500 index every day for the last 30 days!

3. MANIPULATION DETECTION

Historically, there have been many cases with stock investors trying to artificially influence stock prices. In this final section of the paper, we will investigate how abnormal changes in volume could potentially suggest a manipulator in the market, and use that knowledge with our SVM engine to see if there is a way to "ride the wave" with a manipulator to gain some additional profit.

3.1 Introduction to Manipulation Theories

3.1.1 The Rational Model

Allen and Gale[21] have classify 3 different types of manipulations: information-based, action-based, and trade-based. The first two are now regulated by the SEC, so chances of them happening, such as insider trading, has been significantly reduced. However, every now and then, we will see cases like Enron.

On the flip side, trade-based manipulations are much harder to detect due to the difficulty in distinguishing a large trader from a manipulator. Both of them will probably have enough capital to buy/short large volumes of any particular stock, so

to a regular investor, he is uncertain whether or not there is a manipulator in the market.

Allen and Gale's model assumes that the regular traders are rational and that there is information asymmetry because large traders (whether manipulator or not) have access to private information. Under these conditions, there are 3 steps:

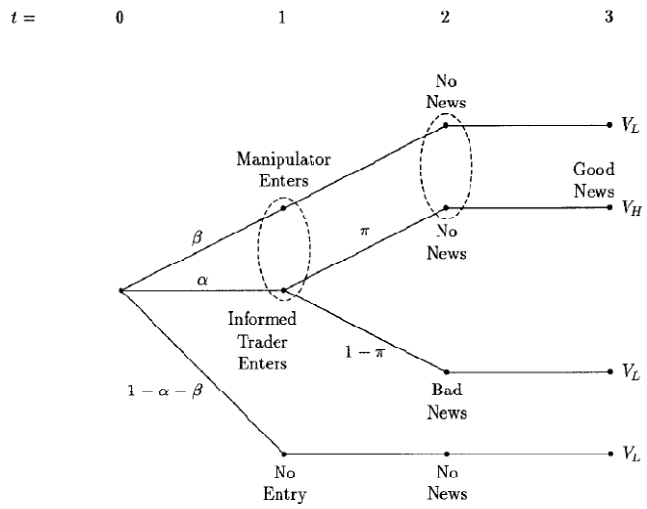


Figure 9: Allen and Gale's Model[21]

At time t=0, there is no large trader in the market. At time t=1, there is an α probability that a informed large trader enters and a β probability that a manipulator enters. Since the manipulator has the resources to mimic a larger trader, the dotted circle represents that in the common investor's perspective, there is an uncertainty whether or not the large trader is a manipulator or not. Then, at t=2, the manipulator will take profits and t=3 is when the true value of the stock is revealed. If the manipulator was in the market before, the price of the stock will drop back down to its fundamental value V_L. If there are good news, we'll see that the true value of the stock is actually high, V_H.

The model is used by Allen and Gale to prove that under the conditions stated above, there can exist a pooling equilibrium in which the manipulator can successfully mimic a large trader and thus always achieve a positive profit. The proof of this claim is left for the reader to read in their paper.

3.1.2 The Behavioral Model

Complementing Allen and Gale's model, Mei, Wu and Zhou[22] incorporated behavioral studies into the model. They did so because there are a large number of cases where the asset prices deviate from their fundamental values and it was difficult to explain using the rational model. Thus, they investigated this situation and utilized the fact that people do not always act rationally to try to explain anomalies in the previous model.

They used various empirical studies to show how manipulators can exploit the irrational traders' behavior biases. For instance, using Jegadeesh and Titman[24] report, which suggests that investors can make substantial abnormal profits by selling past losers and or buying past winners, Mei, Wu and Zhou suggests that manipulators can exploit this behavior bias to cash in some profit. Since these investors are momentum traders, they will buy fewer shares in a down market. Furthermore, when these traders experience a downward market, they are more inclined to keep the shares instead of selling them in hopes that the market will bounce back up. Incorporating this behavior bias into the model makes it more realistic and thus more applicable in real world situations.

This new model utilizes a couple of key assumptions:

1) Time begins at t=0 and ends at t=T. Behavioral traders enter the market at the beginning of each time period and they are price takers. Each has a probability of q_1 of buying a share if $P_t > P_{t-1}$, and q_2 if not. Due to the behavioral bias, we assume $q_1 > q_2$. Behavior traders like to take immediate profits. So the moment $P_t > P_0$, they will sell their shares. Also, if $P_t >= P_{t+k}$ (k =

number of days he has kept his shares), there will be a probability of $q_3<1$ to take a loss and liquidate all his shares at time t+k. Moreover, at t=0, the price of the stock is at its fundamental value.

2) Manipulator enters the market at t=1 without any prior shares.
3) Arbitragers enters at t=1. Shares are traded based on recent price movements. Arbitragers keep the market fluid because if prices increased, they will sell some shares to take profits. If prices decreased, they will buy more shares. They will submit the following order at time t:
   $D_{a,t} = -a(P_t-P_{t-1}) = -a(\Delta P_t)$
   Where a>0 and a = arbitrage parameter
4) A manipulator tends to move the asset price by $\delta > 0$ for t_u consecutive days. He will also start liquidating his shares at time $t=t_u+1$ until T-1 where he sells all his shares. Therefore, we will let $t_d = T-1-t_u$ or the length it takes for the manipulator to liquidate all his shares.
5) Market ends at time = T and investors receive the wealth accumulated from liquidating their positions.

To help see the assumptions more clearly, let's look at a chart given shown in their report:

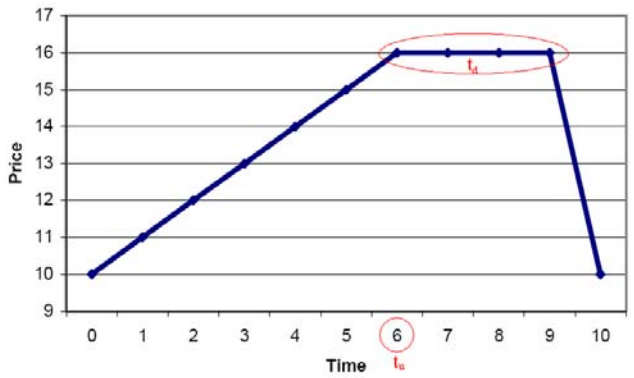


Figure 10: Typical Manipulated Price Fluctuations (Parameters: $t_u = 6$, $t_d = 3$, a = 0.1, $q_1 = 0.8$, $q_2 = 0.4$, $q_3 = 0$)

In figure 10, we see that $t_u = 6$, which means the manipulator is moving the prices of the shares up 6 consecutive days by purchasing shares. $t_d = 3$ means that it takes 3 days for the manipulator to liquidate all his shares and take a profit at the high price. Lastly, T=10 means that the manipulation ends at day 10 and the investors have realized that the fundamental price is actually at its initial value and the sudden jump in price before was due to a manipulator in the market.

Using the assumptions to solve the model, Mei, Wu and Zhou focused on two key propositions:

1) By the end of t_u, the manipulator has accumulated $N=a*t_u*\delta$ shares with an average cost of $P_0 + \frac{1+t_u}{2}\delta$ / share. We get this from assumptions 1-3 because the manipulator buys $a*\delta$ shares at price $P_0 + t\delta$. Recall that the manipulator tends to move the price by δ and in order to do so, they submit an order of $a\delta$ shares each time.
   IMPLICATION: If the arbitrage parameter is large, the manipulator must be very wealthy to move the market. If $a\rightarrow\infty$, there is "no limits to arbitrage" and thus, there is no way for the manipulator to move the market.
2) Assuming $q_3=0$ (a.k.a behavior traders are extremely unwilling to take losses – see assumption #1), manipulators can sell at the high price of $P_3 = P_0 + t_u\delta$ from $t=t_u+1$ to t=T-1 (a.k.a the period of t_d). Therefore the profit becomes the number of shares * (price sold – average price bought):

   $\pi=N*[P_{t_u} - (P_0 + \frac{1+t_u}{2}\delta)]$

   After simplifying, we get:

   $\pi=N*(\frac{t_u-1}{2}\delta) = a*(\frac{t_u-1}{2}t_u)*\delta^2$

IMPLICATION: From t=1 to $t=t_u$, the price increases by δ for each period. Past behavioral traders sell q_1 shares at each period to take profits because the price went up while new behavioral traders buy q_1 shares. Arbitragers will sell aδ shares (from assumption #3), and thus for the manipulator to move the price up, they have to buy aδ shares in each time period up to $t=t_u$. Furthermore, from $t=t_u+1$ to t=T-1, past behavioral traders already bought shares at the highest price at t_u. Thus, they won't sell because $q_3=0$. Price stays constant so arbitragers won't trade either. Thus, new behavioral traders in these periods will buy q_2 shares from the manipulators while they take profits. By t=T, the manipulators will have sold all their shares and the price of the stock will return to its fundamental value.

To summarize, this model shows that under a few assumptions, the manipulator will be able to mimic a large trader successfully and make some decent profits. This model goes beyond Allen and Gale's model because this one factors in empirical studies of how people behave in the market – mainly that many investors are momentum traders and thus they like to buy when the market is going up. Moreover, these behavioral traders try to avoid selling their shares at a loss, since they all hope that the down market will recover soon.

3.1.3 The Information-Seeker Model

Aggarwal and Wu[23] extended Allen and Gale's framework by considering the case of what happens when a manipulator can trade in the presence of other rational investors who are able to seek information about the fundamental values of a stock. In other words, what happens if information-seekers, who are knowledgeable of the market, enter the markets too? Will the manipulators still be able to consistently make profits? The answer is: yes.

In this particular model, we have three types of investors:

1) Informed Party (I)
    a. Truthful large traders (T)
    b. Manipulators (M)
2) Information Seekers ("Arbitragers") ($A_i \in N$)
3) Uninformed Traders (U)

The N numbers of information seekers all observe past prices and volumes, and they are susceptible to rumors. Moreover, they have no access to fundamental information. Lastly, the uniformed traders provide liquidity to the market, so whenever someone wants to make a trade, we can be certain that the transaction will happen.

Before we discuss the model, we will state a couple of key terms and assumptions:

Market Price: P(Q) = a+bQ, where

a=price of stock if no one buys any shares

b=slope of supply curve

Q=quantity demanded

Total Share Outstanding: $\frac{V_H - a}{b}$, where

V_H=price of stock if someone buys all the shares

t=0: All shares are held by uninformed traders

t=1: Informed trader may enter the market:

Prob(M) = γ

Prob(T) = δ = Prob(V_H) – Value High

Prob(~I && V_L) = 1- γ – δ

a = unconditional expected value of final cash flows = $\delta V_H + (1- \delta)V_L$

t=2: Information seeks can buy/sell shares

t=3: Fundamental stock price is reveal to be V_H or V_L. The cost of holding shares of stock at this period is k.

Now consider a model with manipulators, information seekers, and uninformed traders in the economy. A_i's posterior beliefs that the purchaser of the shares at t=1 is:

$$\beta=\frac{\gamma}{\gamma+\delta}$$

Each arbitrager (A_i) will solve the following problem at t=2:

$$max_{q_2{}^{\wedge}i}\ [(1-\beta)\{V_H{*}q_2{}^{\wedge}i\text{-}[a+b(\textstyle\sum_{i\in N} q_2{}^{\wedge}i)]\ q_2{}^{\wedge}i\}$$
$$+\ \beta\{V_L q_2{}^{\wedge}i\text{-}[a+b(\textstyle\sum_{i\in N} q_2{}^{\wedge}i)]\ q_2{}^{\wedge}i\}]$$

Imposing symmetry on the total shares outstanding, we calculate the following of the information seekers:

$$q_2{}^{\wedge}i* = \frac{V_H-a}{(N+1)b} = \frac{(1-\beta)V_H+\beta V_L-a}{(N+1)b}$$

Aggregate Demand= $Q_2 = q_2{}^{\wedge}* = \frac{N}{N+1}\frac{(1-\beta)V_H+\beta V_L-a}{b}$

$$P_2{}^* = a + \frac{N}{N+1}[(1-\beta)V_H + \beta V_L - a]$$

$$\pi^{\wedge}i* = (1-\beta)\{V_H{*}q_2{}^{\wedge}i\text{-}(a+bQ_2)\ q_2{}^{\wedge}i\}$$
$$+\ \beta\{V_L q_2{}^{\wedge}i\text{-}(a+bQ_2)\ q_2{}^{\wedge}i\}$$
$$= \frac{[(1-\beta)V_H+\beta V_L-a]^2}{(N+1)^2 b}$$

The informed party (either M or T) will solve the following at t=1:

$$max_{q_1} P_2{*}q_1\text{-}(a+bq_1)q_1$$

$$q_1{}^{M*} = q_1{}^{T*} = \frac{(1-\beta)V_H+\beta V_L-a}{2b}$$

$$P_1{}^* = a + \frac{N}{N+1}\frac{(1-\beta)V_H+\beta V_L-a}{2}$$

$$\pi^{M*} = \pi^{T*} = \frac{N^2}{(N+1)^2}\frac{[(1-\beta)V_H+\beta V_L-a]^2}{4b}$$

Note that in order for this pooling equilibrium to be sustainable, the truthful trader's actions must not deviate from that of the manipulator's. Thus, the truthful trader must want to sell shares at t=2 than hold them till t=3 since that is what the manipulators will do (take profits). The value for holding shares at t=3 is $V_H - k$, where k is the cost for holding shares given from the above assumptions. So, the "incentive compatibility" condition is:

$$P_2{}^* = a + \frac{N}{N+1}[(1-\beta)V_H + \beta V_L - a] \geq V_H - k$$

Using $a= \delta V_H + (1- \delta)V_L$ and $\beta=\frac{\gamma}{\gamma+\delta}$, the pooling equilibrium is sustainable if:

$$\frac{k(N+1)-(1-\delta)(V_H-V_L)}{-k(N+1)+(1-\delta)(V_H-V_L)+N(V_H-V_L)}\delta \geq \gamma$$

IMPLICATIONS:

1) ↓ in γ: The more likely that the purchaser at t=1 is a manipulator, the less likely of pooling
2) ↑ in δ: The more likely that a truthful large trader enters, the easier it is to sustain pooling
3) ↑ in k: The higher the cost of holding the shares at t=3, the more likely the truthful trader will pool with the manipulator
4) ↓ in $(V_H\text{-}V_L)$: The greater the dispersion, the more valuable it is for the truthful trader to wait till t=3; hence, decreasing the chance of pooling
5) ↑ in N: The more information seekers there are, the better the price is at t=2. Thus, both manipulators and truthful traders will be tempted to take profits at the time, increasing the chances of pooling

Through this model, Aggarwal and Wu have shown that even with information seekers in the economy, the manipulator can still make a profit. As long as the pooling equilibrium is sustainable, it will be difficult for traders to distinguish a truthful large trader from a manipulator.

3.2 Attempt to Detect Abnormal Volumes

3.2.1 Concept

The theoretical models are only practical if they can be useful in real life situations. Thus, we leveraged the Simlio engine to help us check how often manipulation cases actually happen in the recent years.

Instead of checking the legitimacy of all the models, we have decided to combine ideologies from all 3 frameworks. To check if there had been any manipulations, we created the following rules:

1) First, we check to see if there are any abnormal volumes. We define abnormal volume to be:

   $Vol_{today} > Vol_{200\ moving\ average}$

2) For all days with abnormal volume, we check to see what the prices were for X previous days. According to the above models, "today" is the day the manipulators entered; hence the abnormal volume. Thus, the prices the previous days are the fundamental prices of the stock
3) We also check to see if prices have increased today. This is because as manipulators buy a lot of shares, they also move the prices up. For the purpose of this experiment, we will assume that as the price increases more and more, the manipulators are still buying. The moment the prices start falling is the when the manipulators begin to sell.
4) Lastly, we check to see if the prices fall back to their fundamental price. If so, we have found a potential period of time where this particular stock has been manipulated. If not, then most likely a truthful trader had entered and thus the price stays at V_H.

To see an example of how this experiment works:

Detection	Date	Volume	Price	Notes
0	20090911	61935	89.8	t=0
0	20090914	63620	88.75	
1	20090915	113153	93.85	t=1
0	20090916	100348	94.18	
0	20090917	104319	96.51	
0	20090918	61286	97.14	t=2
1	20090921	124496	93.09	
0	20090922	72236	94.3	
0	20090923	59021	92.92	
0	20090924	54149	90.86	t=3

Table 11: A Manipulation Case Found Through Experiment for Stock: Potash

Table 11 shows an example of a potential manipulation case that the Simlio engine has detected for the stock Potash. From Sept 11, 2009 to Sept 24, 2009, we see that the fundamental price of Potash went from $89.8 to a high of $97.14 and dropped back down to $90.86. Analyzing the notes column:

1) At t=0, no large traders or only uninformed traders are in the market.
2) At t=1, there's a volume spike. We interpret this as a large trader has entered the market.
3) At t=2, the price of the stock has reached its climax. A large trader will keep his shares till t=3; hence, maintaining the high price. A manipulator will start selling now, causing the price to decreases.
4) At t=3, we see that the price has essentially returned back to its fundamental value. Thus, we conclude that this temporary price spike is due to a manipulator in the market.

The Simlio engine allows us to give it any ticker symbol, and it will report all potential manipulation cases like the one shown in Table 11. In the next section, we will analyze our results.

3.2.2 Results and Model Confirmation

We decided to run the automatic manipulation detector on the following stocks below, since we used them for our HMM and SVM experiments too. In our first attempt, we have two filters.

The first filter says that the manipulator must be a large trader; hence, he will buy a lot of shares, causing an abnormal spike in volume. This will cause the price to increase and the first filter says that this price increase must be at least $2.

The second filter says that the manipulator would only successfully manipulate the market if he profits from this scheme. So the second filter is set to $2/profit per share.

If both filters are met, the Simlio engine marks the period of time as a possible case for manipulation activities to happen. Using these two filters, we get the following results:

Stock	Cases	Year Range:
goog	6	2004 to 2009
pot	4	1996 to 2009
aapl	9	1996 to 2009
gs	9	1999 to 2009
mos	6	1996 to 2009
msft	2	1996 to 2009
gg	5	1996 to 2009
wfc	1	1996 to 2009
bac	3	1996 to 2009
sp_500	44	1996 to 2009

Table 12: Manipulation opportunities over the years (profit of over $2 per share)

As one can see from Table 12, profitable manipulation cases do happen quite often over the years. Considering that each manipulation case allows the manipulator to profit $2/share and they probably buy thousands of shares at a time, this can be quite a profitable scheme.

We decided to run another experiment with the same two filters, but with different values. The first filter is the same: when the manipulator purchases stocks, it must be at a high enough volume that the price of the stock increases by at least $2. We changed the second filter so that the profit must be at least $4/share instead of $2/share. With these parameters, we get the following results:

Stock	Cases	Year Range:
goog	6	2004 to 2009
pot	3	1996 to 2009
aapl	7	1996 to 2009
gs	2	1999 to 2009
mos	1	1996 to 2009
msft	0	1996 to 2009
gg	1	1996 to 2009
wfc	1	1996 to 2009
bac	1	1996 to 2009
sp_500	37	1996 to 2009

Table 13: Significant manipulation opportunities over the years (profit of over $4 per share)

Comparing Table 12 to Table 13, we see that there are fewer cases for our second experiment, but for each one of these potential manipulation cases, the manipulator will be able to profit $4/share. Assuming that manipulator buys 100,000 shares per case and that he only plays the S&P500 index, he averages around 3 manipulation cases per year with a profit of:

3 cases * 100,000 shares/case *4 dollars / share

= $1,200,000 annually

Those numbers are quite conservative considering that the manipulator is a large trader. Moreover, if we factor that the manipulator can make an average profit of $1.2M annually on just the S&P500 alone, and that there are thousands of other stocks and indices the manipulators can play with, we can easily project the manipulator's profits to be in the hundreds of millions annually!

3.3 Incorporation with SVM Engine

3.3.1 Adding the Abnormal Volume Detection Feature

Now that we have analyzed a couple manipulation models and the Simlio engine can tell us when there are abnormal volume spikes, we decided to add this into our SVM. Before we examine the results, we will first discuss how we will train the SVM and what we will test it on.

Since the Simlio engine reports a 1 when there is an abnormal volume spikes and 0 otherwise, we figured that this will work perfectly as a feature for the SVM. Thus, besides the 10 features we used in our SVM experiment discussed on page 12, we also added an abnormal volume feature to the list. Our idea is that we hope the feature will help the SVM better detect when to buy/sell stocks. Since we know that if we have successfully detected a manipulation case, the price of the stock will shoot up soon after the abnormal volume detection. Afterwards, it will return back to its fundamental price if there was a manipulator in the market or the price will stay high if the larger trader was a truthful one. Using this additional feature, perhaps the SVM will be able to learn from past manipulation cases. For our experiments, we will train the SVM on a couple years worth of data and test the model on the most recent 6 months of data.

After that experiment, we will also incorporate the news features and see if using both features will help the SVM even more. Although the idea sounds promising, we must mention one issue that we can foresee. All our SVMs are trained to tell the user whether or not to buy/sell stocks the next day. However, the abnormal volume detection feature does not necessary tell the investor when to buy stocks the next day. When there is a 1, we know that a large trader may have entered. All the other days will return a 0 even if a large trader just entered. In other words, due to the fact that our manipulation detection only suggests what will happen for a period of time instead of tomorrow, the SVM may get confused from the feature. Let us see what the results are.

3.3.2 The Results / Analysis

C	78
gamma	100
kernel	Rbf

ticker	VolDetect W/O News	VolDetect W/ News
pot	0.52	0.51
aapl	0.54	0.57
gs	0.53	0.53
mos	0.51	0.54
ibm	0.51	0.54
msft	0.5	0.5
gg	0.55	0.51
bac	0.54	0.54
goog	0.52	0.53
c	0.51	0.51
sp_500	0.54	0.54

Table 14: Results from SVM using volume detection for both no news and with news with (RBF kernel 2)

Test data: Last 180 days. Without retraining

Table 14 shows the accuracy results of testing our most recent 180 days of data on the models our SVM trained. As we can see on the left column, adding the volume detection feature without news produces mediocre results. The average accuracy is around 53%. On the right hand side, if we add both features, we get a slightly better accuracy prediction of around 54-55%. However, both sets of results do not surpass what we got earlier in Table 10. We suspect that the reason for this is because the abnormal volume detection is not a great feature to use in our SVM. We have developed our SVM to predict what investors should do the next day. However, our manipulation detection models that we surveyed in the previous section span multiple days. Moreover, whenever the detector reports a 1, we know that there has been an abnormal volume spike. However, when it reports a 0, there are mix interpretations. For example, it could be 0 because nothing special has happened. It could also be 0 the days after an abnormal spike because even though the price increases the next few days, the volume

does not deviate enough from the 200 moving average, so it still reports a 0. Thus, due to the multiple interpretations of the results with zeros, the SVM might have gotten confused on what to predict the next day. Nonetheless, this was still an interesting experiment because if our SVM models were different, such as instead of predicting what to do the next day they forecast the general trends in prices for the next few days, than perhaps this feature would have been of great significance. However, due to our time constraints, we were unable to explore this further, but the experiments that we have done do show that this is worthy of future research.

4. CONCLUSION

Traditional financial models answer many questions we may have about the financial markets. However, many suffer from over-simplistic assumptions that do not truly reflect the real state of the financial market. The two main problems associated with the conventional approaches described in this paper are: symmetric information and limited input factors. With these two problems, the traditional approaches can only provide a macro-representation of the market.

Artificial Intelligence is seen as a plausible approach to financial modeling that overcomes the two flaws of traditional models. There has been an increase in interest in financial prediction markets, where many hedge funds have been using similar models discussed in this paper to make quick inform decisions for high frequency trading. Although no model discussed so far is close to predicting stock movements perfectly, the results shown in this paper warrant that more research should be done. If the EMH were true, there should be no way for an investor to gain an edge by using technical indicators and news. However, as we see from the results (especially with news), we were able to train the SVM to accurately predict the S&P500 up to 70%!

5. OPEN QUESTIONS / FUTURE WORK

1) Instead of just using the number of documents on Google News for a particular stock on a specific day, we can modify the TF-IDF algorithm to give us the important terms for a particular stock and use that as features in our SVMs. We have already implemented a prototype of this, but due to time constraints, we were unable to finish this portion. We believe there is much potential in this area because R. Cooley's paper demonstrated that using SVM with TF-IDF resulted in a 96.75% accuracy in classifying news stories[16]. After classifying them, we could potentially pass them in as features to SVM and that might enhance the ability of our SVM to accurately forecast stock movements.
2) For the HMM model, we assumed that the observations or inputs follow a Gaussian distribution. However, the technical indicators or even the stock prices may not follow such a distribution. We should plug in other distributions to see if they would help the HMM better forecast stock movements.
3) For the SVM approach, we used the RBF kernel because papers have suggested that results are superior than using other basic kernels. However, there are so many possible parameters to tweak that it is not entirely impossible that some other kernel might allows the SVM to perform better when forecasting stock movements.
4) Code up a random walk algorithm and compare that result to the results we got from using HMM and SVMs. Is it possible that we got lucky and just so happened to get promising results? More research will need to be done before we can confidently conclude anything.
5) Instead of training SVM models on predicting stock movements for the next day, train it to predict general trends for the next few days. With that, we can try to add in the abnormal volume detection feature and see if that would help the SVM better predict future stock trends.

6. ACKNOWLEDGMENTS

We thank Simlio LLC and all its team members for providing many useful market tools to allow us to graph and display market trends with ease. We also thank Professor Peter Bartlett for providing many useful lectures about statistical graphical models, such as exponential families and maximum likelihood, and for helping us get started on this topic. Furthermore, Professor Satish Rao played a crucial role in helping us determine what feature sets to try as well as suggesting different trading algorithms. Satish's ideas were instrumental in developing the manipulation detection section of this paper.

7. REFERENCES

[1] F. Allen and S. Morris, *Finance Applications of Game Theory*, Game Theory and Business Applications, Sept 1998

[2] G. S. Atsalakis and K. P. Valavanis, *Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods*, Expert Systems with Applications, 2009

[3] C. Truong, *Value Investing Using Price Earnings Ratio in New Zealand*, University of Auckland, Business Review Volume 11 No.1, 2009

[4] B. G. Malkiel, *The Efficient Market Hypothesis and Its Critics*, Princeton University, CEPS Working Paper No. 91, April 2003

[5] Wikipedia on MACD. http://en.wikipedia.org/wiki/MACD

[6] Wikipedia on HMM http://en.wikipedia.org/wiki/Hidden_Markov_model

[7] R. Hassan and B. Nath, *Stock Market Forecasting Using Hidden Markov Model: A New Approach*, IEEE, 2005

[8] Y. Zhang, *Prediction of Financial Time Series with Hidden Markov Models*, Simon Fraser University, May 2004

[9] http://en.wikipedia.org/wiki/Baum-Welch_algorithm

[10] S. Borman, *The Expectation Maximization Algorithm: A Short Tutorial*, http://www.isi.edu/natural-language/teaching/cs562/2009/readings/B06.pdf, 2004-2009

[11] M. Jordan, *Graphical Models*, University of California, Berkeley, Ch12

[12] R. Dugad and U. B. Desai, *A Tutorial on Hidden Markov Models*, Technical Report No: SPANN-96.1, May 1996

[13] D. Klein, Lecture 21, http://inst.eecs.berkeley.edu/~cs188/fa09/slides/FA09%20cs188%20lecture%2021%20--%20speech%20%282PP%29.pdf, Fall 2009

[14] K. Kim, *Financial Time Series Forecasting Using Support Vector Machines*, Elsevier, March 2003

[15] C. Hsu, C. Chang, C. Lin, *A Practical Guide to Support Vector Classification*, National Taiwan University, 2003

[16] R. Cooley, *Classification of News Stories Using Support Vector Machines*, University of Minnesota, April 1999

[17] Z. Hua, Y. Wang, X. Xu, B. Zhang, L. Liang, *Predicting Corporate Financial Distress Based on Integration of Support Vector Machine and Logistic Regression*, Expert Systems with Applications, 2007

[18] Jahmm, http://code.google.com/p/jahmm/

[19] JavaML, http://java-ml.sourceforge.net/

[20] LibSVM, http://java-ml.sourceforge.net/

[21] F. Allen and D. Gale, *Stock-Price Manipulation*, The Review of Financial Studies, Volume 5 No. 3, 1992

[22] J. Mei, G. Wu, and C. Zhou, *Behavior Based Manipulation: Theory and Prosecution Evidence*, JEL Classifications: G12, G18, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=457880, April 2004

[23] R. K. Aggarwal and G. Wu, *Stock Market Manipulations*, Journal of Business, 2006

[24] Jegadeesh, Narasimhan and S. Titman, *Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency*, The Journal of Finance, 1993

[25] R. J. Shiller, *From Efficient Market Theory to Behavioral Finance*, Cowles Foundation Discussion Paper, 2002