Learning to Trade with Insider Information

Sanmay Das

AI Memo 2005-028
CBCL Memo 255

October 2005

Learning to Trade with Insider Information

Sanmay Das
Center for Biological and Computational Learning and
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
sanmay@mit.edu

October 7, 2005

Abstract

This paper introduces algorithms for learning how to trade using insider (superior) information in Kyle's model of financial markets. Prior results in finance theory relied on the insider having perfect knowledge of the structure and parameters of the market. I show here that it is possible to learn the equilibrium trading strategy when its form is known even without knowledge of the parameters governing trading in the model. However, the rate of convergence to equilibrium is slow, and an approximate algorithm that does not converge to the equilibrium strategy achieves better utility when the horizon is limited. I analyze this approximate algorithm from the perspective of reinforcement learning and discuss the importance of domain knowledge in designing a successful learning algorithm.

1 Introduction

In financial markets, information is revealed by trading. Once private information is fully disseminated to the public, prices reflect all available information and reach market equilibrium. Before prices reach equilibrium, agents with superior information have opportunities to gain profits by trading. This paper focuses on the design of a general algorithm that allows an agent to learn how to exploit superior or "insider" information.[1] Suppose a trading agent receives a signal of what price a stock will trade at n trading periods from now. What is the best way to exploit this information in terms of placing trades in each of the intermediate periods? The agent has to make a tradeoff between the profit made from an immediate trade

[1] The term "insider" information has negative connotations in popular belief. I use the term solely to refer to superior information, however it may be obtained (for example, paying for an analyst's report on a firm can be viewed as a way of obtaining insider information about a stock).

and the amount of information that trade reveals to the market. If the stock is undervalued it makes sense to buy some stock, but buying too much may reveal the insider's information too early and drive the price up, relatively disadvantaging the insider.

This problem has been studied extensively in the finance literature, initially in the context of a trader with monopolistic insider information [6], and later in the context of competing insiders with homogeneous [4] and heterogeneous [3] information.[2] All these models derive equilibria under the assumption that traders are perfectly informed about the structure and parameters of the world in which they trade. For example, in Kyle's model, the informed trader knows two important distributions — the ex ante distribution of the liquidation value and the distribution of other ("noise") trades that occur in each period.

In this paper, I start from Kyle's original model [6], in which the trading process is structured as a sequential auction at the end of which the stock is liquidated. An informed trader or "insider" is told the liquidation value some number of periods before the liquidation date, and must decide how to allocate trades in each of the intervening periods. There is also some amount of uninformed trading (modeled as white noise) at each period. The clearing price at each auction is set by a market-maker who sees only the combined order flow (from both the insider and the noise traders) and seeks to set a zero-profit price. In the next section I discuss the importance of this problem from the perspectives of research both in finance and in reinforcement learning. In sections 3 and 4 I introduce the market model and two learning algorithms, and in Section 5 I present experimental results. Finally, Section 6 concludes and discusses future research directions.

2 Motivation: Bounded Rationality and Reinforcement Learning

One of the arguments for the standard economic model of a decision-making agent as an unboundedly rational optimizer is the argument from learning. In a survey of the bounded rationality literature, John Conlisk lists this as the second among eight arguments typically used to make the case for unbounded rationality [2]. To paraphrase his description of the argument, it is all right to assume unbounded rationality because agents learn optima through practice. Commenting on this argument, Conlisk says "learning is promoted by favorable conditions such as rewards, repeated opportunities for practice, small deliberation cost at each repetition, good feedback, unchanging circumstances, and a simple context." The learning process must be analyzed in terms of these issues to see if it will indeed lead to agent behavior that is optimal and to see how differences in the environment can affect the learning process. The design of a successful learning algorithm for agents who are not necessarily aware of who else has inside information or what the price formation process is could elucidate the conditions that are necessary for agents to arrive at equilibrium, and could potentially lead to characterizations of alternative equilibria in these models.

[2] My discussion of finance models in this paper draws directly from these original papers and from the survey by O'Hara [8].

One way of approaching the problem of learning how to trade in the framework developed here is to apply a standard reinforcement learning algorithm with function approximation. Fundamentally, the problem posed here has infinite (continuous) state and action spaces (prices and quantities are treated as real numbers), which pose hard challenges for reinforcement learning algorithms. However, reinforcement learning has worked in various complex domains, perhaps most famously in backgammon [11] (see Sutton and Barto for a summary of some of the work on value function approximation [10]). There are two key differences between these successes and the problem studied here that make it difficult for the standard methodology to be successful without properly tailoring the learning algorithm to incorporate important domain knowledge.

First, successful applications of reinforcement learning with continuous state and action spaces usually require the presence of an offline simulator that can give the algorithm access to many examples in a costless manner. The environment envisioned here is intrinsically online — the agent interacts with the environment by making potentially costly trading decisions which actually affect the payoff it receives. In addition to this, the agent wants to minimize exploration cost because it is an active participant in the economic environment. Achieving a high utility from early on in the learning process is important to agents in such environments. Second, the sequential nature of the auctions complicates the learning problem. If we were to try and model the process in terms of a Markov decision problem (MDP), each state would have to be characterized not just by traditional state variables (in this case, for example, last traded price and liquidation value of a stock) but by how many auctions in total there are, and which of these auctions is the current one. The optimal behavior of a trader at the fourth auction out of five is different from the optimal behavior at the second auction out of ten, or even the ninth auction out of ten. While including the current auction and total number of auctions as part of the state would allow us to represent the problem as an MDP, it would not be particularly helpful because the generalization ability from one state to another would be poor. This problem might be mitigated in circumstances where the optimal behavior does not change much from auction to auction, and characterizing these circumstances is important. In fact, I describe an algorithm below that uses a representation where the current auction and the total number of auctions do not factor into the decision. This approach is very similar to model based reinforcement learning with value function approximation, but the main reason why it works very well in this case is that we understand the form of the optimal strategy, so the representations of the value function, state space, and transition model can be tailored so that the algorithm performs close to optimally. I discuss this in more detail in Section 5.

An alternative approach to the standard reinforcement learning methodology is to use explicit knowledge of the domain and learn separate functions for each auction. The learning process receives feedback in terms of actual profits received for each auction from the current one onwards, so this is a form of direct utility estimation [12]. While this approach is related to the direct-reinforcement learning method of Moody and Saffell [7], the problem studied here involves more consideration of delayed rewards, so it is necessary to learn something equivalent to a value function in order to optimize the total reward.

The important domain facts that help in the development of a learning algorithm are based on Kyle's results. Kyle proves that in equilibrium, the expected future profits from auction i onwards are a linear function of the square difference between the liquidation value and the last traded price (the actual linear function is different for each i). He also proves that the next traded price is a linear function of the amount traded. These two results are the key to the learning algorithm. I will show in later sections that the algorithm can learn from a small amount of randomized training data and then select the optimal actions according to the trader's beliefs at every time period. With a small number of auctions, the learning rule enables the trader to converge to the optimal strategy. With a larger number of auctions the number of episodes required to reach the optimal strategy becomes impractical and an approximate mechanism achieves better results. In all cases the trader continues to receive a high flow utility from early episodes onwards.

3 Market Model

The model is based on Kyle's original model [6]. There is a single security which is traded in N sequential auctions. The liquidation value v of the security is realized after the Nth auction, and all holdings are liquidated at that time. v is drawn from a Gaussian distribution with mean p_0 and variance Σ_0, which are common knowledge. Here we assume that the N auctions are identical and distributed evenly in time. An informed trader or insider observes v in advance and chooses an amount to trade Δx_i at each auction $i \in \{1, \ldots, N\}$. There is also an uninformed order flow amount Δu_i at each period, sampled from a Gaussian distribution with mean 0 and variance $\sigma_u^2 \Delta t_i$ where $\Delta t_i = 1/N$ for our purposes (more generally, it represents the time interval between two auctions).[3] The trading process is mediated by a market-maker who absorbs the order flow while earning zero expected profits. The market-maker only sees the combined order flow $\Delta x_i + \Delta u_i$ at each auction and sets the clearing price p_i. The zero expected profit condition can be expected to arise from competition between market-makers.

Equilibrium in the monopolistic insider case is defined by a profit maximization condition on the insider which says that the insider optimizes overall profit given available information, and a market efficiency condition on the (zero-profit) market-maker saying that the market-maker sets the price at each auction to the expected liquidation value of the stock given the combined order flow.

Formally, let π_i denote the profits made by the insider on positions acquired from the ith auction onwards. Then $\pi_i = \sum_{k=i}^{N} (v - p_k)\Delta x_k$. Suppose that X is the insider's trading strategy and is a function of all information available to her, and P is the market-maker's pricing rule and is again a function of available information. X_i is a mapping from $(p_1, p_2, \ldots, p_{i-1}, v)$ to x_i where x_i represents the insider's total holdings after auction i (from which Δx_i can be

[3] The motivation for this formulation is to allow the representative uninformed trader's holdings over time to be a Brownian motion with instantaneous variance σ_u^2. The amount traded represents the change in holdings over the interval.

calculated). P_i is a mapping from $(x_1 + u_1, \ldots, x_i + u_i)$ to p_i. X and P consist of all the component X_i and P_i. Kyle defines the sequential auction equilibrium as a pair X and P such that the following two conditions hold:

1. *Profit maximization*: For all $i = 1, \ldots, N$ and all X':

$$E[\pi_i(X, P)|p_1, \ldots, p_{i-1}, v] \geq E[\pi_i(X', P)|p_1, \ldots, p_{i-1}, v]$$

2. *Market efficiency*: For all $i = 1, \ldots, N$, $p_i = E[v|x_1 + u_1, \ldots, x_i + u_i]$

The first condition ensures that the insider's strategy is optimal, while the second ensures that the market-maker plays the competitive equilibrium (zero-profit) strategy. Kyle also shows that there is a unique linear equilibrium [6].

Theorem 1 (Kyle, 1985). *There exists a unique linear (recursive) equilibrium in which there are constants $\beta_n, \lambda_n, \alpha_n, \delta_n, \Sigma_n$ such that for:*

$$\begin{aligned}
\Delta x_n &= \beta_n(v - p_{n-1})\Delta t_n \\
\Delta p_n &= \lambda_n(\Delta x_n + \Delta u_n) \\
\Sigma_n &= var(v|\Delta x_1 + \Delta u_1, \ldots, \Delta x_n + \Delta u_n) \\
E[\pi_n|p_1, \ldots, p_{n-1}, v] &= \alpha_{n-1}(v - p_{n-1})^2 + \delta_{n-1}
\end{aligned}$$

Given Σ_0 the constants $\beta_n, \lambda_n, \alpha_n, \delta_n, \Sigma_n$ are the unique solution to the difference equation system:

$$\begin{aligned}
\alpha_{n-1} &= \frac{1}{4\lambda_n(1 - \alpha_n\lambda_n)} \\
\delta_{n-1} &= \delta_n + \alpha_n\lambda_n^2\sigma_u^2\Delta t_n \\
\beta_n\Delta t_n &= \frac{1 - 2\alpha_n\lambda_n}{2\lambda_n(1 - \alpha_n\lambda_n)} \\
\lambda_n &= \beta_n\Sigma_n/\sigma_u^2 \\
\Sigma_n &= (1 - \beta_n\lambda_n\Delta t_n)\Sigma_{n-1}
\end{aligned}$$

subject to $\alpha_N = \delta_N = 0$ and the second order condition $\lambda_n(1 - \alpha_n\lambda_n) = 0$.[4]

The two facts about the linear equilibrium that will be especially important for learning are that there exist constants $\lambda_i, \alpha_i, \delta_i$ such that:

$$\Delta p_i = \lambda_i(\Delta x_i + \Delta u_i) \tag{1}$$

$$E[\pi_i|p_1, \ldots, p_{i-1}, v] = \alpha_{i-1}(v - p_{i-1})^2 + \delta_{i-1} \tag{2}$$

[4] The second order condition rules out a situation in which the insider can make unbounded profits by first destabilizing prices with unprofitable trades.

Perhaps the most important result of Kyle's characterization of equilibrium is that the insider's information is incorporated into prices gradually, and the optimal action for the informed trader is not to trade particularly aggressively at earlier dates, but instead to hold on to some of the information. In the limit as $N \to \infty$ the rate of revelation of information actually becomes constant. Also note that the market-maker imputes a strategy to the informed trader without actually observing her behavior, only the order flow.

4 A Learning Model

4.1 The Learning Problem

I am interested in examining a scenario in which the informed trader knows very little about the structure of the world, but must learn how to trade using the superior information she possesses. I assume that the price-setting market-maker follows the strategy defined by the Kyle equilibrium. This is justifiable because the market-maker (as a specialist in the New York Stock Exchange sense [9]) is typically in an institutionally privileged situation with respect to the market and has also observed the order-flow over a long period of time. It is reasonable to conclude that the market-maker will have developed a good domain theory over time.

The problem faced by the insider is similar to the standard reinforcement learning model [5, 1, 10] in which an agent does not have complete domain knowledge, but is instead placed in an environment in which it must interact by taking actions in order to gain reinforcement. In this model the actions an agent takes are the trades it places, and the reinforcement corresponds to the profits it receives. The informed trader makes no assumptions about the market-maker's pricing function or the distribution of noise trading, but instead tries to maximize profit over the course of each sequential auction while also learning the appropriate functions.

4.2 A Learning Algorithm

At each auction i the goal of the insider is to maximize

$$\pi_i = \Delta x_i (v - p_i) + \pi_{i+1} \tag{3}$$

The insider must learn both p_i and π_{i+1} as functions of the available information. We know that in equilibrium p_i is a linear function of p_{i-1} and Δx_i, while π_{i+1} is a linear function of $(v - p_i)^2$. This suggests that an insider could learn a good representation of next price and future profit based on these parameters. In this model, the insider tries to learn parameters a_1, a_2, b_1, b_2, b_3 such that:

$$p_i = b_1 p_{i-1} + b_2 \Delta x_i + b_3 \tag{4}$$

$$\pi_{i+1} = a_1 (v - p_i)^2 + a_2 \tag{5}$$

These equations are applicable for all periods except the last, since p_{N+1} is undefined, but we know that $\pi_{N+1} = 0$. From this we get:

$$\pi_i = \Delta x_i(v - b_1 p_{i-1} - b_2 \Delta x_i - b_3) + a_1(v - b_1 p_{i-1} - b_2 \Delta x_i - b_3)^2 + a_2 \tag{6}$$

The profit is maximized when the partial derivative with respect to the amount traded is 0. Setting $\frac{\partial \pi_i}{\partial(\Delta x_i)} = 0$:

$$\Delta x_i = \frac{-v + b_1 p_{i-1} + b_3 + 2a_1 b_2(v - b_1 p_{i-1} - b_3)}{2a_1 b_2^2 - 2b_2} \tag{7}$$

Now consider a repeated sequential auction game where each *episode* consists of N auctions. Initially the trader trades randomly for a particular number of episodes, gathering data as she does so, and then performs a linear regression on the stored data to estimate the five parameters above *for each auction.* The trader then updates the parameters periodically by considering all the observed data (see Algorithm 1 for pseudocode). The trader trades optimally according to her beliefs at each point in time, and any trade provides information on the parameters, since the price change is a noisy linear function of the amount traded. There may be benefits to sometimes not trading optimally in order to learn more. This becomes a problem of both active learning (choosing a good Δx to learn more, and a problem of balancing exploration and exploitation.

```
Data: $T$: total number of episodes, $N$: number of auctions, $K$: number of initialization
      episodes, $D[i][j]$: data from episode $i$, auction $j$, $F_j$: estimated parameters for
      auction $j$
for $i = 1 : K$ do
    for $j = 1 : N$ do
        Choose random trading amount, save data in $D[i][j]$
    end
end
for $j = 1 : N$ do
    Estimate $F_j$ by regressing on $D[1][j] \dots D[K][j]$
end
for $i = K + 1 : T$ do
    for $j = 1 : N$ do
        Choose trading amount based on $F_j$, save data in $D[i][j]$
    end
    if $i \mod 5 = 0$ then
        for $j = 1 : N$ do
            Estimate $F_j$ by regressing on $D[1][j] \dots D[i][j]$
        end
    end
end
```

Algorithm 1: The equilibrium learning algorithm

4.3 An Approximate Algorithm

An alternative algorithm would be to use the same parameters for each auction, instead of estimating separate a's and b's for each auction (see Algorithm 2). Essentially, this algorithm is a learning algorithm which characterizes the state entirely by the last traded price and the liquidation price, irrespective of the particular auction number or even the total number of auctions. The value function of a state is given by the expected profit, which we know from equation 6. We can solve for the optimal action based on our knowledge of the system. In the last auction before liquidation, the insider trades knowing that this is the last auction, and does not take future expected profit into account, simply maximizing the expected value of that trade.

Stating this more explicitly in terms of standard reinforcement learning terminology, the insider assumes that the world is characterized by the following.

- A continuous state space where the state is $v - p$, where p is the last traded price.
- A continuous action space where actions are given by Δx, the amount the insider chooses to trade.
- A stochastic transition model mapping p and Δx to p' (v is assumed constant during an episode). The model is that p' is a (noisy) linear function of Δx and p.
- A (linear) value function mapping $(v - p)^2$ to π, the expected profit.

In addition, the agent knows at the last auction of an episode that the expected future profit from the next stage onwards is 0.

Of course, the world does not really conform exactly to the agent's model. One important problem that arises because of this is that the agent does not take into account the difference between the optimal way of trading at different auctions. The great advantage is that the agent should be able to learn with considerably less data and perhaps do a better job of maximizing finite-horizon utility. Further, if the parameters are not very different from auction to auction this algorithm should be able to find a good approximation of the optimal strategy. Even if the parameters are considerably different for some auctions, if the expected difference between the liquidation value and the last traded price is not high at those auctions, the algorithm might learn a close-to-optimal strategy. The next section discusses the performance of these algorithms, and analyzes the conditions for their success. I will refer to the first algorithm as the equilibrium learning algorithm and to the second as the approximate learning algorithm in what follows.

5 Experimental Results

5.1 Experimental Setup

To determine the behavior of the two learning algorithms, it is important to compare their behavior with the behavior of the optimal strategy under perfect information. In order to

```
Data: $T$: total number of episodes, $N$: number of auctions, $K$: number of initialization
      episodes, $D[i][j]$: data from episode $i$, auction $j$, $F$: estimated parameters
for $i = 1 : K$ do
    for $j = 1 : N$ do
        Choose random trading amount, save data in $D[i][j]$
    end
end
Estimate $F$ by regressing on $D[1][] \ldots D[K][]$ for $i = K + 1 : T$ do
    for $j = 1 : N$ do
        Choose trading amount based on $F$, save data in $D[i][j]$
    end
    if $i \mod 5 = 0$ then
        Estimate $F$ by regressing on $D[1][] \ldots D[i][]$
    end
end
```

Algorithm 2: The approximate learning algorithm

elucidate the general properties of these algorithms, this section reports experimental results when there are 4 auctions per episode. For the equilibrium learning algorithm the insider trades randomly for 50 episodes, while for the approximate algorithm the insider trades randomly for 10 episodes, since it needs less data to form a somewhat reasonable initial estimate of the parameters.[5] In both cases, the amount traded at auction i is randomly sampled from a Gaussian distribution with mean 0 and variance $100/N$ (where N is the number of auctions per episode). Each simulation trial runs for 40,000 episodes in total, and all reported experiments are averaged over 100 trials. The actual parameter values, unless otherwise specified, are $p_0 = 75, \Sigma_0 = 25, \sigma_u^2 = 25$ (the units are arbitrary). The market-maker and the optimal insider (used for comparison purposes) are assumed to know these values and solve the Kyle difference equation system to find out the parameter values they use in making price-setting and trading decisions respectively.

5.2 Main Results

Figure 1 shows the average absolute value of the quantity traded by an insider as a function of the number of episodes that have passed. The graphs show that a learning agent using the equilibrium learning algorithm appears to be slowly converging to the equilibrium strategy in the game with four auctions per episode, while the approximate learning algorithm converges quickly to a strategy that is not the optimal strategy. Figure 2 shows two important facts. First, the graph on the left shows that the average profit made rises much more sharply for the approximate algorithm, which makes better use of available data. Second, the graph on the right shows that the average total utility being received is higher from episode 20,000 onwards for the equilibrium learner (all differences between the algorithms

[5] This setting does not affect the long term outcome significantly unless the agent starts off with terrible initial estimates.

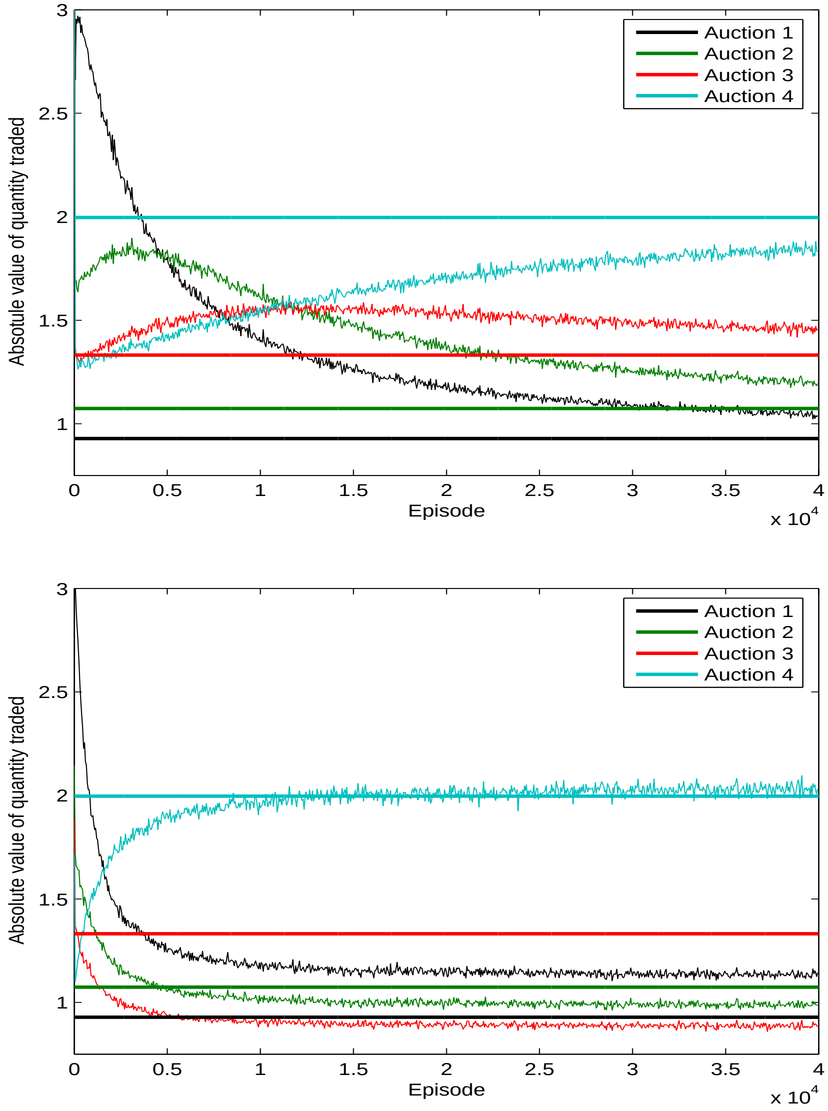
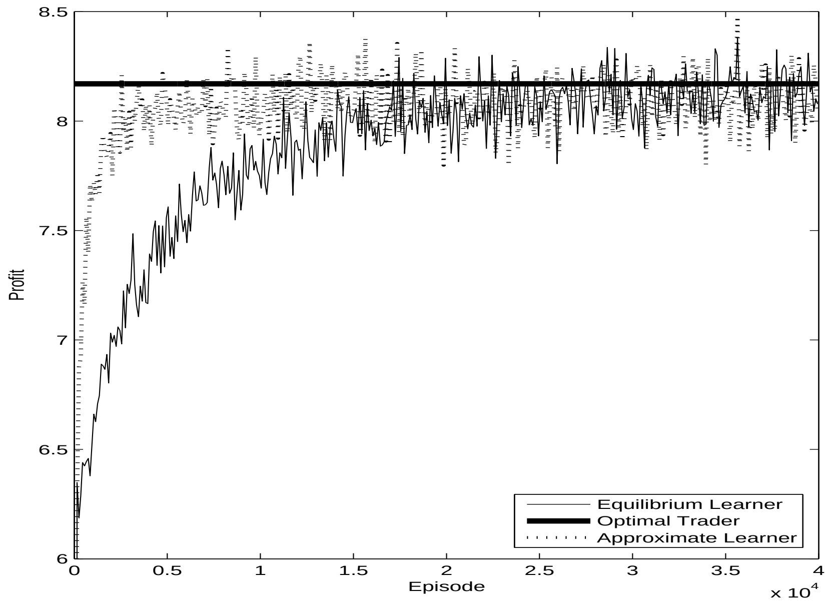


Figure 1: Average absolute value of quantities traded at each auction by a trader using the equilibrium learning algorithm (above) and a trader using the approximate learning algorithm (below) as the number of episodes increases. The thick lines parallel to the X axis represent the average absolute value of the quantity that an optimal insider with full information would trade.



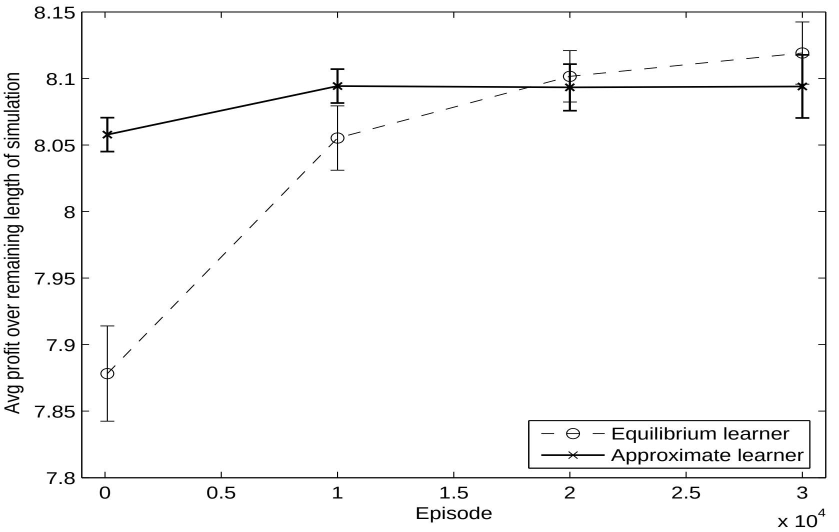


Figure 2: Above: Average flow profit recieved by traders using the two learning algorithms (each point is an aggregate of 50 episodes over all 100 trials) as the number of episodes increases. Below: Average profit received until the end of the simulation measured as a function of the episode from which we start measuring (for episodes 100, 10,000, 20,000 and 30,000).

in this graph are statistically significant at a 95% level). Were the simulations to run long enough, the equilibrium learner would outperform the approximate learner in terms of total utility received, but this would require a huge number of episodes per trial.

Clearly, there is a tradeoff between achieving a higher flow utility and learning a representation that allows the agent to trade optimally in the limit. This problem is exacerbated as the number of auctions increases. With 10 auctions per episode, an agent using the equilibrium learning algorithm actually does not learn to trade more heavily in auction 10 than she did in early episodes even after 40,000 total episodes, leading to a comparatively poor average profit over the course of the simulation. This is due to the dynamics of learning in this setting. The opportunity to make profits by trading heavily in the last auction are highly dependent on not having traded heavily earlier, and so an agent cannot learn a policy that allows her to trade heavily at the last auction until she learns to trade less heavily earlier. This takes more time when there are more auctions. It is also worth noting that assuming that agents have a large amount of time to learn in real markets is unrealistic.

The graphs in Figures 1 and 2 reveal some interesting dynamics of the learning process. First, with the equilibrium learning algorithm, the average profit made by the agent slowly increases in a fairly smooth manner with the number of episodes, showing that the agent's policy is constantly improving as she learns more. An agent using the approximate learning algorithm shows much quicker learning, but learns a policy that is not asymptotically optimal. The second interesting point is about the dynamics of trader behavior — under both algorithms, an insider initially trades far more heavily in the first period than would be considered optimal, but slowly learns to hide her information like an optimal trader would. For the equilibrium learning algorithm, there is a spike in the amount traded in the second period early on in the learning process. This is also a small spike in the amount traded in the third period before the agent starts converging to the optimal strategy.

5.3 Analysis of the Approximate Algorithm

The behavior of the trader using the approximate algorithm is interesting in a variety of ways. First, let us consider the pattern of trades in Figure 1. As mentioned above, the trader trades more aggressively in period 1 than in period 2, and more aggressively in period 2 than in period 3. Let us analyze why this is the case. The agent is learning a strategy that makes the same decisions independent of the particular auction number (except for the last auction). At any auction other than the last, the agent is trying to choose Δx to maximize:

$$\Delta x(v - p') + W[S_{v,p'}]$$

where p' is the next price (also a function of Δx, and also taken to be independent of the particular auction) and $W[S_{v,p'}]$ is the value of being in the state characterized by the liquidation value v and (last) price p'. The agent also believes that the price p' is a linear function of p and Δx. There are two possibilities for the kind of behavior the agent might exhibit, given that she knows that her action will move the stock price in the direction of her trade (if she buys, the price will go up, and if she sells the price will go down). She could try

From episode	$\Sigma_0 = 5, \sigma_u^2 = 25$		$\Sigma_0 = 5, \sigma_u^2 = 50$		$\Sigma_0 = 10, \sigma_u^2 = 25$	
	Approx	Equil	Approx	Equil	Approx	Equil
100	0.986	0.964	0.986	0.983	0.986	0.964
10,000	0.991	0.986	0.990	0.997	0.990	0.986
20,000	0.991	0.992	0.990	0.999	0.989	0.992
30,000	0.991	0.994	0.989	1.000	0.989	0.994

Table 1: Proportion of optimal profit received by traders using the approximate and the equilibrium learning algorithm in domains with different parameter settings. The leftmost column indicates the episode from which measurement starts, running through the end of the simulation (40,000 periods).

to trade *against* her signal, because the model she has learned suggests that the potential for future profit gained by pushing the price away from the direction of the true liquidation value is higher than the loss from the one trade.[6] The other possibility is that she trades *with* her signal. In this case, the similarity of auctions in the representation ensures that she trades with an intensity proportional to her signal. Since she is trading in the correct direction, the price will move (in expectation) towards the liquidation value with each trade, and the average amount traded will go down with each successive auction. The difference in the last period, of course, is that the trader is solely trying to maximize $\Delta x(v - p')$ because she knows that it is her last opportunity to trade.

The success of the algorithm when there are as few as four auctions demonstrates that learning an approximate representation of the underlying model can be very successful in this setting as long as the trader behaves differently at the last auction. Another important question is that of how parameter choice affects the profit-making performance of the approximate algorithm as compared to the equilibrium learning algorithm. In order to study this question, I conducted experiments that measured the average profit received when measurement starts at various different points for a few different parameter settings (this is the same as the second experiment in Figure 2). The results are shown in Table 1. These results demonstrate especially that the profit-making behavior of the equilibrium learning algorithm is somewhat variable across parameter settings while the behavior of the approximate algorithm is remarkably consistent. The advantage of using the approximate algorithms will obviously be greater in settings where the equilibrium learner takes a longer time to start making near-optimal profits. From these results, it seems that the equilibrium learning algorithm learns more quickly in settings with higher liquidity in the market.

[6] This is not really learnable using linear representations for everything unless there is a different function that takes over at some point (such as the last auction), because otherwise the trader would keep trading in the wrong direction and never receive positive reinforcement.

6 Conclusions and Future Work

This paper presents two algorithms that allow an agent to learn how to exploit monopolistic insider information in securities markets when agents do not possess full knowledge of the parameters characterizing the environment, and compares the behavior of these algorithms to the behavior of the optimal algorithm with full information. The results presented here demonstrate how domain knowledge can be very useful in the design of algorithms that learn from experience in an intrinsically online setting in which standard reinforcement learning techniques are hard to apply.

It would be interesting to examine the behavior of the approximate learning algorithm in market environments that are not necessarily generated by an underlying linear mechanism. For example, if many traders are trading in a double auction type market, would it still make sense for a trader to use an algorithm like the approximate one presented here in order to maximize profits from insider information?

I would also like to investigate what differences in market properties are predicted by the learning model as opposed to Kyle's model. Another direction for future research is the use of an online learning algorithm. Batch regression can become prohibitively expensive as the total number of episodes increases. While one alternative is to use a fixed window of past experience, hence forgetting the past, another plausible alternative is to use an online algorithm that updates the agent's beliefs at each time step, throwing away the example after the update. Under what conditions do online algorithms converge to the equilibrium? Are there practical benefits to the use of these methods?

Perhaps the most interesting direction for future research is the multi-agent learning problem. First, what if there is more than one insider and they are all learning?[7] Insiders could potentially enter or leave the market at different times, but we are no longer guaranteed that everyone other than one agent is playing the equilibrium strategy. What are the learning dynamics? What does this imply for the system as a whole? Another point is that the presence of suboptimal insiders ought to create incentives for market-makers to deviate from the complete-information equilibrium strategy in order to make profits. What can we say about the learning process when both market-makers and insiders may be learning?

Acknowledgements

I would like to thank Leslie Kaelbling, Adlar Kim, Andrew Lo, Tommy Poggio and Tarun Ramadorai for helpful discussions and suggestions. I also acknowledge grants to CBCL from Merrill-Lynch, the National Science Foundation, the Center for e-Business at MIT, the Eastman Kodak Company, Honda R&D Co, and Siemens Corporate Research, Inc.

[7] Theoretical results show that equilibrium behavior with complete information is of the same linear form as in the monopolistic case [4, 3].

References

[1] Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[2] John Conlisk. Why bounded rationality? *Journal of Economic Literature*, 34(2):669–700, 1996.

[3] F.D. Foster and S. Viswanathan. Strategic trading when agents forecast the forecasts of others. *The Journal of Finance*, 51:1437–1478, 1996.

[4] C.W. Holden and A. Subrahmanyam. Long-lived private information and imperfect competition. *The Journal of Finance*, 47:247–270, 1992.

[5] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[6] Albert S. Kyle. Continuous auctions and insider trading. *Econometrica*, 53(6):1315–1336, 1985.

[7] John Moody and Matthew Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, 2001.

[8] M. O'Hara. *Market Microstructure Theory.* Blackwell, Malden, MA, 1995.

[9] Robert A. Schwartz. *Reshaping the Equity Markets: A Guide for the 1990s.* Harper Business, New York, NY, 1991.

[10] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[11] Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, March 1995.

[12] B. Widrow and M.E. Hoff. Adaptive switching circuits. In *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4*, pages 96–104, 1960.