Thesis Proposal

James D Thomas
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jthomas+@cs.cmu.edu

May 22, 2000

Abstract

AI has long been applied to the problem of predicting financial markets. Recently, developments in both AI and financial economics have opened up the possibility for close collaboration between the two fields. First, a line of economics research has emerged that uses AI market forecasting as a form of applied econometrics. Just as importantly, an entirely new source of financially relevant data has become available and amenable to computational analysis: text. Access to text data – and the associated AI techniques for analyzing it – not only hold out the hope of improved prediction on the AI side, but also enable financial economics to ask new kinds of questions about how markets react to events.

I propose a line of research that develops a set of AI tools, specifically adapted to financial markets, to exploit this convergence for both economics and AI. This research has three main elements. The first thread takes representations from technical analysis – a semi-rigorous method of financial analysis – and combines them with search techniques and rule learning methods from AI to produce domain specific prediction algorithms. Second, I plan to adapt text classification and related techniques for financial market forecasting based on text from internet stock chat boards and news stories. And finally, I plan to adapt traditional economics techniques such as event studies to take advantage of the possibilities that text data offers.

Thesis Committee:
Katia Sycara, RI (Chair)
Bryan Routledge, GSIA
Andrew Moore, RI and CSD
Blake LeBaron, Brandeis University

Contents

1 Introduction

As long as there have been markets, probably, there have been schemes to beat them. The Dow Jones index itself was constructed as part of a scheme to beat the market.

AI has always had a side interest in prediction, dating back to expert system days through the neural network/machine learning revolution of the 1980's, through today. Traditionally, success has been mixed at best, with AI often promising far more than it could deliver. But for AI, this pursuit has largely been a hobby – while many researchers have dabbled, few have devoted their careers entirely to extracting meaningful signal out of financial data.

For economics and finance, however, this issue – can markets be predicted? – strikes at the heart of one of their deepest issues of all economics: market efficiency. The very core of modern financial economics – the efficient market hypothesis (discussed in more detail below, in section 2.1) implies that any past information is useless for forecasting the future. The necessary converse to that is that if one can use past information to predict future prices, that market cannot be efficient. Of course this is a slippery task. Slippery enough, in fact, that until recently the dominant orthodoxy of economics held that it was impossible[1].

This proposal aims at exactly that question – can past information be used to predict future prices?

The work proposed below proposes to answer that question, from the point of view of both a fascinating domain for Artificial Intelligence research, and a real drive to increase understanding of economic ideas of market efficiency. I believe that recent developments in AI and economics hold the potential for applying AI to the question of market inefficiency more powerfully than ever before. While there has been a long tradition of applying sophisticated, general purpose AI algorithms such as neural nets and decision trees to financial forecasting, recently economists have taken another approach, combining AI search methodologies such as Genetic Algorithms with representations taken from technical analysis, an informal school of techniques used by financial practitioners to analyze trends in markets. Even more importantly, an entirely new source of data has become available to computer processing: text. Not only is there a growing amount of financially relevant text available in machine readable form, generated both by web stock discussion boards and by targeted news services; but also, AI has shown great success recently in automating the classification and understanding of text.

The core idea of this proposal is that the development and integration of these techniques will provide will provide both an interesting opportunity for AI and economics. Since this proposal needs to speak to both AI and economics, this core idea leads to two different kinds of results:

[1] One of the reasons it is so slippery is that a test of market efficiency in isolation is impossible; market efficiency can only be tested jointly with an asses pricing model – this will be discussed further in 2.1

- AI side: A decision support system for financial decision making that integrates analysis of both traditional price and volume data as well as text data gathered from the web.

- Economics side: A set of AI tools, designed specifically with finance in mind, that allows economists to examine issues of market efficiency and how markets react to information, both from price and volume information, but more intriguingly, from the vast amount of financially relevant text information.

The rest of this document develops the ideas behind this proposal. The next section (section 2 sets the stage with the relevant background in AI and finance. I first discuss the core economics issue that lies at the heart of this effort, the efficient market hypothesis, and how the discipline of economics has traditionally attempted to empirically evaluate the relevant issues. Then I briefly review the history of AI's attempts to forecast financial markets, and finally I discuss a line of economic inquiry that focusses on using AI tools to predict future price patterns based on the past, with the goal of exploring market efficiency.

The following section (section 3) lays out a detailed roadmap of the proposed work for the thesis. The following two sections represent the core of the proposal. Section 4 discusses the proposed work on the artificial intelligence side. I start by describing supporting work on learning simple rules for prediction given numerical data, using text classification on stock bulletin board posts, and the integration of the two methods. Section 5 discusses the proposed work in economics, including some preliminary work on adapting the methodology of the traditional economics event study to ask questions about the reaction of financial markets to text data.

The following three sections address technical questions about the process of writing the thesis itself; section 7 lays out the evaluation criteria for the work proposed here, section 6 reviews the contributions to both AI and economics that this work proposes to deliver, and section 8 proposes a time line for progress.

2 Background

This section briefly summarizes the previous work needed to contextualize the proposed thesis work. Essentially, the core background is as follows. The idea of market efficiency – that markets process all available information efficiently, thus rendering them immune to forecasting – is a core concept of modern economics and finance. It is a long running thread of work in economics to understand when markets are and aren't efficient; recently an intriguing approach (from the AI point of view) has gained respectability: using simple genetic search on representations derived from technical analysis to demonstrate consistent out of sample predictability – this is interpreted as a sign of market inefficiency. On the AI side of the situation, there is a long running interest in applying function approximation methods to financial forecasting, as well as a recent growth in exploring what information AI can extract out of financially relevant text.

2.1 A Brief Detour into the Efficient Market Hypothesis

To understand why this problem is both extremely difficult and extremely interesting, the efficient market hypothesis needs to be explained. The core idea of the efficient markets hypothesis – the dominant paradigm for studying financial markets – is that markets process all relevant information and integrate it into the price signal efficiently. A necessary corollary of this is the following:

- *In principle*, past data of any kind cannot be used to predict future prices in financial markets

To computer scientists, this seems implausible: of course there should be patterns in past data that are usable to predict the future – there's useful patterns in all data.

But, there are strong theoretical reasons for believing that markets are efficient. The theoretical justification goes something like this: if there were patterns in past data one could use to predict future prices, someone would use those patterns, predict the future, make a huge amount of money, and in the process of making that huge amount of money make the past pattern invalid (there is a lot of math backing this up, see [8] for a deeper exploration, or Casti's [7] for a more casual explanation).

From one point of view, this is just common sense – there are many smart people in the world, and if there were such holes in the market, surely one of them would exploit it, thus making it disappear. In fact, once you start thinking about it that way, it's hard to imagine how markets wouldn't be efficient.

But, theoretical reasoning aside, whether markets are efficient or not is an empirical question. Over 30 years of examination, the EMH has proven to be surprisingly robust (see Fama's papers [10],[11] for a description of just how robust). However, exploring possible market inefficiencies is a well-travelled road in economics, and recently it's been travelled with some success.

One way to check market efficiency is to look for classes of stocks that show abnormally large gains over the long run. One widely studied phenomenon, documented by Debont and Thaler [3], is that stocks that have been underperforming the market for a few years tend to to overperform the market over the next few years. There are lots of papers in this genre (see [22] for a nice collection of papers).

Recently, another approach to exploring the efficient market hypothesis has arisen: instead of looking for classes of mispriced equities, economists look for trading strategies that produce out-of-sample excess profits against a passive strategy of buying and holding the equity in question; I discuss this further in section 2.3.

There is a difficult wrinkle to this. Notably, the notion of market efficiency is only meaningful in the context of a pricing theory: a test of market efficiency is always a joint test of a pricing theory and of market efficiency. Pricing theories (the most familiar of which is probably the Capital Asset Pricing Model) relate risk and return of individual securities and the market aggregates to understand how stocks should be priced in an efficient market.

So, for example, it would certainly be possible to find a trading strategy that had significant excess returns over a buy-and-hold strategy – however, if it incurred significantly higher risk than a buy-and-hold strategy, that result is perfectly consistent with the notion of efficient markets (stocks consistenly outperform bonds, for example, while carrying significantly higher risk). And furthermore, *even* if a trading strategy did have lower risk, the problem might be with the pricing theory itself.

In the economic literature that set the stage for the work proposed here (discussed below in sections 2.3 and 2.4, the way this problem is handled is by insisting on higher excess returns with lower risk (where risk is measured by the standard deviation of returns). While this approach is certainly persuasive – higher returns with lower risk is abnormal for most reasonable pricing models – it is far from an exhaustive answer to this question. Nonetheless, it is the approach I apply to the preliminary work presented in this document. Whether or not it suffices for the thesis is an open question.

2.2 AI and financial markets

There is a long history of using AI for financial forecasting. The typical approach since the mid 80s has been to use general purpose function approximation routines like neural networks or decision trees to predict price changes. For some good examples of these approaches, The Santa Fe time series prediction contest [24] featured tickwise foreign exchange data, and produced entries using neural networks, radial basis functions, and many other techniques. In addition, the long running neural networks in capital markets conference [25] features many papers on prediction.

Recently, work on text in finance has started to appear. Much of this concentrates on the analysis of news stories of financial interest. Of particular note is the work of Wuthrich et al [26], which uses a keyword approach on news stories for the financial forecasting of major indeces.

2.3 Technical Analysis and Market Efficency

Recently, a new approach to studying market efficiency has appeared, one that explores market efficiency through technical analysis style trading rules.

Technical analysis is too large of subject to be discussed in detail here (for a good introduction, see [19]). Roughly speaking, technical analysis is an attempt to use past price data to predict future trends in financial markets. Technical analysis uses many tools, but one of the most common (and the most relevant for my purposes here) is them is the moving average. The intuition is simple, and presented in figure 2.3; a financial time series is plotted (here, dollar/yen foreign exchange), and the 150 day moving average is plotted against it (the smoother, thicker line). When the current price crosses the moving average on the way up (marked with an upwards pointing triangle), a 'buy' signal is issued. Conversely, when the price drops below the moving average, a 'sell' signal is issued. Casual introspection of the graph reveals that many of the buy signals do indicate a steadily rising price.

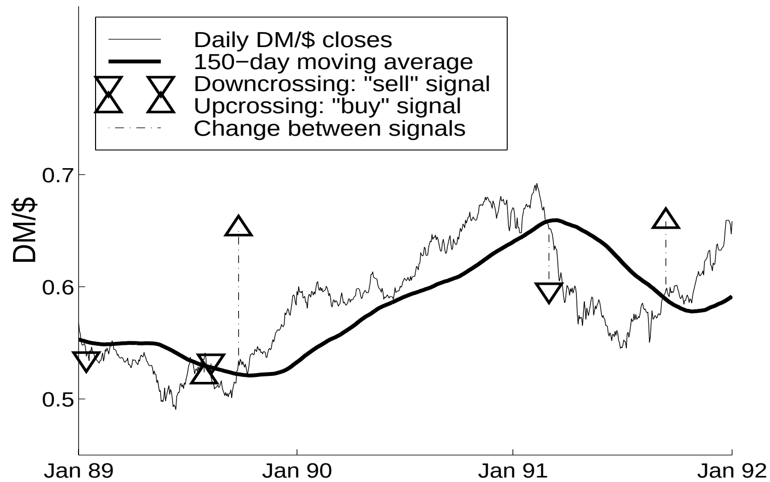

Figure 1: Moving Average

Technical analysis uses many other techniques, usually concentrating on visual representations. Intuitively, they look promising, and are actively used by practitioners, but until relatively recently had not been the subject of much formal study – the techniques lacked statistical verification of effectiveness, and the books explaining the techniques don't seem to be interested in either the formalization of the techniciques nor in studying their efficacy.

It wasn't until Brock, Lakonishok & LeBaron [5] that economists really had a way to measure what they meant. BLL took simple moving average trading rules and measured the excess returns they produced over a buy-and-hold strategy on almost 100 years of Dow Jones data. They found that the simple moving average rules consistenly produced higher returns with lower risk than a buy-and-hold strategy. Their real contribution was to link this observation to a null-model generated by bootstrap statistics; for the first time, economists really had statistics they could take seriously.

This basic idea of using the excess profits (excess with respect to a passive buy and hold strategy) generated by trading rules as a diagnostic for market inefficiency has been spun out into asking more specific questions about market efficiency. For example, LeBaron [13] took this basic technique of using excess returns generated by simple trading rules as a measure of inefficiency and used it to examine the efficiency implications of central bank intervention in foreign exchange rates.

2.4 Economics and AI meet: Genetic Algorithms for learning trading rules

The fact that simple technical rules appeared to produce excess returns under a wide set of situations begged the following question: if simple rule work well, might more complicated rules work better? This idea lead to the integration of two of the research threads discussed above– combining the idea of using technical analysis rules to probe for inefficiencies in markets with Artificial Intelligence techniques that would be used to learn more complex rules.

The basic idea was to use Genetic Programming to learn complex, technical analysis-like trading rules. This line of research was initiated by Allen and Karjalainen [1] on the S&P 500 and later extended by Neely [16] to foreign exchange data.

- Genetic programming trees
  - Leaf nodes were numbers
  - Higher level nodes consisted of functions, both generic math functions $(+, /, -, *)$ and technical analysis derived summary statistics (moving averages, maxes, mins)
- Fitness measure was annualized average daily returns (minus transaction costs)
- Data was split into three sections; training, validation, and test
- Rules were generated and evolved on training data, and the best performer on the validation data was selected and its performance evaluated on the test set

This approach produced promising results – both Allen and & Karjalainen [1] and Neely [16] demonstrated profits that were inconsistent with models of efficient markets.

However, this approach was very basic; it should be thought of as a starting point. It is with this spirit that I approached the problem.

2.5 Supporting work: first try with a GA

The results above show that simple AI techniques show promise for discovering inefficiencies in financial markets. But, I felt that further development of the techniques used could produce better results. In order to understand this domain better, I roughly duplicated the GP experiments (in somewhat simpler form) described above. However, I soon discovered that this domain was unusual. The key observation (discussed in work presented in [23]) was that this domain was much noisier than domains usually addressed by machine learning, and conventional machine learning techniques can be misapplied easily.

In investigating the use of this genetic programming methodology, I used three primary datasets; dollar/dm fx rates from 1979-1995, dollar/yen fx rates from 1979-1995, and S&P 500 data from 1950 to 1986. I used a standard genetic programming tree data structure. The leaves of the tree consisted of simple moving average rules of the following form:

- Genetic programming trees
  - Leaf nodes were simple moving average rules of the following form: if n-day moving average is greater than m-day moving average, be in-market; otherwise, be out-of-market. (n and m ranged over [1,150].
  - Higher level nodes consisted of logical operators: logical 'and' and logical 'or'
- Fitness measure was annualized average daily returns (minus transaction costs)
- Data was split into training and test sets; the first 50% of the trading days for training, the last 50% of the trading days for test.
- Rules were generated and evolved on training data, and the performance was evaluated on the test set by aggregating the rules in the final population.
- The selection method was deterministic tournament selection
- Standard GP mutation and crossover operators were used.
- Population size was small (12) and the number of generations run was small (30).

Note that this approach is considerably simpler than the techniques used in the papers described above. Since financial data is so noisy (remember, according to the EMH it's all noise), I am particularly interested in the idea of overfitting. To understand this, I wanted to measure performance as a function of maximum rule complexity. As a proxy for maximum rule complexity, I used the maximum depth of the genetic programming tree. The results of the algorithm are presented below in figure 2 for dollar/yen fx, dollar/dm fx, and the S&P 500 index. Each plot contains a plot of the out-of-sample test set excess returns for the algorithm against the maximum size of the GP tree.

The results here are startling; in the dollar/yen and dollar/dm case, performance decreases monotonically with the depth of the GP tree; maximum performance is produced with only a single comparison. And in the S&P 500 case, higher performance is found at greater tree depths, but the differences are slight, and the excess returns are all negative.

These results leave us with a depressing conclusion (well, depressing to an AI researcher, at least): the best way to learn technical analysis-like trading rules is to limit ourselves to small searches over spaces of simple rules. Making our representations more complex, just seems to introduce overfitting; it would seem, on the surface, that AI should just pack it up and go home.

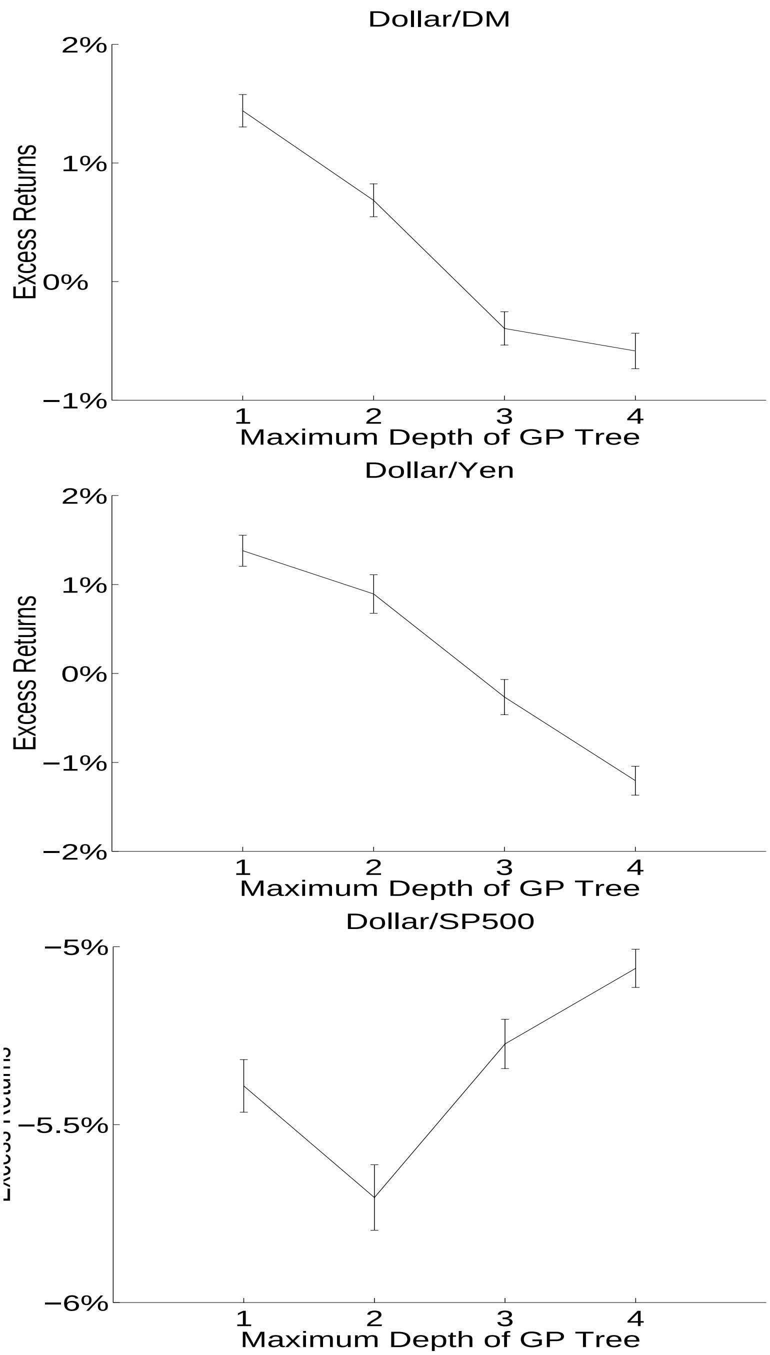

Figure 2: Test set excess returns (against buy and hold) against maximum GP tree size

The thesis I propose turns crucially on this fact. I believe that AI is not doomed in this situation – it just needs to step a little beyond the standard approaches and tailor its tools specifically to this domain.I propose to do this in two ways: first, re-build the simple rule learning algorithms here from the ground up with a primary focus on fighting overfitting, and second, to introduce a very promising new set of data: text.

3 Proposed work: The Roadmap

The previous section discussed existing work in this field. To summarize what I see as the existing state of affairs:

- A long tradition of applying general purpose AI techniques to financial forecasting.
- A recent appreciation by economics that there is a predictable component to stock returns, and that techncial analysis representations maybe capture some of that componennt
- A thread of research in econometrics that applies AI search algorithms to representations based on technical analysis and treats excess out-of-sample profitability (versus a passive buy-and-hold strategy) as evidence of market inefficiency.
- A promising set data of clear financial relevance – text – newly available in computer accessible formats, along with recently developed tools for its analysis and classification.

It is a core belief of this proposal that these conditions present an excellent opportunity to both advance the state of the art of AI in the financial domain, and help AI to make serious contributions to the understanding of efficient markets in economics.

The growing acceptance of the methodology of the papers that use AI to look for excess returns in financial markets demonstrated in [1] and [16] produces an alignment between AI and finance: doing good AI in the financial domain (with care shown to the concerns to economics) means doing good economics.

This means that the first goal of this proposed thesis is to make progress on the state of the art of AI in the financial domain. I believe that AI progress in this area will not come from general purpose function approximation techniques like neural networks or decision trees – approaches that have been well worked over in the literature – but rather from two relatively new approaches.

The first is the combination of technical analysis with AI search techniques, detailed above in sections 2.4 and 2.5 This has proved successful in the economics papers discussed in sections 2.4 – and I believe those attempts have only scratched the surface. Those papers applied genetic algorithms somewhat naively, with little attention paid to issues of overfitting or other traditional AI concerns. Although the experiments in section 2.5 indicate that the approach faces obstacles, section 4.2 is devoted to understanding how to avoid them, with some success.

The second is the use of text data. While there has been some interesting AI forays into this approach, it is still largely undeveloped and hugely promising. I address some of these issues in 4.3, using maximum entropy text classification to predict stock movements based on past text data from internet bulletin boards.

The development of these two approaches, and their integration both with each other and with other AI techniques provide the potential for an AI decision support system for financial professionals.

On the economics side, those same techniques can provide the basis for strong work in economics. Given the principle that AI has become established as a tool for studying market efficiency, a well-developed set of integrated AI tools specifically tailored to the financial domain provide a key first step to this kind of economics research. The key to integrating this work into the economics tradition will be first, turning those tools to questions of specific economic interest, and integrating the results with the traditional methodology of economics and econometrics.

The availability of text data allows for other investigations of financial markets. There is a long tradition of analyzing the reaction of financial markets to external events such as mergers and earnings announcements. Access to large amounts of financially text data allows much greater freedom in pursuing these goals: instead of being limited to clearly defined external events, we can now ask about nearly arbitrary events – so long as we can define their occurrence in text. A sample effort of this kind is discussed in section 5.2

The relationship between the sources of data, techniques, and hoped for research outcomes is presented graphically in figure 3

4 Proposed work: AI Side

This section is devoted to understanding the AI side of the problem. I have proposed an ambitious program, and the first few subsections discuss supporting work – first steps towards building simple rule learners, applying text classification to finance, and integrating the two methods. The following subsections discuss what is left to be done: data collection – setting up a series of benchmark datasets and gathering text date from new sources; the exploration of technical analysis as a source of representation for the sorts of simple rule learners discussed here; the adaptation of text classification methodology to financial domain specific applications; and finally, ideas for integration of these two forms of analysis, both with each other and other pertinent AI techniques.

4.1 A brief note about data mining

.

There is a potential confusion of terminology between AI and economics. To an economist, 'data mining' is a bad thing; it means picking out patterns

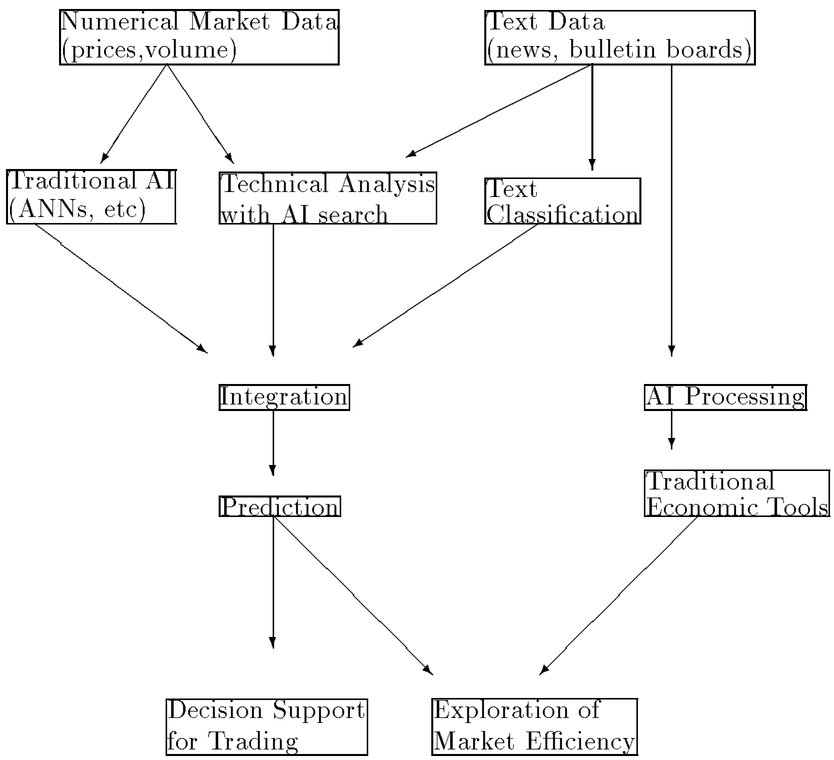

Figure 3: Structure of Proposed Thesis Work

in past data that are an accident of the noise in that data. But to a computer scientist, 'data mining' is a good thing – it means picking out patterns that are truly meaningful while avoiding patterns that are an accident of the noise. In order to perform the AI sense of data mining, you have to avoid the economic sense of data mining.

But the difference isn't entirely semantics. Given the extreme noise in financial data, the field of finance is justified in holding its empirical work to a higher standard. One of the prime dangers of the approaches described in section 2.1, where trading rules are examined for excess returns, is that given a universe of hundreds of trading rules, it is likely that a few will indeed produce statistically significant results (Sullivan, Timmerman, and White [21] address exactly this issue).

One of the promising aspects of applying machine learning algorithms to this problem, as pointed out by Neely et al [16] is that the approach of learning trading rules and testing them out-of-sample helps to get around this problem. The machine learning algorithms make no assumptions about specific rules; the freedom to pick rules that just happen to work is constrained, because we are testing *ways of learning rules* rather than the rules themselves. Since I test all of the algorithms on out-of-sample data, this provides a safeguard against picking rules that function well as an accident of noise. If the algorithm were finding rules that just happened to work well on the training data – or overfitting, in AI terms – those rules would likely perform poorly on the test set.

But this response is not entirely valid – AI algorithms are never completely independent of the data. Assumptions about representations, training times, validation schemes must be made. It is part of the art of AI to understand how to tune those choices to the peculiarities of datasets. AI is in the business of adapting algorithms to the domain at hand.

But this opens this work to the criticism – not that I am finding rules that work well on datasets because of chance – but that I am finding *algorithms* that work well on the relevant datasets by accident.

There is no definitive answer to this criticism. There is an inevitable tension: AI is all about adapting algorithms to specific domains; but from the point of view of economics, this is just asking for trouble.

I propose the following approach to dealing with this problem:

- Set aside a significant chunk of my data as a final test set.
- Split the remaining data into train and test sets, and use that data for the iterative process of algorithm design.
- Only after I have made all design choices for the algorithm, then evaluate the performance of the developed techniques on the final test set.
- Conclusions about the performance of the algorithm should be drawn based only on the performance on the final test set.

Of course, conclusions will be supported by bootstrap hypothesis testing. Leaving aside a final set of data until the algorithm design process is completed will provide insurance the I am not simply overfitting my algorithms to the data.

I plan two ways of setting aside data. The first is to chop off temporally connected chunks off of datasets I am examining. So, for example, if I were to look at the S&P 500 index, I could leave the last 1/3 of the trading days in my final test set.

Of course there is a danger in this: a significant regime change in market dynamics could render any algorthm here useless. But that must be thought of as part of the challenge of working in the domain of financial markets.

4.2 Learning Simple Rules

The financial domain poses unique challenges to AI. The fact that overfitting sets in such simple representations (in the results described in section 2.5) suggests that this domain is extremely noisy by machine learning standards.

Traditionally, the issue in GP is how to search a huge parameter space efficiently. This data appears to leave us with a different set of challenges; neither building more complex representations nor understanding how to search them more efficiently will gain us anything.

How does one make progress in such an environment? I plan to address these challenges with two specific strategies: the first is to examine ways to fight the enormous amounts of noise in these environments, the second, to integrate a promising new source of data: text.

4.2.1 Supporting Work: Representation

Getting the correct representation is often key to avoiding overfitting. There is a tradeoff: sometimes representational flexibility is needed to capture complex hypothesis about the data; but also, that extra representational flexibility can serve as fodder for overfitting if it is not needed and the data is noisy.

One way to attempt to fight overfitting in the financial data is to narrow the representation. As just a first step in this, we examined the role of the moving averages presented to the algorithm. In the results presented in section 2.4, the genetic program was free to search over all moving averages. However, if this freedom is just causing the algorithm to overfit, then limiting the GP search to a limited subset of moving averages should improve performance. To test this, we re-ran the experiments, limiting the GP to searching over 10 moving averages (1, 2, 4, 6, 10, 17, 29, 49, 86, and 149 – chosen because they cover the distance between 1 and 150 and are exponentially distributed). The results, averaged over 300 trials, are presented below, in figure 4.

The results here are straightforward; using a subset of moving averages for comparison is dramatically superior to using all moving averages across the board. Furthermore, in the dollar/DM case, performance now increases with greater GP tree size (at least until a depth of two). In the dollar/yen case,

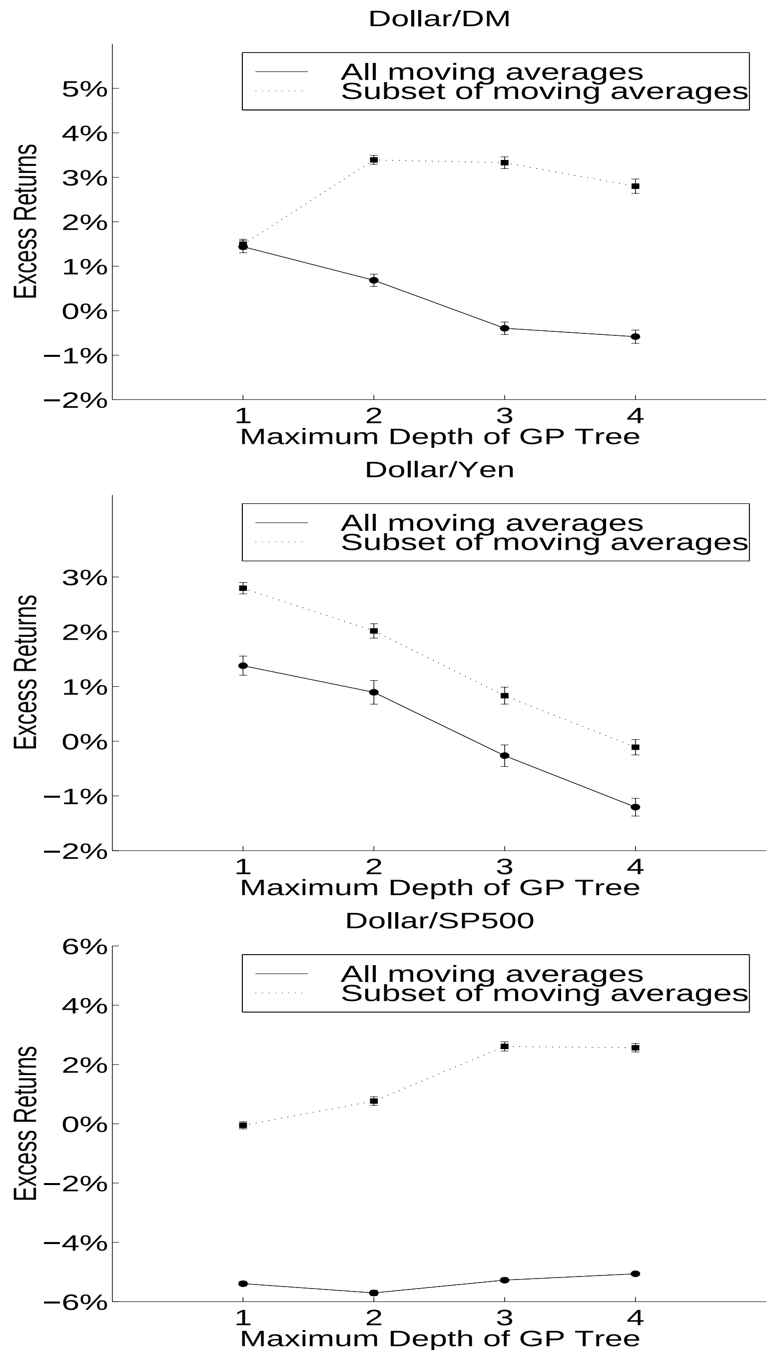

Figure 4: Test set excess returns against maximum depth of GP tree

performance is superior, but still decreases as tree depth increases. In the S&P 500 case, there is dramatic performance improvement across the board.

Although the results presented in this section may not seem like substantial AI research, I feel they do illustrate the following conclusion: in dealing with financial data, getting every aspect of the representation right is more important than the traditional AI concern of efficient search of huge spaces of potential rules.

4.2.2 Supporting Work: Voting Methods

One way to fight noise is to combine multiple predictors in a voting scheme. Recently, there has been much interest in methods like bagging [4] and boosting [20]. This is another promising methodology for improving generalization ability in the face of extreme noise.

To explore this idea in a financial context we tried two methods: simple aggregation over trials, and bagging. In the case of simple aggregation, since we had run our algorithm over 300 trials, we simply took n trials of the in market/out of market signal and averaged them together (giving us effectively $300/n$ trials of the aggregation algorithm).

Bagging functions similarly; the key difference is the training set used to train the learner. For each trial, a novel bootstrap training set of exactly the same size as the original training set is constructed by sampling from the original training set with replacement. The GP learner is then run on this bootstrap training set, and the resulting in market/out of market signals are averaged over n trials, as above.

The results are presented below, excess returns plotted against the number of trials aggregated for both boosting and simple aggregation. The key benchmark here is the case of 1 trial aggregated (which is really no aggregation at all– this is the result presented in figure 4).

Two conclusions are immediately clear. First, except for the dollar/yen case, the simple voting method is clearly superior to the boosting technique. In the dollar/yen case, the simple voting method seems slightly superior across the board, but the standard error bars overlap enough to avoid drawing a strong conclusion. In any case, given the consistent performance of the simple voting method, it is clearly preferable.

Second, aggregating multiple indicators gives clear, steady improvement in all three cases, showing marked improvement from 3.3% in the dollar/DM case to well over 5%, a jump from 2% in the dollar/yen case to roughly 3.5%, and an increase from less than 1% to 2.5% in the S&P 500 case.

The results presented above demonstrate that with some attention paid to issues of overfitting, progress can be made in learning these simple rules.

4.3 New Data: Text

While the techniques and proposed work described above are promising, the most interesting potential avenue of progress is in introducing a whole new

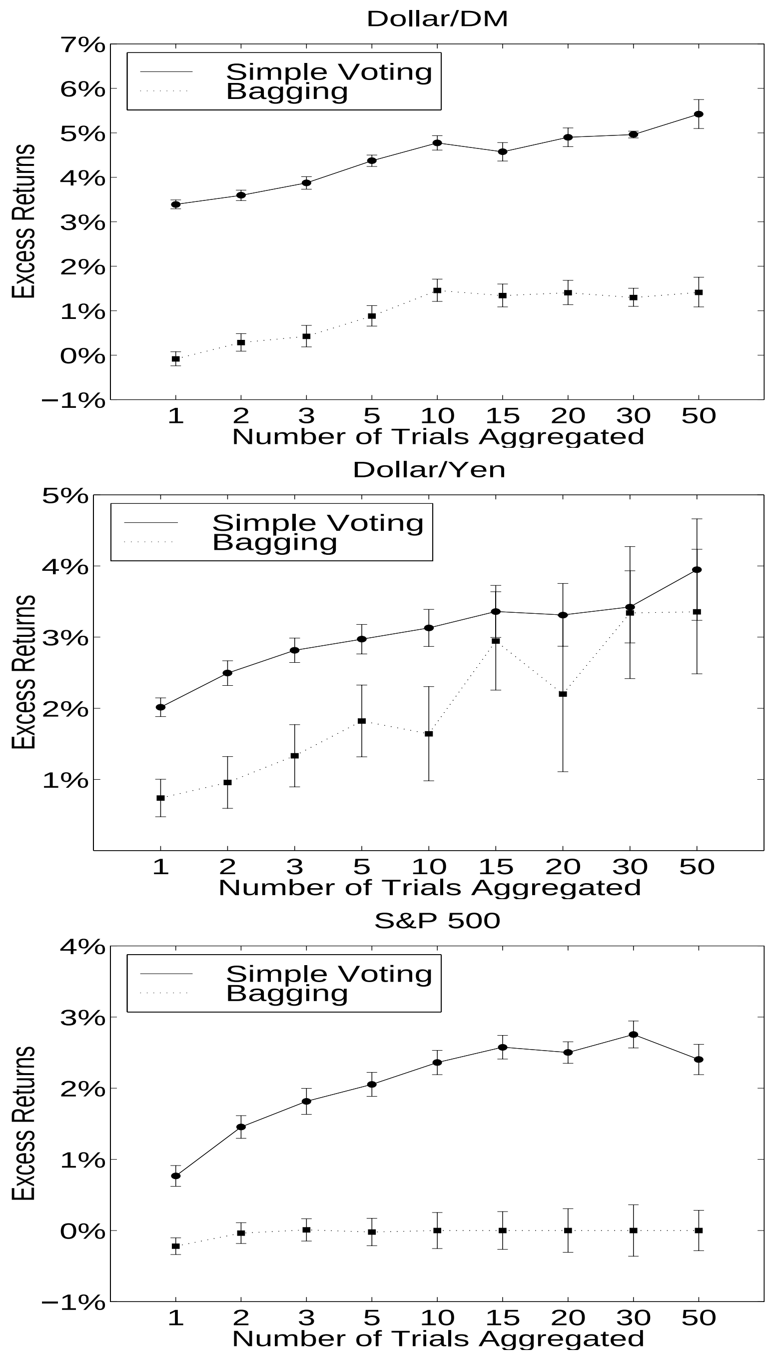

Figure 5: Test set excess returns against number of aggregated trials

source of data: text. Recent events have made this possible. First, more and more relevant text is available in computer-readable format. And second, there has been an explosion of work in AI approaches to understanding and classifying text.

Finance has long been interested in the influence of outside events on financial markets. The classic event study methodology (summarized nicely in chapter 4 of Campbell, Lo, & McKinlay [6]) studies the reaction of markets to outside events.

4.3.1 Supporting Work: Ragingbull.com Bulletin Boards

For data, we are specifically concerned with using text information for prediction. As such, we took the forty most popular discussion boards on the financial website www.ragingbull.com for the week of November 18, 1999. We eliminated all boards not devoted to a specific stock, all boards whose stocks were not trading on NASDAQ or the NYSE as of Jan 1, 1999 those boards with stock prices of less than $1 on Jan 1, 1999, and boards with less than 10000 messages. This left us with 12 stocks and the accompanying text from their bulletin boards.

I downloaded every post for each one of the bulletin boards from January 1, 1999 to December 31, 1999, and downloaded daily closing prices and trading volume for each stock from the quote server at finance.yahoo.com. For each day t, we aggregated all text produced after market close on day $t-1$ and before market close on day t[2]. This data provided the raw material for our prediction algorithms – text produced before close on day t could be used to predict whether the closing price on day $t+1$ would be higher or lower, and an appropriate trading strategy implemented.

This one year of data gave us 252 training days. In all the results that follow, we use the first 52 days as the start of the training set and report results averaged over the last 200 trading days.

I took two main approaches to using this text data for forecasting. The first used a text classification methodology to map the entire set of text data produced on a trading day onto an in market/out of market signal.

The second approach concentrated on merely the *volume* of messages and words produced, treating them as time series signals. This allows us to apply the GP learning approach we developed in sections 2.5 and 4.2, treating the volume of messages and words just like past price signals.

Text Classification

For text classification, we used the rainbow package developed by McCallum [15], which provides a variety of potential classification methodologies. In general, text classification techniques work as follows. The entire training corpus is lexed, and the number of occurances of each word is calculated for each document, producing a large matrix. This can be visualized as an n-dimensional space where each dimension corresponds to a specific word in the corpus (here,

[2]For simplification data produced on non-trading days was discarded

n is the number of unique words in the corpus). Each document's position on a specific dimension is determined by the number of occurances of the word corresponding to that dimension in the document. Since we now have a set of labelled examples in n-dimensional space, we are left with a simple classification problem, to which various algorithms like naive bayes classification [14] can be applied. Maximum entropy text classification is simply the application of maximum entropy techniques to this specific classification framework [17].

Starting at trading day 53, we applied the algorthm as follows. For day t, we inserted the text for days $[1...t-1]$ to the training set, trained the classifier, and had it give probabilities for "up" or "down" based on the text for day t. If the 'up' probability was greater than .5, we issued an in market signal; otherwise, an out of market signal. As we moved to day $t+1$, we added the text from day t to the training set and retrained the classifier.

We used both maximum entropy and naive bayes classifiers. Results, averaged over the last 200 trading days over all twelve stocks, are presented in the table below (The buy and hold strategy is presented for comparison). We applied a .1% round trip transaction cost.

Approach:	Total Returns	Excess Returns
Buy and hold	107.70%	N/A
Maximum Entropy	108.83%	1.13%
Naive Bayes	89.40%	−18.30%

These results are a little disappointing; the excess returns for the maximum entropy approach is barely postive, and for the naive bayes approach is strongly negative. If we remove transaction costs, the excess returns rise to 6.91% for the maximum entropy approach and −13.78% for the naive bayes approach, indicating decent performance for the maximum entropy approach – however, it is difficult to make a convincing case for market inefficiency while ignoring transaction costs.

GP Learning

Another approach is to think purely in terms of the *volume* of text produced. It is easy to count both the number of messages and the number of words produced for a given trading days worth of text. This leaves two sets of simple time series data – time series data that could be used as a raw data for trading rules, analagous to the way past price signals were discused in section 4.2

To test this, I used the same GP methodology developed in section 4.2, replacing the past price signals with daily counts of message volume and word volume.

However, because the nature of this data is different than the forex and S&P 500 data used above, I made a few slight modifications, all made from reasonable first principles. There were two key concerns motivating these changes:

- Volatility: The stocks at issue in this dataset are extremely volitile. Direct comparisons between moving average rules produce rules that are in market roughly half of the time and out of market roughly half of the time. In a case where a stock is likely to move up (or down) consistently, being in market half the time and out of market half the time is not a good policy. So, we want to add flexibility to our representations to allow for rules that can be in market (or out of market) most of the time. I realize that this extends the representational flexibility, potentially leading to overfitting, but since there is an a priori reasonable motivation for it, and I restrict representational flexibility in other ways, I feel it is justified.

- Smaller dataset: In the cases above, we had thousands of datapoints. For this data, we have 250. This means that the dangers of overfitting, already large in financial datasets, are even more of an issue. That, combined with the additional representational flexibility we introduce because of the concerns describe above, make constraining the representation in other ways crucial.

These concerns led us to make the following modifications to the representation:

- Comparison multipler: in order to allow for rules sent in market (or out of market) signals most of the time, I introduced a multiplier to the moving average comparison. Instead of a straightfoward comparison between moving averages, the algorithm compares one moving average to k times another moving average, where k can range in discrete steps over the range [0,2].

- Smaller subset of moving averages: Since we want to reduce representational flexibility, and since there is an initial training set of 50, I limit the moving averages considered by the GP learner to three: 10, 30, and 50.

- Constrained comparisons: To further constrain representational flexibility, we allow only comparisons between the message/word volume of the current day and the message/word moving averages (*not* between two moving averages, as in section 4.2)

- Updated learning: Since we only have 250 datapoints, we first run the algorithm with the first 50 datapoints as a training set, and report the results for the next 25 datapoints. Then, we add those 25 datapoints to the training set and re-run the algorithm. We repeat this process for the rest of the data, all 250 points, for each stock.

- Smaller number of generations: As an additional guard against overfitting, we reduce the number of generations searched from 30 to 10.

In summary, the rules that we want to learn, that map past information onto in market/out of market signals, have the following form: let v be today's message volume or word volume, $ma(v, n)$ be the n day moving average of past volume values, and k be an arbitrary constant in the range [0,2]. Then each leaf-node rule has the following form:

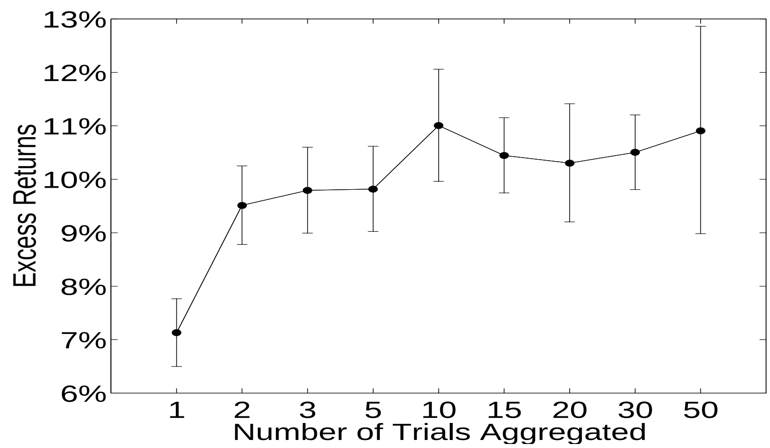

Figure 6: Test set excess returns number of aggregated trials

if $v > k * ma(v, n)$, issue a 'buy/hold' signal.

Since aggregating signals from multiple trials proved successful in the previous examples, we apply that here as well. The results are presented below in figure 6. I plot the excess returns (averaged over all 12 stocks) against the number of trials aggregated.

The results here are straightforward. Even with no aggregation, excess returns of approximately 7% are shown and aggregating over trials shows significant improvement (to about 10%). It looks like the improvement continues as the number of trials aggregated gets larger, from 9.5% to almost 11%, but given the size of the error bars, that conclusion is tentative.

Integration

The two methods described above are promising individually, but the promise of integrating the two together holds out the possibility for even greater success.

In order to do this, I simple took a weighted average of the probability output by maximum entropy text classification algorithm and the '1' or '0' signal given by the GP learner (and aggregated GP learners). However, the question of how to weight the two approaches is not trivial. The GP learning approach gives us 0 or 1 values, while the maximum entropy approach gives us values typically between .35 and .65. This suggests that we don't want to weight them equally, since the GP vote would dominate the combined value.

A good first guess might be to give the maximum entropy text classification signal a weight of .75 and the GP trading rule learner a weight of .25. The results are plotted below in figure 7, both the GP trading rule learner and the weighted combination, excess returns against the number of aggregated trials.

The .75 weighting used in the above calculations seems arbitrary; it is possible that the excellent results presented above are an artifact of that exact

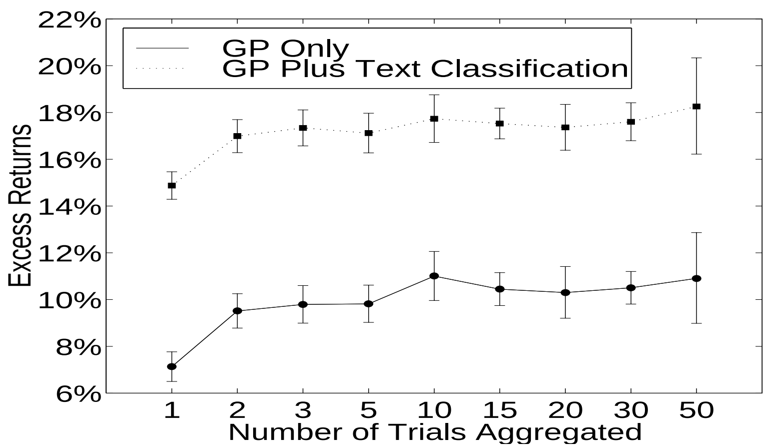

Figure 7: Test set excess returns number of aggregated trials

weighting, which was not selected based on firm theoretical principles. It is logical to ask how sensitive the results are to the precise weighting. To test this question, I examined the performance with weightings ranging from 1 to 0 (with no aggregation across trials). The results are presented below, in figure 8.

First, examining the left side of the plot, in the areas between weighting of .95 and .7. It is clear that the integration of the two approaches produces superior results, except in the case of a weighting of .8. As expected, as the weighting approaches .5, the GP almost takes over and the results are almost identical to that of the GP alone for weightings of .5 or less. But, what is important here is that the integration of the two methods works for a large range of reasonable values (except for the dip around .8, which looks like some sort of odd outlier – other runs of these algorithms with slightly different parameters follow roughly similar patterns, but without the gap).

It is somewhat surprising that the maximum entropy text classification approach produces virtually no returns by itself, but produces large excess returns with even a small contribution from the GP learner. This fact warrants further investigation.

Statistical Testing

Given the immense amount of noise present in financial data, proper statistical testing is crucial. Following the lead of similar economics papers on trading rules and excess returns (such as [5], [16]), we turn to bootstrap hypothesis testing (see Efron & Tibshrani [9] for a nice introduction) to examine the statistical significance of our results.

Bootstrap hypothesis testing works as follows:

- Define the null hypothesis.

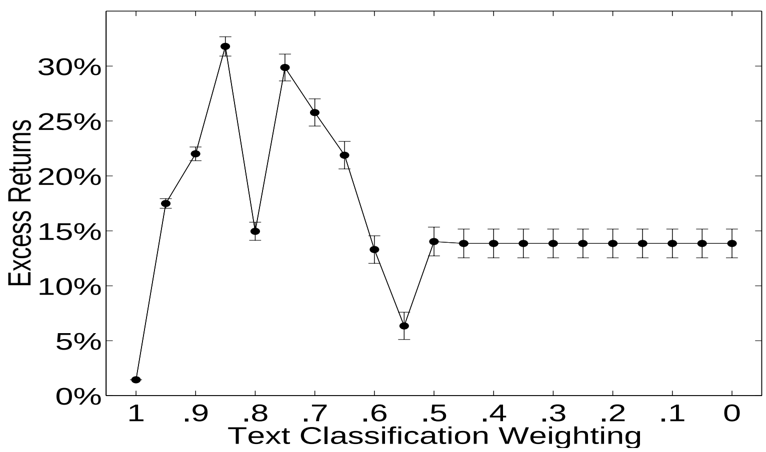

Figure 8: Test set excess returns against text classification weighting

- Generate a number of datasets by the null hypothesis.
- Run the algorithm on these bootstrap datasets.
- Compare what proportion of the bootstrap datasets produce results exceeding that of the real dataset; this is the appropriate p-value.

In our case, our null hypothesis is that the text associated with a trading day has no predictive power. So, to generate our bootstrap datasets we simply scrambled the chunks of text associated with each daily return. I generated 300 bootstrap datasets, and then ran both our GP learner and the maximum entropy text classification algorithm over each one. Then, I calculated the proportion of our bootstrap datasets which had returns greater than the results reported in this section. This proportion plays the same role as the P-value of traditional hypothesis testing – it is a measure of the probability that our results were generated by chance (ie, the null hypothesis). The results are plotted below, in figure 9.

This figure plots the p-values against the relative weighting of the text classification scheme, to see the p-values both for the raw text classification case (weighting of 1), the pure GP learner case (weighting of 0), and the integrated cases. Here, the bootstrap p-values for high text classification weightings are all significant; however, despite relatively high excess returns in the case of the GP learner alone, the bootstrap p-values pop up above the .05 mark.

In addition one must account for risk. There is a well known tradeoff between risk and return in investment; it is possible that our trading strategy is simply gaining excess returns by taking on more risk. Thus, we must measure the risk involved in an investment strategy (here, I measure risk by the standard deviation of monthly and daily returns). Since the stocks involved are extremely

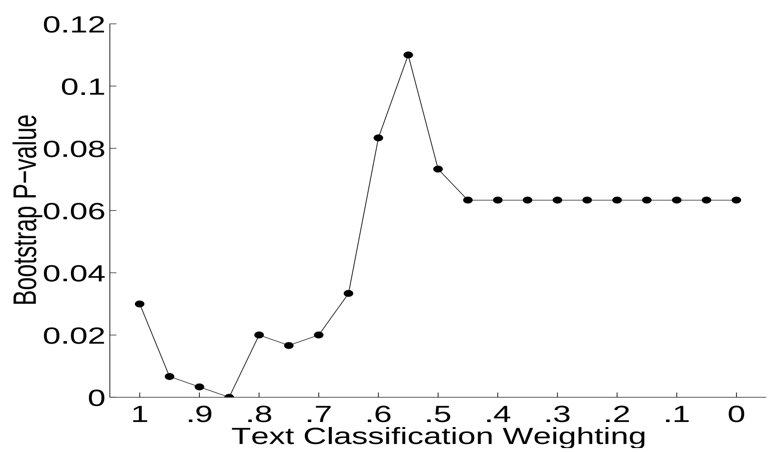

Figure 9: Bootstrap P-values against text classification weighting

volatile, and our trading strategy spends time not invested in the stock, we would expect our risk to be lower. The table below – which presents the total returns, excess returns, bootstrap p-values, and mothly standard deviation of returns for both the buy-and-hold and the integrated strategies – bears this out.

	Total Return	Excess Return	Daily Std	Monthly Std
Buy and hold	107.70%	N/A	3.38%	23.12%
1	108.83%	1.13%	2.46%	14.95%
.95	116.59%	8.89%	2.56%	16.95%
.9	118.77%	11.07%	2.72%	18.75%
.85	123.50%	15.80%	2.84%	19.68%
.8	115.37%	7.67%	2.88%	19.81%
.75	122.57%	14.87%	2.95%	20.44%
.7	120.59%	12.89%	2.95%	20.87%
.65	118.72%	11.02%	2.96%	21.15%
.60	114.56%	6.86%	2.99%	21.21%
.55	111.20%	3.50%	3.00%	21.11%
.5	114.91%	7.21%	3.00%	21.16%
.45 - 0	114.83%	7.13%	3.00%	21.14%

4.4 Proposed Work, the details

The work described above indicates some promise in the basic approach I have proposed. However, it is only the beginning. Benchmark datasets need to be gathered, other forms of technical analysis (besides moving averages) need to be investigated for suitability, and perhaps most interestingly, techniques for dealing with text need to be tailored to the financial domain. These issues are discussed in detail below.

4.4.1 Proposed Work: More Datasets

These results clearly need to be extended to more data. I propose to set up a number of benchmark datasets, both for the strictly numerical approaches described in section 4.2 and for the integrated text approach of 4.3.

For the numerical approach, I propose to develop a series of benchmark financial time series, using the foreign exchange and S&P 500 data described above as the core. Although the exact composition, it should include futures and options contracts, commodities, equity indeces, and more. Once set up, a thorough comparison of numerical techniques across 10 or 15 datasets would be more satisfying that what has been presented here, and would also allow for easier comparions with other techniques.

For the text approaches, I am limited by markets for which relevant text exists in machine readable formats. Certainly, the angle of web bulletin board posts should be pursued – I am currently gathering more data from raging-bull.com, as well as similar bulletin boards on yahoo.com. In addition, linking professionally generated news texts such as the Wall Street Journal and other related news organizations. Undoubtedly, the two different classes of text would require different approaches to exploit each information source to its fullest.

In addition to providing stronger comparisons between techniques, having a wide set of benchmarks would be a first step towards comparing the efficiencies of different markets.

4.4.2 Proposed Work: Other Technical Analysis Techniques

The rules used above in section 4.2 are simple, and based on moving average statistics, the probably the best known technical analysis rule. But even a casual perusal of books on technical analysis [19] shows that there are many other possible rules. Trendlines, relative strength indicators, oscillators and many other rules have been used by practitioners for decades. Recently, some investigation of the effectiveness of these techniques has started; for example, Osler [18] has studied the 'head-and-shoulders' pattern on foreign exchange trading and found some excess profitability.

But, from an AI perspective, we want to think of these sorts of rules as *representations*. Given a suite of benchmarks and an undertanding of how to avoid overfitting when learning simple rules, I propose to systematically test the other standard techniques of technical analysis for their suitability as reprsentations. It's clear that the moving average comparison has some power as a representation – do moving oscillators have the same power? Given a set of benchmark datasets, I propose to systematically examine technical analysis constructions for their suitability as representations for the sorts of simple rule learners explored here.

This would have two main advantages:

- Since representation is a key to handling the noise in financial data, the representations developed by technical analysis that have (anectodally, at least) functioned well and have been used by practitioners for years seems to be the best place to start exploring.

- This would also serve as a systematic exploration of the efficacy of such technical analysis constructs. There is a large informal literature based on them, but little (until recently) rigorous empirical analysis.

4.4.3 Proposed Work: Adapting text classification methodologies to the financial domain

A deeper understanding of how to adapt text classification to the financial domain is a key component of the proposed work here. Some initial ideas for exploration include:

- A thorough comparison of text classification techniques across multiple datasets.
- A deeper exploration of the role of the volume of text. For the bulletin board text data sets, the work presented here strongly suggests that this is an important factor. For other text datasets – news stories, for example – the volume of text is reasonably fixed.
- Maximizing total return instead of classification accuracy – the traditional text classification methodology targets classification accuracy, whereas the work here focusses on the excess returns of a trading strategy.
- Integrating the text classification approaches with domain knowledge about finance. Clearly, some keywords and combinations of keywords are more important than others, and integrating that knowledge as a priori information potentially offers added performance.

4.5 Proposed Work: Integration and Comparison

As discussed in section 2.2, there is a long history of applying general purpose function approximation algorithms to financial prediction. The establishment of a broad benchmark suite of test datasets will allow for a thorough comparison of the methodologies proposed for development here with existing general purpose function approximators.

Also as shown above, performance has the potential to increase with the integration of multiple predictors. It is possible that the best prediction system would come out of thoughtfully integrating the techniques here with existing metholdologies into a single prediction system.

5 Proposed work: Economics side

5.1 Using our methodology as a test for efficient markets

Given a well developed AI methodology, the next step is to adapt it to specific questions in economics. So far, most of these questions have centered around the predictability of future prices given past prices. The exploration of the techicanal analysis-based techniques described here holds hope for stronger apraoches to these questions. However, the integration of sources of text data into this picture allows for more interesting questions to be asked: since text data is a far better proxy for real world events than past prices,

In order to take the AI work and make it relevant to economics, however, special attention to statistical testing and issues of risk must be addressed. The specific areas where I propose work follow.

5.2 Proposed Work: Adapting event study methodology

One long-standing thread in the examination of market inefficiency is the event study. The basic format of the event study is simple. Identify some important event, and then measure the abnormal returns (relative to some null model) that follow said event. If the statistics check out, you have evidence that the market isn't reacting completely efficiently to outside events. A good example of this is the study of Jarrell and Poulsen [12], which examines the impact of takeovers and finds that shareholders in the acquired firms consistenly receive high abnornmal returns over time, while those of the acquiring firms often produce little (or negative) abnormal returns.

Event studies have become one of the most powerful tools in examining issues of corporate finance – they allow the market role played by clear changes in corporate structure to be explored.

The computational infrastructure being developed for this thesis allows me to adapt the event study methodology to apply to the newly available text information. Traditionally, event studies have been linked to easily definable, discrete events such as corporate equity issuance announcements [2] or takeover/merger announcements [12]. But now, we have the ability to do two new kinds of examinations of market efficiency combining the newly available text data with event study methodology. I have the following specific sorts of experiments in mind:

- Arbitrary text events: Instead of being limited to event studies based on clearly defined, exogenous events, we can run event studies based on arbitrary events in our text stream. Instead of performing an event study on the actual event of a takeover, one could perform event studies on the event of a *rumor* of a takeover (limited, of course, by our ability of our text classification technology to properly define our event).
- 'Reverse' event studies: Given a time series of price data and an accompanying body of text, we can reverse the usual causal structure of event

studies. Instead of picking an event and looking at the resulting price behavior, we could pick the price behavior we are interested in, and study the sorts of text events that preceed such price behavior.

The sorts of studies that might come out of this work might not look like traditional event studies. In terms of sample density, what I propose is certainly different – the Jarrell & Poulsen [12] study examines only 663 events for hundreds of stocks over 30 years; we could easily define certain text events with hundreds of times that occurance frequency. Certainly, this fact alone introduces a whole host of methodological problems (summarized nicely in [6]), but that shouldn't detract from the promise of a novel way to explore market inefficiency. The basic framework – measuring the abnormal returns produced by clearly definable event – is intact, and our flexibility in defining what that 'event' really is can only increase.

5.2.1 Supporting Work: Ragingbull.com bboards event study

As a sample of what this approach might bring us, I performed a simple event study based on the Ragingbull.com data. Let us start with a simple idea: increasing hype translates into increasing share returns. How would I test this as an event study? First, we have to define a suitable proxy for increasing hype in terms of the data we have – the number of messages posted per day. Let $V(t)$ represent the number of messages posted per trading day, normalized by a 10-day moving average. Then let us represent our idea of increasing hype as follows:

We will set out two conditions. First, we start out with an above average number of messages; second, that number of messages increases with each day. This leaves us with the following formal conditions:

- $V(t-k) > 1$
  item $V(t-k+1) > c \cdot V(t-k); (c > 1)$

Here, c represents a constant that denotes the amount of increase required per day. For simplicity, we picked $k = 2$ (two days of increasing numbers of messages) and $c = 1$, giving us the following specific conditions:

- $V(t) > V(t-1) > V(t-2) > 1$

We then searched through the dataset of stocks and their associated Raginbull.com postings for all days which met these criteria (To avoid overlap in measurment periods, we excluded events that fell within the measurement window of a previous event). We then measured the cumulative abnormal returns [3] from the ten days preceeding the event day t to the 10 days after event day t. The plot is displayed below in figure 10

[3] cumulative abnormal returns, or CAR, is typical event study terminology – it means the same thing as excess returns, aggregated over a number of days after the event happens. We assuming a constant-mean-return model; the returns for these stocks are so high that a market model wouldn't make much sense

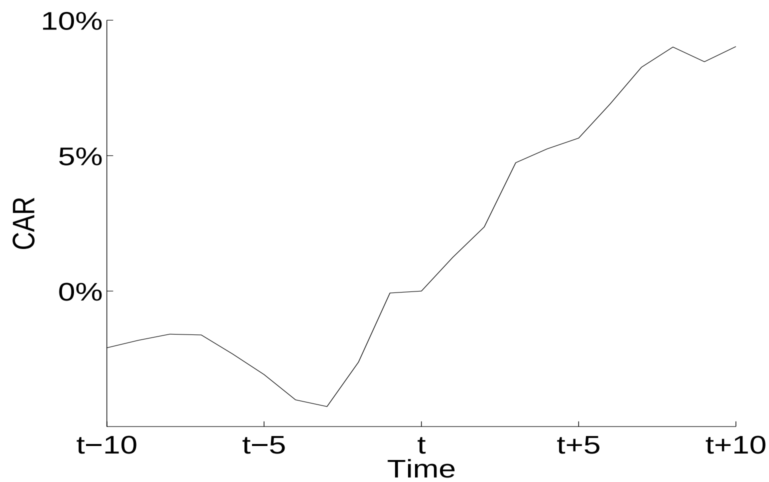

Figure 10: Event Study

To understand this figure, it helps to examine the CAR from both before and after t. Unsurprisingly, from $t-4$ onwards, we see a steady rise in CAR. This is probably what causes the increasing amount of discussion on the bulletin boards – as returns increase, discussion increases.

After t, CAR rises steadily – a clear sign that the 'event' of increasing buzz gives rise to abonormal returns. The day-by-day CAR results, as well as Boostrap p-values are presented below[4]. For the bootstrap hypothesis testing [9], we worked under the null hypothesis that the normalized message counts were irrelavant to returns. Thus, we generated 200 random scramblings of the message counts, ran the event study on each scrambled time series, and present as p-values the percentage of bootstrap returns that were higher than the real return.

The key thing to note here is to note that CAR steadily rises over time, and is statistically significant over much of the range from $t+3$ on. I offer these results in the spirit of the proposal – as strong evidence that there is much to be pursued here.

[4]I use bootstrap hypothesis testing because a proper calcuation of the variance of returns is difficult, because of possible covariance between returns on stocks with overlapping event windows; I know that using bootstrap hypothesis testing is not usual methodology for event studies, but I hope that it at minimum clearly demonstrates that this is a promising area for further exploration

Date	CAR	Boostrap P
t	0%	N/A
$t+1$	1.26%	.1350
$t+2$	2.40%	.1150
$t+3$	4.85%	.0100
$t+4$	5.39%	.0400
$t+5$	5.81%	.0700
$t+6$	7.15%	.0600
$t+7$	8.62%	.0350
$t+8$	9.43%	.0300
$t+9$	8.84%	.0550
$t+10$	9.03%	.0350

6 Expected Contributions

To recapitulate the key expected contributions of the thesis:

AI side:

- An understanding of how to adapt AI techniques to the unique challenges of the financial domain, including techniques for handling text, and integrating representations currently in use by practitioners (technical anlysis) in simple rule learners.

- The construction of an integrated data mining/machine learning system that takes both traditional numerical financial data and text data, applies methods of text classification, technical analysis-derived AI methods, and traditional function approximation methodologies, for financial forecasting and decision support

Economics side:

- A thorough understanding of the application of AI techniques to testing market efficiency across a wide set of benchmark datasets, both numerical and textual.

- The application of event study methodology to understanding how text (both as a proxy for real world events and as a source of information itself) plays into movements of financial markets

7 Evaluation Criteria

Since the work in this thesis has to answer to both economics and AI, the evaluation criteria differ. Traditionally, work in AI compares itself against other techniques. Unfortunately, due to the fragmented nature of work in this field, that is difficult. But, given a wide set of benchmark data, comparisons with more traditional function approximation methods can be constructed. For economics, the emphasis is much more on clearly demonstrating that the results are not an artifact of the data – this requires careful attention to bootstrap hypothesis testing and measures of risk. Both of these sets of criteria will have to be addressed for the work to be considered a success.

For these results to be meaningful to either AI or economics, the problem of overfitting has to be addressed. In section 4.1, I discussed my plan to hold out a final test set for evaluation after all algorithm design has been accomplished. I feel this is a strong enough methodology to protect against overfitting, and I feel that it should become a standard approach in AI and finance.

For the work on adopting event study methodology to deal with text data, the evaluation criteria are clear cut and well established: statistical significance in abnormal returns for the event in question. Again, however, the results should be examined across multiple text datasets.

8 Approximate Time Line

Month	**Year**	**Activity**
May	2000	Proposal
June	2000	Industrial Internship + Data Collection
August	2000	Simple Rules + Data Collection
October	2000	Text Classicication Development
December	2000	Integration + Other Techniques?
February	2001	Event Studies: Bboard and News Data
April	2001	Adapt AI prediction work for economics
June	2001	Final Coordination
August	2001	Writing
October	2001	Writing
December	2001	Defense

References

[1] Franklin Allen and Risto Karjalainen. Using genetic algorithms to find technical trading rules. Technical report, Rodney L. White Center for Financial Research, 1995.

[2] P. Asquith and D. Mullins. The returns to acquiring firms in tender' offers: Evidence from three decades. *Journal of Financial Economics*, 15:61–69, 1986.

[3] Werner F. M. De Bondt and Richard H. Thaler. Does the stock market overreact? *Journal of Finance*, 40(3):793–807, July 1986.

[4] Leo Breiman. 'bagging predictors'. *Machine Learning*, 24:123–140, 1996.

[5] William Brock, Josef Lakonishok, and Blake LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5):1731–1764, 1992.

[6] John Y. Campbell, Andrew W. Lo, and A. Craig MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997.

[7] John L. Casti. *Searching for Certainty*. William Morrow, 1990.

[8] Darrell Duffie. *Dynamic Asset Pricing Theory*. Princeton University Press, 1996.

[9] Bradley Efron and Robert J. Tibshrani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[10] Eugene Fama. Market efficiency, long-term returns, and behavioral finance. Technical report, University of Chicago Business School, 1997.

[11] Eugene F. Fama. Efficient capital markets: II. *Journal of Finance*, 46, 1991.

[12] G. Jarrell and A. Poulsen. The returns to acquiring firms in tender' offers: Evidence from three decades. *Financial Management*, 18:12–19, 1989.

[13] Blake LeBaron. Technical trading rule profitability and foreign exchange intervention. *Journal of International Economics*, forthcoming 1998.

[14] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[15] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[16] Chris Neely, Paul Weller, and Rob Dittmar. Is technical analysis in the foreign exchange market profitable? a genetic programming approach. Technical report, Federal Reserve Bank of St. Louis, 1997.

[17] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[18] C. L. Osler and P. H. Kevin Chang. Head and shoulders: Not just a flaky pattern. Technical report, 1995.

[19] Martin J Pring. *Technical Analysis Explained*. McGraw Hill, 1991.

[20] Robert E. Shapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1–5, 1999.

[21] Ryan Sullivan, Allan Timmermann, and Halbert White. Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 1998.

[22] Richard H. Thaler. *Advances in Behavioral Finance*. Russell Sage Foundation, 1993.

[23] James D Thomas and Katia Sycara. The importance of simplicity and validation in genetic programming for data mining in financial data. In *Proceedings of the joint GECCO-99 and AAAI-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions*, 1999.

[24] Andreas S. Weigand and Neil A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison Wesley, 1994.

[25] Andreas S. Weigend, Yaser Abu-Mostafa, and A. Paul N. Refenes, editors. *Decision Technologies for Financial Engineering: Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets*. World Scientific, 1997.

[26] B. Wüthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang, and W. Lam. Daily prediction of major stock indices from textual www data. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining - KDD-98*, 1998.