Predicting Stock Price from Financial Message Boards with a Mixture of Experts Framework

Alexander Y Liu Bin Gu Prabhudev Konana Joydeep Ghosh

aliu [at] ece.utexas.edu
bin.gu [at] mccombs.utexas.edu
prabhudev.konana [at] mccombs.utexas.edu
ghosh [at] ece.utexas.edu

IDEAL-2006-07*

Intelligent Data Exploration & Analysis Laboratory

(Web: http://www.ideal.ece.utexas.edu/)

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712
U.S.A.

September 29, 2006

Abstract

Many online financial message boards exist where users can discuss the latest information about their favorite stocks. However, the sheer number of postings and the high level of noise in these postings present a significant challenge to message board users seeking to extract recommendations on whether to buy or sell a stock. Earlier attempts to extract sentiment and information from these boards and relate it to stock markets have been largely unsuccessful. This paper presents a new approach which focuses on identifying the most historically accurate posters in message boards. Specifically, we use a mixture of experts framework to identify these posters and analyze their sentiments in a completely automated fashion. We find that this approach not only extracts information from message boards, but that implementable strategies based on this extracted information can automatically be devised to achieve statistically significant returns even after adjusting for both market effects and commission rates.

1 Introduction

A growing source of potential information is online message boards. On these message boards, people with questions can post questions and comments which are potentially answered and addressed by domain experts. Many message boards do not require registration or over-zealous authentication, allowing users to post publically, anonymously, and easily, creating a realm where anyone with access to a computer can learn at the feet of the masters. Unfortunately, the ease with which message boards can be accessed is a potential weakness as well as a potential strength. It is often difficult for a casual user to determine who is a domain expert, who is a reputable poster, and who might be someone with ulterior motives, such as a person trying to surreptitiously sell a product in which they have more than simple, objective interests. The difficulty becomes even higher for users who do not know much about the domain, users who are, ironically, the very same who would benefit most from expert guidance.

Financial message boards on which users discuss stock recommendations are known to suffer from these problems. On financial message boards, posters may have a wide variety of backgrounds and knowledge about the company being discussed or about the market in general. Moreover, the inherent financial incentives make this domain particularly attractive to those with ulterior motives, making it even more important to determine which posters are truthful. Unfortunately, users may have difficulty distinguishing between posts containing fact, rumors, guesses, and outright lies. It would therefore be useful to have ways of determining which users tend to create highly authoritative posts and which users are historically inaccurate.

In this paper, we empirically address a variety of problems related to financial message boards and the stock market. The first is whether or not messages posted on financial message boards contain meaningful information with respect to stock market movements. The second is whether one can identify which users are historically accurate and which users are rarely correct in their postings. Finally, we study whether or not one can use the information posted to financial message boards and create implementable, real-time stock trading strategies that are profitable even after adjusting for market effects and commission rates. More specifically, we find that the sentiment tags posted by users on message boards indicating whether or not they feel a stock will go up or go down in price can be used to accurately predict stock market movements. Moreover, a mixture of experts framework can be applied to learn and create profitable, real-time strategies. We test our approach on actual message board data obtained from Yahoo! Finance.

2 Related Work

There have been many other works that have studied financial message boards. For example, [Wys99] examined user behavior on financial message boards. Other studies have attempted to extract information from financial message boards or other sources and to correlate this information with stock market activity. In particular, a majority of these studies have attempted to use text mining approaches. For example, in [TS00], the authors try a maximum entropy text classifier trained on message board data and a genetic algorithm trained on attributes such as trading volume and number of messages posted in order to predict whether a stock will move up or down the next day. [AF04] also used text classification to extract sentiments and then combined these sentiments to create an aggregate prediction of bullishness for the entire board.

However, this approach assumes that all posters contribute equally to a board and that all posters are equally accurate. Moreover, it does not account for users who post multiple times per day. We will discuss [AF04] in more detail in section 4.5. [DC01] also studied a number of text classifiers in order to automatically group financial message board posts based on sentiment (e.g., buy or sell), although one should note that the text classification algorithm used in [AF04] is much more advanced and much more indicative of what is actually used in practice in the information retrieval and text mining communities. Finally, in an interesting approach that looked outside of message boards, [LSL+00] and [FYL05] attempted to correlate news stories with stock market activity.

While these studies focus on extracting sentiment from postings, they ignore the fact that not all postings are equal. Predictions of a few informed experts may well overweight predictions from thousands of uninformed traders. With this insight, we take a different approach. Instead of focusing on extracting sentiments from postings, we focus on identifying top experts and give more weight to their predictions. In addition, in contrast to many past studies that have used text mining techniques to extract sentiment, we use posts that are specifically tagged for sentiment by the user. This approach is similar to work in [Mis06] where the authors mine blog sentiments by looking at the posted sentiment. However, the way in which these sentiments are analyzed is very different from our current study. One major advantage of using user-tagged sentiment is that a large percentage of message board posts are not about the stock itself (e.g., discussion of unrelated current events). In addition, one drawback of using statistical based text mining approaches (e.g., as in [TS00], [DC01], [AF04], etc.) is that it is often difficult to distinguish users discussing a "good movie" with users discussing a "good time to buy."

In summary, this work differentiates itself from previous work in the following way: it is one of the first studies of applying a mixture of experts framework to financial message board data in order to predict stock market movements. Previous works on analyzing message boards for a similar purpose have used different approaches with little success. While a mixture of experts framework fits the problem well from a machine learning perspective, we are unaware of any studies that have empirically tested whether a mixture of experts solution works well in practice. Moreover, this is one of the first studies to examine whether messages specifically tagged by the users to express a certain sentiment actually contain useful information.

3 Dataset

Our dataset consists of message posts crawled from Yahoo! Finance from May 14, 2005 to April 29, 2006. Note that the format of the Yahoo! Finance message boards have changed since we created our dataset, so the descriptoin we provide below may not match the current format. When we crawled the data, the Yahoo! Finance stock message boards were divided such that a single message board corresponded to a single stock symbol. A user posted messages to the message board using a screen name. The message itself consisted of the user name, a subject, a message, the time and date the post was created, and an optional sentiment tag. The optional sentiment tag could be used to explicitly tag whether the user thought the current stock was a "strong buy," "buy," "hold," "sell," or "strong sell." Finally, a particular post could be recommended by other users; the number of recommendations was readily available from the message board as well.

As mentioned, Yahoo! Finance divides its financial message boards by stock symbol. In

Table 1: Stocks in our dataset

aa	cnet	mdrx
aapl	dgin	mgm
abtl	dis	mnst
acn	driv	mo
adbe	ek	mot
agil	elnk	mrk
amd	et	mro
amtd	fdg	msft
amzn	gm	nbr
aqnt	goog	ostk
arba	hlth	pcar
axp	hon	pcln
ba	ibm	pfe
bbby	insp	pg
beas	intc	qcom
bge	ip	sape
bmc	iwov	sunw
brcm	jnj	tibx
c	jpm	untd
cat	jupm	uvn
cce	ko	vert
chkp	lll	vign
ckfr	lu	xom
cmgi	mcd	

our experiments, we do not consider any interactions between stock message boards. Instead, we treat each message board for a single stock symbol as a single, autonomous dataset that is independent of all other message boards, resulting in a total of 71 datasets for 71 stock symbols. Table 3 lists the 71 stocks in our dataset. These stocks were chosen to cover a wide variety of company types and message board characteristics (e.g., number of users and number of tagged posts per day) although there is a bias towards technology related companies.

4 A Mixture of Experts Framework

In our current problem setting, we have a number of posters creating messages each day on a particular financial message board. Each poster has the option of including a prediction via the sentiment tag of whether one should buy or sell the stock in order to make the most profit. Abstractly, one can view a particular day as a time period t and all predictions/sentiment tags on a particular time period t as the data we are trying to use to predict stock market returns. Each time period t brings a new set of predictions. There exist a set of solutions in data mining and machine learning to solve this type of problem where there is a stream of data from which we are trying to learn and predict a target function (in this case, our stream of data consists of message board posts on day t and our target function corresponds to stock market returns).

Collectively, these data mining algorithms can be grouped together as "online algorithms" [1].

One such online learning algorithm that is particularly suited to solving this problem is a mixture of experts framework. Informally, a mixture of experts framework observes the predictions of a group of experts and combines the individual predictions into a single prediction; in addition, it automatically learns which experts are typically most accurate. In a general mixture of experts framework, there is a set of experts $\mathcal{E}$ where the ith expert is indicated by the notation e_i. The goal of the mixture of experts framework is to predict a time series y, where the value of y at time t is given by $y(t)$. This prediction is indicated by the notation $\hat{y}(t)$. $\hat{y}(t)$ is calculated based on the prediction of each expert at time t (denoted as $e_i(t)$) and the weight of each expert (denoted as $w_i(t)$). The weight $w_i(t)$ of an expert reflects how accurate the expert has been in the past (i.e., all times before the current value of t). Higher value of $w_i(t)$ indicate that the ith expert is more accurate and trustworthy than an expert with a lower value of $w_i(t)$.

Note that the term "expert" is a bit of an unfortunate misnomer that has become standard in the data mining and machine learning literature. An "expert" can be incorrect 100% of the time and would still be called an "expert." Thus, whenever we use the term "expert" in this paper, it is in the same vein as the data mining and machine learning literature. That is, an "expert" is simply someone that makes a prediction, not necessarily someone that makes an accurate prediction.

A mixture of experts framework works by repeating three basic steps: getting individual predictions from the experts, aggregating the predictions of all experts, and updating the weights of the experts based on their latest prediction. We will now talk about each of these steps in more detail.

4.1 Making predictions

The first step in a mixture of experts framework is to obtain the individual predictions of each expert for the current time step. That is, we need to determine $e_i(t)$ for all i. In our experiments, the predictions of each expert is based on the sentiment tags created by the poster. In our dataset, posters can state that their current position on a stock is "strong buy," "buy," "hold," "sell," or "strong sell." Let $M_i^c(t)$ be the number of messages of type c that expert i creates at time t. For example, $M_i^{sell}(t) = 2$ means that the ith poster created two messages on day t that indicated they wanted to "sell" the stock. Let $M_i^{total}(t)$ be the total number of posts with sentiment created by a poster. Then the prediction $e_i(t)$ is calculated as follows.

If $M_i^{total}(t) > 0$, then:

$$e_i(t) = \frac{M_i^{strongbuy} + M_i^{buy} - M_i^{sell} - M_i^{strongsell}}{M_i^{total}} \tag{1}$$

Otherwise, the ith expert has not made a prediction at time t. Note that the prediction $e_i(t)$ has a number of amenable properties. First, $e_i(t)$ can be viewed as a score between -1 and 1 corresponding to poster i's opinion of the stock, where a score of +1 corresponds to a bullish view about the stock and a score of -1 corresponds to feelings that the stock price will decrease (cf. with bullishness scores described in [AF04] and in section 4.5). Secondly, if a poster

[1] Note that the term "online" has nothing to do with the Internet, although certainly online algorithms can be used in many Internet related problems.

makes a single prediction, then $e_i(t)$ is 1 if the poster predicts "buy" or "strong buy", 0 if the poster predicts "hold", and -1 if the poster predicts "sell" or "strong sell". If a poster makes multiple predictions at time t, equation 1 collapses these multiple posts into a single prediction $e_i(t)$. Thus, if a poster is trying to "pump" a stock by flooding a board with "strong buy" messages, their attempts at flooding and biasing the board are neutralized by creating a single $e_i(t)$. Similarly, if a poster has multiple conflicting sentiments at time t, then $e_i(t)$ can be seen as an average of these sentiments. For example, if a poster creates two messages indicating "buy" and one message indicating "sell," then $e_i(t) = 1/3$.

Note, however, that if $M_i^{total} = 0$ for expert i at time t, then equation 1 does not tell us what to do concerning expert i. In machine learning, there is a specialized version of the mixture of experts framework called the "sleeping experts" framework. In the sleeping experts framework, not all experts make predictions at all times t. Our experiment fits very naturally into the sleeping experts framework. If the ith expert makes no prediction (i.e., if $M_i^{total}(t) = 0$), then we say that the ith expert is asleep. Otherwise, if the ith expert makes a prediction $e_i(t)$, then we say the ith expert is awake at time t and is a member of the set $\mathcal{E}_{awake}(t)$.

4.2 Aggregating predictions

In a sleeping experts framework, only the experts in $\mathcal{E}_{awake}(t)$ are used to create the aggregate prediction $\hat{y}(t)$. The standard definition of $\hat{y}(t)$ is simply the weighted average of the predictions $e_i(t)$ for all experts in $\mathcal{E}_{awake}(t)$, where the weights for each $e_i(t)$ are given by the accuracy weights $w_i(t)$ (discussed in 4.3). For now, it is sufficient to note that higher values of $w_i(t)$ roughly correspond to experts that have historically been more correct. In particular, experts with a weight of zero have never been correct in the past.

However, from a practical standpoint, we may want to only listen to certain types of experts when creating $\hat{y}(t)$. For example, we might decide that there is no reason that we should listen to experts with a low weight (i.e., experts who have rarely been correct in the past). We can define a set $\mathcal{E}_{rel}(t)$ that consists of the only experts whose predictions are relevant to predicting $\hat{y}(t)$. Then, given $\mathcal{E}_{rel}(t)$, $\hat{y}(t)$ is defined as:

$$\hat{y}(t) = \frac{\sum_{e_i(t) \in \mathcal{E}_{rel}(t)} w_i(t) \times e_i(t)}{\sum_{e_i(t) \in \mathcal{E}_{rel}(t)} w_i(t)} \tag{2}$$

In our experiments, we compare a number of definitions of $\mathcal{E}_{rel}$. Each of these definitions can be considered one type of strategy for using message board information to create trading strategies. The definitions of $\mathcal{E}_{rel}$ used in this paper are as follows.

1. All awake experts: $\mathcal{E}_{rel}(t) = \mathcal{E}_{awake}(t)$
2. Top k experts: $\mathcal{E}_{rel}(t)$ consists of the k experts in $\mathcal{E}_{awake}(t)$ with the highest non-zero weights. If the number of non-zero weighted experts in $\mathcal{E}_{awake}(t)$ is less than k, then $\mathcal{E}_{rel}(t) = \mathcal{E}_{awake}(t)$.
3. Worst k experts: $\mathcal{E}_{rel}(t)$ consists of the k experts in $\mathcal{E}_{awake}(t)$ with the lowest non-zero weights. If the number of non-zero weighted experts in $\mathcal{E}_{awake}(t)$ is less than k, then $\mathcal{E}_{rel}(t) = \mathcal{E}_{awake}(t)$.
4. Thresholded experts: $\mathcal{E}_{rel}(t) = \{e_i | w_i > w_\theta, e_i \in \mathcal{E}_{awake}(t)\}$

In our experiments, we let $k = 5$ and $w_\theta = .9$. If $\sum_{e_i(t) \in \mathcal{E}_{rel}(t)} w_i(t) = 0$, then $\hat{y}(t) = 0$.

We define $\mathcal{E}_{rel}(t)$ in four different ways for the sake of empirical comparison. Each of the above four strategies has a certain bias that makes it interesting for empirical study. The "all experts" strategy is simply the sleeping experts strategy. The "top experts" and "thresholded experts" strategies both have some filtering in place such that $\mathcal{E}_{rel}(t)$ consists of only those experts with high weights. For the "top experts" strategy, the set $\mathcal{E}_{rel}(t)$ consists of the best possible experts at each time t, while for the "thresholded experts" strategy, the calculation of $\hat{y}(t)$ depends only on those experts that have been most accurate in the past. One drawback of the sleeping experts framework is that it is possible that the experts in $\mathcal{E}_{awake}(t)$ all have extremely low weights. This problem seems to be exacerbated when the experts in question are human predictors instead of computer algorithms. In particular, humans are known to err due to various reasons such as overgeneralizing based on anecdotal evidence, a herd mentality, or lying outright. Thus, many of the experts posting each day have a low accuracy weight, and the "top experts" and "thresholded experts" strategies are designed to combat this problem.

Finally, the "worst experts" strategy is simply present to study the empirical effect of listening to the lowest weighted (but non-zero) experts at time t.

Note that we also create a baseline strategy called "unweighted" experts. It can be defined as an average of all expert predictions in $\mathcal{E}_{awake}$. That is, the baseline strategy does not take into account accuracy weights $w_i(t)$.

4.3 Updating weights

Once the aggregate prediction $\hat{y}(t)$ is made, one receives the true value of the target function $y(t)$. Based on the true value $y(t)$, the accuracy weights of all experts are updated in the hope that $\hat{y}$ will be close to y in the future. Weight updates are the third and final step in the three step, iterative mixture of experts framework.

We used a variety of different weight updates (e.g., multiplicative updates) but found an additive update rule to empirically perform the best. For the sake of brevity, we will only describe additive weight updates. Additive weight updates have been studied extensively in a variety of communities and can be traced at least as far back to [Ros58]. The additive update rule we use is as follows:

$$w_i(t+1) = \alpha \times w_i + (1 - \alpha) \times (e_i(t) == y(t)) \tag{3}$$

where α is a parameter known as the "learning rate" and $0 \leq \alpha \leq 1$. α controls and limits how quickly w_i changes from time t to $t+1$. In the extreme cases, if $\alpha = 0$, then $w_i(t+1)$ is dependent only on whether $e_i(t)$ was correct or not. If $\alpha = 1$, then $w_i(t+1)$ is independent of whether or not $e_i(t)$ was correct. Typically, α is set to some moderate value between 0 and 1 such that $w_i(t+1)$ changes gradually over time. Equation 3 also limits how quickly $w_i(t+1)$ increases such that an expert that is merely a lucky guesser does not receive too high a weight. Note that when using the additive update rule, $0 \leq w_i \leq 1$.

In our experiments, we had to change a few of the above steps in order for our framework to be implementable in real-time. In our experiments, $y(t)$ is the sign of the return for buying a stock at time t and selling it at time $t + \omega$. Thus, $y(t)$ will not be known until time $t + \omega$. In order for our weight updates to be done in real time, we can only use data up to time t. Thus, in our experiments, we use the following equation:

$$w_i(t+1) = \alpha \times w_i + (1-\alpha) \times (e_i(t-\omega) == y(t-\omega)) \quad (4)$$

Note that $w_i(t)$ therefore lags its "true value" by ω time periods. Interestingly, this does not seem to cause a problem empirically as we shall see in section 5.

4.4 An Implementable Strategy

Given the definitions of our experts and target function given above, we can describe an implementable trading strategy.

We first gather all predictions for a given time t (i.e., all posts with predictions made 24 hours before the current closing time). For a given definition of $\mathcal{E}_{rel}$, we calculate $\hat{y}(t)$ as described in 2. We buy the stock at time t if $\hat{y}(t) > 0$ or short the stock if $\hat{y}(t) < 0$. We then sell/buy back the stock in ω days. If $\hat{y}(t) = 0$, then we do nothing (i.e., we hold our current position). Thus, the return on our investment by following this strategy at time t would be equal to $sign(\hat{y}(t)) \times y(t)$, where $y(t)$ is the sign of the rate of return.

To evaluate our framework empirically, we simply use the average rate of return one would receive by following the implementable trading strategy described above averaged over all time steps t in our experiment. The rate of return at time t is also adjusted for market effects. We use the following equation to calculate the adjusted rate of return:

$$r_{adj}(t,\omega) = -1 + \prod_{k=t-\omega}^{t} 1 + (s(k)/s(k-1) - 1) - \beta * (s_{S\&P}(k)/s_{S\&P}(k-1) - 1) \quad (5)$$

where $s(k)$ is the stock price at time k, $s_{S\&P}(k)$ is the price of the S&P 500 index at time k, and β is the Beta-value of a stock, a measure of how much the stock fluctuates with regards to the market. We calculate β via linear regression, where our dependent variable consists of the stock returns from April 30, 2004 to April 30, 2005 and our independent variable consists of the returns from the S&P 500 index over the same time period. β is the coefficient for the independent variable in the learned linear regression model. Note that the time period we used to calculate β does not overlap with the time period for which we ran experiments; that is, the data used to learn β does not overlap with the data described in section 3. Intuitively, $r_{adj}(t,\omega)$ represents the percentage of money one would gain from investing in a stock at time $t-\omega$ and selling the stock at time t after adjusting for market fluctuations. For the sake of clarity, $y(t) = sign(r_{adj}(t,\omega)$.

Empirically, we set $e_i(0) = 0$ for all experts i. For a given definition of $\mathcal{E}_{rel}$, we vary ω between 1 and 50 trading days and find the adjusted return averaged over all of the datasets described in section 3.

4.5 Connection with Antweiler's Approach

Interestingly, the approach for predicting bullishness from message boards described in [AF04] has some connections with the approach described above. In particular, these connections form an interesting bridge between techniques developed independently in the realm of machine learning and in the realm of finance.

Three measures for predicting bullishness from a message board were given in [AF04]. Let $M_t^c = \sum_c w_t^c * m_t^c$, where c represents either all buy messages at time t or all sell messages at

time t, w_t^c is the weight of message m_t^c, and m_t^c represents a single buy or sell message at time t. In [AF04], w_t^c was set to 1 for all t and c. Thus, M_t^{buy} represents the number of messages that stipulate buying a stock at time t and M_t^{sell} is the number of messages that recommend selling the stock at time t.

The three measures of bullishness B_t, B_t^*, and B_t^{**} can now be defined as:

$$B_t = \frac{M_t^{buy} - M_t^{sell}}{M_t^{buy} + M_t^{sell}} \tag{6}$$

$$B_t^* = \ln \frac{1 + M_t^{buy}}{1 + M_t^{sell}} \tag{7}$$

$$B_t^{**} = M_t^{buy} - M_t^{sell} \tag{8}$$

In [AF04], it was noted that all three measures of bullishness gave approximately the same results and B_t^* was used for analysis. Note, however, the similarity between the calculation of B_t and equations 1 and 2. All three of these equations can be considered weighted averages. In particular, in equations 6 and 1, this can be done if buy messages are weighted with a positive 1, sell messages are weighted with a -1, and hold messages are weighted with a 0.

Assuming posters create a single post per day, if we were to ignore messages recommending to hold a stock and if we were to let $w_i(t) = 1 \forall i, t$, then the aggregate prediction found in equation 2 is exactly the same as B_t. Moreover, we could use the same weights in the mixture of experts framework for calculated M_t^{buy} and M_t^{sell} in equation 6. Thus, the bullishness score B_t described in [AF04] can be considered a specific case of the more general mixture of experts approach when all weights are set to 1. In addition, the mixture of experts approach allows an empirically well-studied and theoretically principled method of assigning weights to individual messages to increase the accuracy of predictions (i.e., the bullishness score).

While [AF04] preferred B_t^* over B_t, we prefer the weighted average corresponding to B_t for several reasons. The first is that B_t is bounded by -1 and 1, whereas the magnitude of B_t^* has no bound. This could be used in future implementable strategies since one knows the confidence in the current strategy on a known, finite scale. Secondly, B_t corresponds to the weighted average that has been used extensively in the data mining community with regards to mixture of experts. Finally, the sign of B_t^* is always equal to the sign of B_t. In our implementable strategies, only the sign of the prediction matters, and so the choice of bullishness measure becomes one of personal preference.

In section 5 we will show empirically that Antweiler's methods do indeed correspond with the standard mixture of experts approach. However, there are some important differences between the study in [AF04] and the current study. Thus, in order to directly compare Antweiler's approach with our own, we have made the following changes. The first is that Antweiler used text classification to determine whether a message was advocating to buy, hold, or sell the stock [2]. Instead of using text classification methods, we are using the same sentiment tags as our methods in order to determine whether a poster is recommending to buy/sell the current

[2]Interestingly, the highest percentage of messages advocated to "hold" the stock in Antweiler's study. There is also no category for off-topic posts. Given these two facts combined with the types of priors we observe in our datasets, we surmise that off-topic posts may also have been grouped into the "hold" category, a reasonable grouping since hold messages are ignored in [AF04]

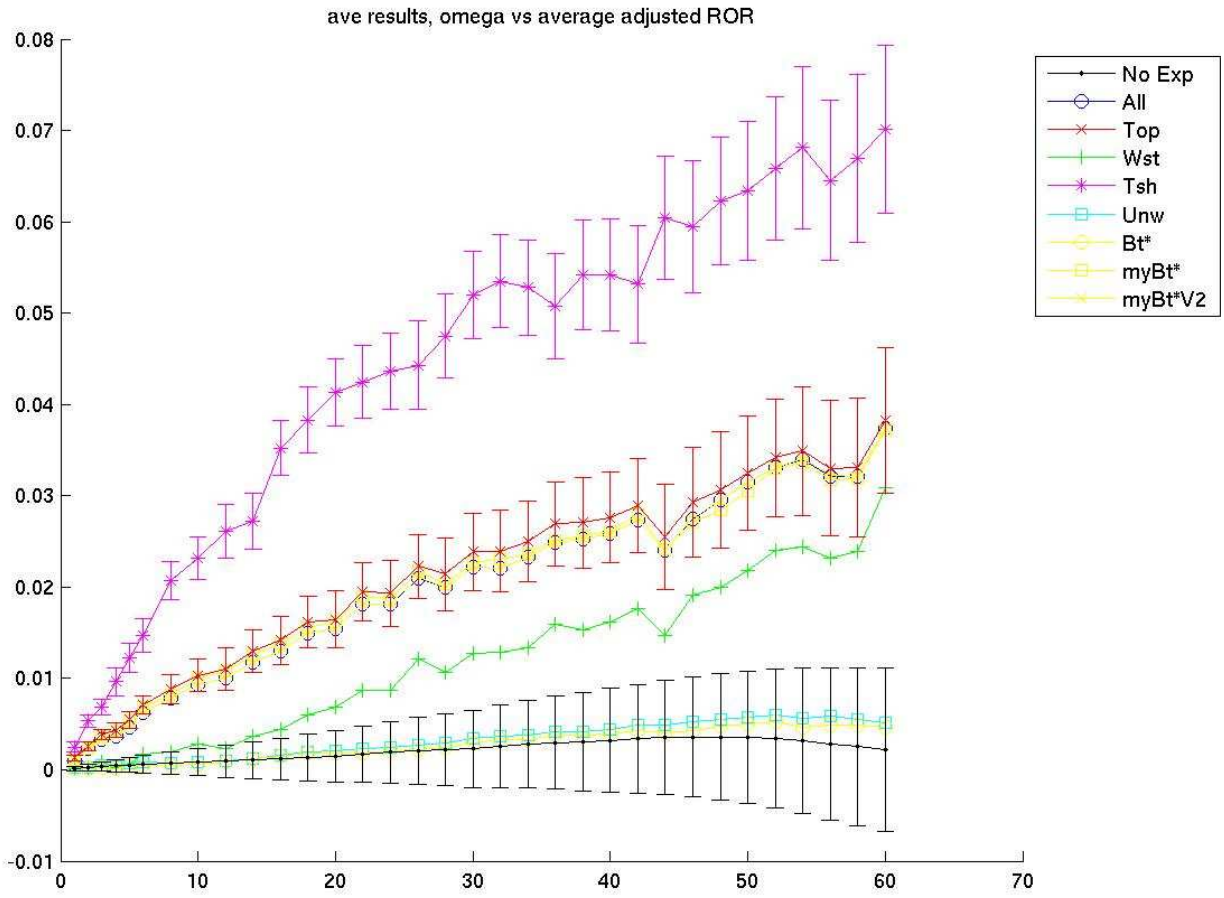

Figure 1: Adjusted Rate of Return; $\alpha = .5$

stock. In [AF04], it was assumed that all messages were relevant to the stock (i.e., all messages either recommended to buy, sell, or hold the stock). Here, the only relevant messages are those that recommend to either buy or sell the stock. Note that all three bullishness scores described in [AF04] ignore hold messages, so this is a reasonable and straightforward extension. Finally, the most important difference is the ability to find message weights in an automated fashion, something that was impossible in [AF04].

5 Results

Figure 5 contains the main results of our experiment. The horizontal axis corresponds to a specific time window ω, while the vertical axis corresponds to the average adjusted rate of return. Each curve describes the behavior of a specific strategy, and each point on the curve corresponds to the adjusted rate of return averaged over all datasets/stock symbols for a particular value of ω. The results in the figure were obtained by setting $\alpha = .5$. We also tried

various other values for α (0.1, 0.3, 0.7, 0.9) and found that our results were consistent for all experimented values. Thus, our strategy is robust to the choice of α. For the sake of brevity, we include only results for $\alpha = .5$.

From Figure 5, we see that the threshold strategy outperforms all other strategies. Intuitively, one can think of the threshold strategy as a way of keeping track of posters who are the most accurate in predicting stock market movements in the past and listening only to the most accurate posters. It is therefore not surprising but quite satisfying to see the threshold strategy performing so well when compared to the other strategies.

Interestingly, we see that the "all experts" strategy and "top experts" strategy perform almost identically. One would expect a priori that the "top experts" strategy would outperform the "all experts" strategy. However, when delving into the data a bit further, we see that on many boards, there are few predictions made per day. An even smaller number of these predictions are made by posters with non-zero weights. Thus, the set that makes up $\mathcal{E}_{rel}$ is often the same for the "all experts" and "top experts" strategies. Secondly, a drawback to the definition of $\mathcal{E}_{rel}$ for the "top experts" strategy is that if all the experts $e_i(t)$ awake at time t have a low weight $w_i(t)$, the "top experts" strategy still tries to make a prediction. That is, the "top experts" strategy looks at only the ordering of $w_i(t)$, whereas the "threshold" strategy looks at the magnitude of $w_i(t)$ and uses a prediction only if the magnitude of $w_i(t)$ is higher than some absolute threshold.

Note that the strategy using the weighted version of B_t^* performs almost identically with the "all experts" strategy. Given the discussion in section 4.5, this is exactly as expected. The small difference is due to the fact that the "all experts" strategy takes into account messages that predict "hold" whereas the weighted B_t^* strategy does not. This also supports the observation that taking "hold" messages into account does not affect results very much.

Finally, let us look at the performance of the worst strategies. Interestingly, the "worst experts" strategy outperforms the "buy and hold" strategy, the "unweighted" strategy, and the unweighted B_t^* strategy. Note, however, that, as described previously, the "worst experts" strategy only listens to experts with non-zero weight w_i whereas the "unweighted" and unweighted B_t^* strategies do listen to experts with non-zero weight. In addition, it is theoretically possible that the "worst experts" strategy listens to posters with high weights if the only posters posting at time t all have high weights w_i. In practice, however, we should note that most posters do not have high weights, and this scenario would definitely be the exception rather than the norm.

Note that the "unweighted" strategy and unweighted B_t^* strategy perform almost identically for the same reason given as when discussing the "all experts" and weighted B_t^* strategies. The fact that the unweighted strategies barely outperform the "buy and hold" strategy matches past work where it was found that there was some information on financial message boards, but not enough to significantly outperform a simple "buy and hold" strategy.

Thus, we can make two important conclusions from our empirical results. The first is that there is significant information in the sentiment tags that can be used to create real-time, profitable, implementable strategies. The second is that some method of keeping track of the past predictive performance of users is needed in order to fully use the information found in the sentiment tags. Here, we see that using expert weights significantly outperforms using sentiment tags without using weights (e.g., the unweighted strategy or [AF04]).

We also reran the above experiments when there was a 1% commission rate on all trans-

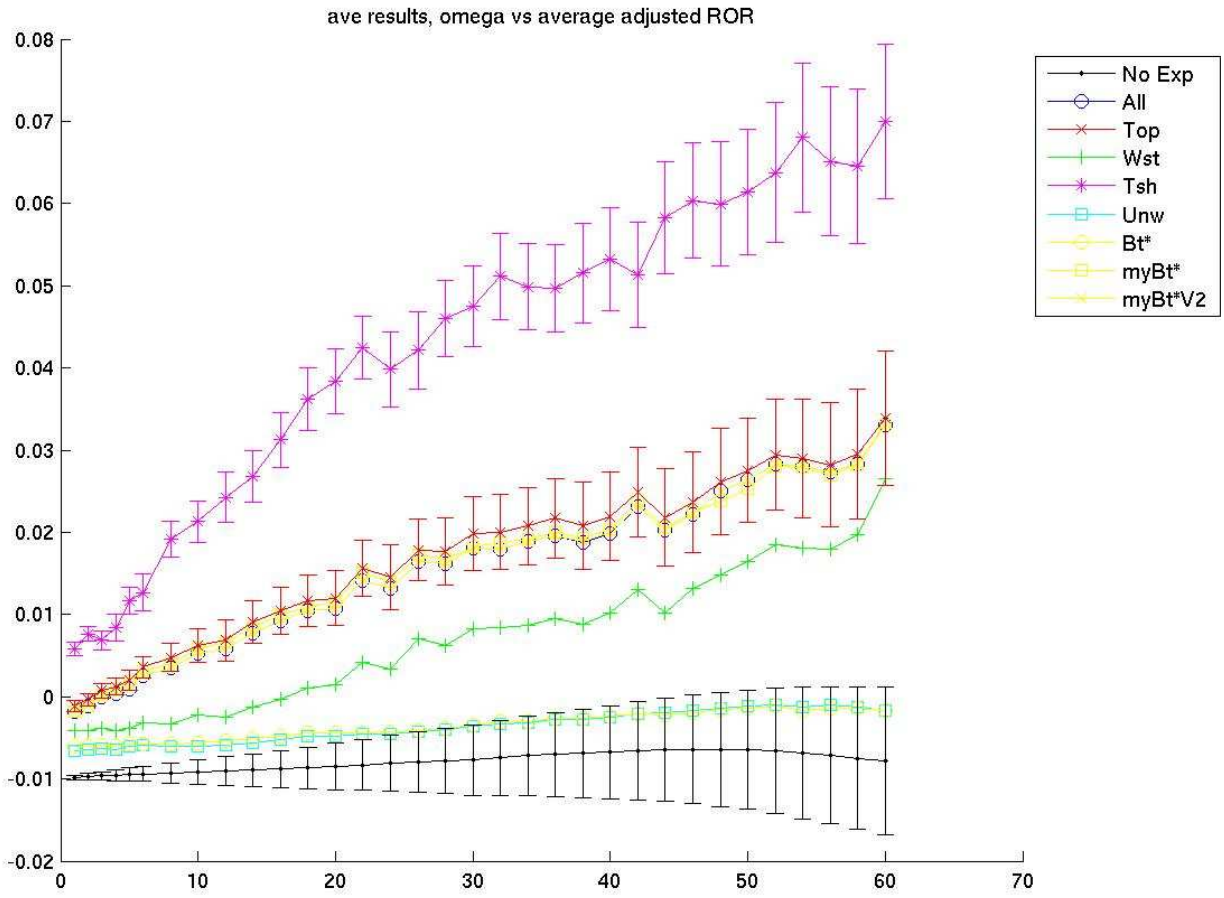

Figure 2: Adjusted Rate of Return with Commission Rate of 1%; $\alpha = .5$

actions. Figure 5 contains the results. Note that the general trends are the same as above. However, when we include commission costs, it becomes even more important to take into account past performance of posters via a mixture of experts framework. On average, only the strategies that use mixture of experts weights are able to produce a positive rate of return.

6 Future Work

Our current study can be expanded in a number of different ways based on approaches from different domains. Below we describe a number of them.

6.1 Improvements to current approach

In terms of the algorithms used in our experiments, there are a number of ways that they could be improved. For example, posters need to be accurate quite frequently in order to increase

their accuracy weight, and so posters who rarely post, regardless of accuracy, will never have a high accuracy weight. Thus, assigning weights to rarely predicting but highly accurate posters needs to be addressed in future work.

In addition, we currently rely on message board posters to accurately label their own posts with their sentiment. We plan to use various techniques from text mining to improve our current approach. For example, future work needs to take into account that some of the unlabeled posts will express sentiment in addition to the possibility that some of the labeled posts may be labeled incorrectly. While work on classifying message board posts according to sentiment have been used in past studies on financial message boards (e.g., Das 2001), many more sophisticated sentiment classification techniques have been proposed in other domains.

In addition to mining messages for sentiment, no current approach we are aware of can tell whether a poster indicates they believe their position should be held for a short term or long term period. An even more ambitious approach would be able to predict exactly how long a poster feels they should hold their position. Possibilities for solving this problem include modifications to the mixture of experts framework to account for predicted delays, text mining or information retrieval type approaches, or looking for the next post by the same user. All of these approaches have a number of interesting problems to overcome. For example, any natural language approach needs to deal with ambiguities in text such as hyperbole (e.g., a poster who unequivocally states that the current stock will keep rising forever). Thus, how to choose a time window in a principled fashion is an interesting area of future work for which there may be no simple answers. Moreover, although we acknowledge the trade off between small and large windows, we have yet to formally study the empirical effects of this tradeoff.

We could also improve our current approach by including other sources of information. One large pool of potential data is outside sources of information such as news stories or the time of year. For example, it is known that the number of posts increases significantly around days when earnings reports are released [Wys99]. We observe this phenomenon in our own datasets as well, in addition to the fact that many posters only post during these time periods. There are also additional sources of information within the message board itself such as the number of times a particular post has been recommended by other readers. However, we have not been able to find an effective way to capitalize on this information in initial experiments. One particularly interesting source of data is the interactions between different message boards. For example, we have observed that the same two posters often interact on different message boards, particularly for stocks in related areas (e.g., Intel and AMD). These interactions and others provide a rich dataset on which various social network experiments can be conducted. For example, one interesting question that we hope to study is whether or not people tend to listen to posters that accurately predict stock movements or whether most posters tend to listen to other users with the same, presupposed viewpoints. In addition, information about a poster on one message board could be used to leverage analysis of that same poster on other message boards. If a particular user is known to do nothing but provide useless information on one board, that information might be useful when that same user starts posting on another message board.

6.2 The psychology of message boards

Examining the sentiment of users on financial message boards raises some additional questions not often seen in other domains. For example, it is known that posters may lie for various

reasons, such as trying to "pump and dump" a stock (i.e., creating the illusion the stock is worth more than it is before selling their stock at inflated prices) or simply as a prank. In addition, some posters spread and speculate on rumors which they truly believe to be accurate, confirmed information. Being able to distinguish between a poster who is simply wrong but believes they are posting accurate information and a poster who lies for malicious reasons is an interesting open problem. Further differentiating the above posters from a poster who is actually correct but "goes against the grain" of conventional wisdom is another interesting question.

From a psychological standpoint, it would be interesting to determine whether message board posters are more prone to agreeing with those whose viewpoints tend to correspond with their own or whether posters tend to listen to the opinions of posters who have proven to be correct in the past. From our own observations, our conjecture is that many posters tend to listen only to those opinions most closely aligned with their predisposed conclusions. In addition, it would be psychologically interesting to examine the type of language that is used, particularly the linguistic patterns of successful pumpers and dumpers.

Other studies such as [GKRC06] examines the fact that there is a significant amount of "noise" posts on message boards (e.g., posters flaming each other or chatting about the weather). Given such a hostile environment, there is the open question of why users with "insider information" would blatantly post their tips on a public message board, and, if so, what economic advantage they would have in doing so. Another observation is that, for a message board that one assumes to be composed of mostly adults, some of whom claim to make millions of dollars per year, the overall tone of some of these posters is surprisingly childish. The amount of gratuitous swearing, name-calling, speculation, mocking ridicule, and arguing begs the question of whether boards where posters call each other idiots is a fertile breeding ground for posters looking for a place to share actual information.

6.3 New problems to explore

One interesting open problem is how one can generalize our approach to message boards and message board type communities in other domains. In stock message boards, there is an external indicator of whether posters are accurate or not. In other areas, such as product reviews, there is no objective, external indicator. However, some type of user filtering would still be useful in order to determine, for example, whether one is a domain expert or whether one is being paid to market or endorse a particular product.

Finally, we have not looked into whether one can create an automated portfolio selection algorithm based on information gleaned from financial message boards. In portfolio selection, one has a finite amount of money to invest and a pool of candidate stocks to invest in. Using information from message boards, one could envision an automated strategy that learns which boards are most accurate and when boards start to lose their predictive power. Based on this information, the strategy would allocate the current funds appropriately based on what the posters on each of the boards were currently predicting.

7 Conclusion

In conclusion, we have studied the issue of whether there is useful information on financial message boards pertinent to stock market movements and whether that information can be

used automatically. We have found that not only is there useful information, but that this information can be used to make accurate predictions about the return on investment one can expect after adjusting for market effects. Future work needs to be done in order to refine this approach. However, even the basic approaches outlined in this paper appear to work reasonably on real data and can be implemented as realizable strategies.

Acknowledgments: Thanks to NSF for supporting this work.

References

[AF04] W. Antweiler and M. Z. Frank. Is all that talk just noise? the information content of internet stock message boards. In *Journal of Finance*, 2004.

[DC01] S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment parsing from small talk on the web. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*, 2001.

[FYL05] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Hongjun Lu. The predicting power of textual information on financial markets. pages 1–10, 2005.

[GKRC06] B. Gu, P. Konana, B. Rajagopalan, and M. Chen. Competition among virtual communities and user valuation: The case of investor communities. In *Under review*, 2006.

[LSL+00] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. 2000.

[Mis06] Mishne. Moodviews: Tools for blog mood analysis. 2006.

[Ros58] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. In *Psychological Review*, pages 386–408, 1958.

[TS00] James D. Thomas and Katia Sycara. Integrating genetic algorithms and text learning for financial prediction. In Alex A. Freitas, William Hart, Natalio Krasnogor, and Jim Smith, editors, *Data Mining with Evolutionary Algorithms*, pages 72–75, Las Vegas, Nevada, USA, 8 2000.

[Wys99] Peter D. Wysocki. Investor relations and stock message boards: Who is chatting about your company on the web? In *Forthcoming in Investor Relations Quarterly*, 1999.