Working Notes of the AAAI-96 Workshop on Agent Modeling, 90–97, 1996.

Building Agent Models in Economic Societies of Agents

José M. Vidal and Edmund H. Durfee
Artificial Intelligence Laboratory, University of Michigan
1101 Beal Avenue, Ann Arbor, MI 48109-2110
jmvidal@umich.edu

Abstract

We build an economic society of agents in which buyers and sellers compete with each other, and try to increase their total values and total profits, respectively. We give a precise method for the construction of these agents and for the incremental incorporation of modeling capabilities, following the intuitions behind the Recursive Modeling Method (RMM). These agents were built and run in a simulated economic society. Early test results show that deeper (i.e. 1-level) models are more effective in heterogeneous societies than in homogeneous ones, and that price volatility is a good predictor of the relative benefits of deeper models.

1 Introduction

We all know that, in order to win most competitions, an agent need only be somewhat better than everyone else. If the competition in question only involves deal-making and economic exchange, and given a level playing field for all the competitors, then it seems clear that the winners will be those who can "outsmart" the rest. It is an agent's *relative* "smartness", with respect to the other agents, that is important in this case. Of course, since there are usually other costs associated with being smart, the question then becomes, when does it pay to be smart?

In this paper, we build an economic society of agents in which buyers and sellers compete with each other, and try to increase their total values and total profits, respectively. We give a precise method for the construction of these agents and for the incremental incorporation of modeling capabilities, following the intuitions behind the Recursive Modeling Method (RMM)[7]. The increased modeling capabilities allow agents to represent and use more knowledge about each other, hence increasing their "smartness". By building these agents and running them in a simulated economic society, we hope to provide a better characterization of where and when each of the different modeling levels works better.

We start with a description of the economic society and then describe how the agents make their decisions and build their models. There are agents with no models (0-level agents) that are unaware of the existence of other agents in the world, agents with sub-intentional models (1-level agents) that keep models of others, and agents with intentional models (2-level agents) that have deeper models of others. Populations of agents with 0-level and 1-level models have been tested and the results and lessons learned are presented in Sections 5 and 6. The main results are the correlation of price volatility to profit gains for the 1-level modeling agents, and our analysis of the "equilibrium" solution— how it can be derived from a population description and how its characteristics affect volatility and, therefore, the performance of 1-level agents.

2 Description of Agents and Society

We define an economic society of agents as one where each agent is either a *buyer* or a *seller*. The set of buyers is B and the set of sellers is S. These agents exchange goods by paying some price $p \in P$, where P is a finite set. The buyers are capable of assessing the quality of a good received and giving it some value $q \in Q$, where Q is also a finite set.

The exchange protocol, seen in Figure 1, works as follows: When a buyer $b \in B$ wants to buy a good g, she will advertise this fact. Each seller $s \in S$ that sells that good will give his bid in the form of a price p_s^g. The buyer will pick one of these and will pay the seller. The seller will then return[1] the specified good. Note that there is no law that forces the seller to return a good of any quality. It is up to the buyer to assess the quality q of the good.

[1] In the case of agent/link failure, each agent is free to set its own timeouts and assess the quality of the never-received good accordingly. Bids that are not received in time will, of course, not be considered.

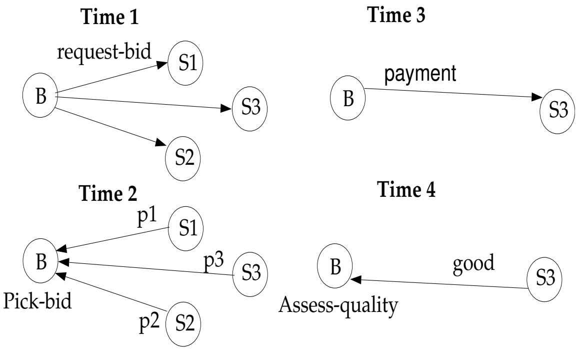

Figure 1: View of the protocol. We show only one buyer B and three sellers $S1$, $S2$, and $S3$. At time 1 the buyer requests bids for some good. At time 2 the sellers send their prices for that good. At time 3 the buyer picks one of the bids, pays the seller the amount and then, at time 4, she receives the good.

Each buyer b also has a value function for each good $g \in G$ that it might wish to buy. The value function, $V_b^g(p, q)$ returns a number that represents the value that b assigns to that particular good at that particular price and quality. Each seller $s \in S$, on the other hand, has a cost c_s^g associated with each good it can produce. Therefore, if seller s gets paid p for good g, his profit will be $\text{Profit}(p, c_s^g)$. Since we assume that cost and payments are expressed in the same unit (i.e. money), the profit equation simplifies to $p - c_s^g$. The buyers, therefore, have the goal of maximizing the value they get for their transactions, and the sellers have the goal of maximizing their profits.

2.1 Justifying the model

In a traditional market-based economy it is assumed that the set of goods is known and accepted by all agents. That is, if two objects are instances of the same good (e.g. two apples), then it is assumed that all agents will treat them as completely interchangeable. There is no reason to prefer one over the other, and agents will pay exactly the same for either one. However, while this is a very useful abstraction for some systems, there are many instances where it is an unrealistic assumption. Specifically, information goods/services are very hard to compartmentalize into equivalence classes that all agents can agree on, especially if the agents are software agents who are constantly changing their preferences in order to maximize their profits. For example, if an encyclopedia database access is defined as a good, then all agents providing encyclopedia accesses can be considered as selling the same good. It is likely, however, that a buyer of this good might decide that seller $s1$ provides better answers than seller $s2$. We cannot possibly hope to enumerate the set of reasons an agent might have for preferring one set of answers over another, and we should not try to do so. It should be up to the individual buyers to decide what items belong to the same good category, each buyer clustering items in, possibly, different ways.

This situation is even more evident when we consider an information economy rooted in some information delivery infrastructure (e.g. the Internet). There are two main characteristics that set this economy apart from a traditional economy.

- There is virtually no cost of reproduction. Once the information is created it can be duplicated virtually for free.
- All agents have virtually direct and free access to all other agents.

If these two characteristics are present in an economy, it is useless to talk about supply and demand, since supply is practically infinite for any particular good and available everywhere. The only way agents can survive in such an economy is by providing value-added services that are tailored to meet their customers' needs. Each provider will try to differentiate his goods from everyone else's while each buyer will try to find those suppliers that best meet her value function.

3 Learning recursive models

Agents placed in the economic society we just described will have to learn, via trial and error, what actions give them the highest expected reward and under which circumstances. In this section we will present techniques that these agents might use to maximize their rewards.

An important question we wish to answer is: when do agents benefit from having deeper (i.e. more complex) models of other agents? It should be intuitive that, ignoring computational costs, the agents with more complete models of others will always do better. This seems to be true[2], however, there are instances when it is significantly better to have deeper models, and instances when the difference is barely noticeable. These instances are defined in part by the set of other agents present and their capabilities and preferences. In order to precisely determine what these instances are, and in the hopes of providing a more general framework for studying the effects of increased agent-modeling capabilities within our economic model, we defined a set of techniques that our agents can use for learning and using models.

We divide the agents into classes that correspond to their modeling capabilities. The hierarchy we present is inspired by RMM [6], but is function-based rather than matrix-based, and includes learning. We start with agents with no models (also referred to as 0-level agents), who must base their actions purely on their inputs and the rewards they receive. They are not aware that there are other agents out there. Agents with 1-level models are aware that there are other agents out there but have no idea what the "interior" of these agents looks like. That is, in RMM terminology, they are incapable of ascribing intentions to others. They must make their predictions simply based on the previous actions of the other agents, by building sub-intentional models of others. Agents with 2-level models have intentional models of others (i.e. have models of their beliefs and inference processes) and believe that others keep sub-intentional (i.e. 1-level) models of others. We can similarly keep defining agents of three, four, five-level models, but so far we have concentrated only on the first three levels. In the following sections, we talk about each one of these in more detail and give details about their implementation. Our current theory only considers agents that are either buyers or sellers, but not both.

3.1 Agents with no models

Agents with no models must learn everything they know from observations they make about the environment, and from any rewards they get. In our economic society this means that buyers see the bids they receive and the good received after striking a contract, while sellers see the request for bids and the profit they made (if any). In general, agents get some input, take an

[2] It seems to be true in general. However, as the agent becomes more uncertain of the validity of the deeper models, then the assertion starts to fail. Also, if the market reaches an equilibrium then there are no advantages to thinking, and the best strategy is to become a simple price-taker.

action, then receive some reward. This is the same basic framework under which most learning mechanism are presented. We decided to use a form of reinforcement learning [13] [16] for implementing this kind of learning in our agents, since it is a simple method and the domain is simple enough for it to do a reasonable job.

Both buyers and sellers will use the equations in the next sections for determining what actions to take. However, with a small probability ϵ they will choose to explore, instead of exploit, and will pick their action at random (except for the fact that sellers never bid below cost). The value of ϵ is initially 1 but decreases with time to some empirically chosen, fixed minimum value. That is, $\epsilon_{t+1} = \gamma\epsilon_t$, where $0 < \gamma < 1$ is some annealing factor.

Buyers with no models A buyer b will start by requesting bids for a good g. She will then accept all bids for good g and will pick:

$$s^* = \arg_{s \in S} \max f^g(p_s^g) \tag{1}$$

The function $f^g(p)$ returns the value the buyer expects to get if she buys good g at a price of p. It is learned using a simple form of reinforcement learning, namely:

$$f_{t+1}^g(p) = (1-\alpha) f_t^g(p) + \alpha \cdot V_b^g(p, q) \tag{2}$$

where α is the learning rate, p is the price b paid for the good, and q is the quality she ascribed to it. The learning rate is initially set to 1 and, like ϵ, is decreased until it reaches some fixed minimum value.

Sellers with no models When asked for a bid, the seller s will provide one whose price is greater than or equal[3] to the cost for producing it.

$$p_s^g \geq c_s^g \tag{3}$$

and, from these prices, he will chose the one with the highest expected profit:

$$p_s^* = \arg_{p \in P} \max h_s^g(p) \tag{4}$$

The function $h_s^g(p)$ returns the profit s expects to get if he sells good g at a price p. It is also learned using reinforcement learning, as follows:

$$h_{t+1}^g(p) = (1-\alpha) h_t^g(p) + \alpha \cdot (p - c_s^g) \tag{5}$$

[3] We could just as easily have said that the price must be strictly greater than the cost.

3.2 Agents with One-level Models

The next step is for the agents to keep one-level models of the other agents. This means that it has no idea of what the interior (i.e. "mental") processes of the other agents are, but it recognizes the fact that there are other agents out there whose behaviors influence its rewards. The agent, therefore, can only model others by looking at their past behavior and trying to predict, from it, their future actions.

Buyers with one-level models Each buyer can now keep a history of the qualities it attributes to the goods returned by each seller. She can, in fact, remember the last N qualities returned by some seller s for some good g, and define a probability density function $q_s^g(x)$ over the qualities x returned by s (i.e. $q_s^g(x)$ returns the probability that s returns an instance of good g that has quality x). She can use the expected value of this function to calculate which seller she expects will give her the highest expected value.

$$\begin{aligned} s^* &= \arg_{s \in S} \max E(V_b^g(p_s^g, q_s^g(x))) \\ &= \arg_{s \in S} \max \frac{1}{|Q|} \sum_{x \in Q} q_s^g(x) \cdot V_b^g(p_s^g, x) \end{aligned} \tag{6}$$

The buyer does not need to model other buyers since they do not affect the value she gets.

Sellers with one-level models Each seller will try to predict what bid the other sellers will submit (based solely on what they have bid in the past), and what bid the buyer will likely pick. A complete implementation would require the seller to remember past combinations of buyers, bids and results (i.e. who was buying, who bid what, and who won). However, it is unrealistic to expect a seller to remember all this since there are at least $|P|^{|S|} \cdot |B|$ possible combinations.

We believe, however, that the seller's one-level behavior can be approximated by having him remember the last N prices accepted by each buyer b for each good g, and form a probability density function $m_b^g(x)$, which returns the probability that b will accept(pick) price p for good g. Similarly, the seller remembers other sellers' last N bids for good g and forms $n_s^g(y)$, which gives the probability that s will bid y for good g. The seller s can now determine which bid maximizes his expected profits.

$$p^* = \arg_{p \in P} \max(p - c_s^g) \cdot \prod_{s' \in \{S-s\}} \sum_{p' \in P} \begin{cases} n_{s'}^g(p') & \text{if } m_b^g(p') \leq m_b^g(p) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Note that this function also does a small amount of approximation by assuming that s wins whenever there is a tie[4]. The function calculates the best bid by determining, for each possible bid, the product of the profit and the probability that the agent will get that profit. Since the profit for lost bids is 0, we only need to consider the cases where s wins. The probability that s will win can then be found by calculating the product of the probabilities that his bid will beat the bids of each of the other sellers.

3.3 Agents with Two-level Models

Two level models consist of an intentional model of the agent being modeled (which contains the agent's desires), and the one-level models that the agent being modeled keeps of the other agents (these form part of the agent's beliefs about others). Our intentional models correspond to the procedures or functions used by agents that use one-level models.

Buyers with two-level models Since the buyer receives bids from the sellers, there is no need for her to try to out-guess, or predict what the sellers will bid. She is also not concerned with what the other buyers are doing so, in our current economic model, there does not seem to be any need for the buyer to keep deeper models of others.

Sellers with two-level models He will model other sellers as if they were using the one-level models. That is, he thinks they will model others using policy models and make their decisions using the equations presented in Section 3.2. He will try to predict their bids and then try to find a bid for himself that the buyer will prefer more than all the bids of the other sellers. His model of the buyer will also be an intentional model. He will model the buyers as though they were implemented as explained in Section 3.2.

[4] The complete solution would have to consider the probabilities that s ties with 1, 2, 3, …other agents. In order to do this we would need to consider all $|S|^{|P|}$ subsets.

The algorithm he follows is to first use his models of the sellers to calculate what bids p_i they will submit. He has a model of the buyer $C(s_1 \cdots s_n, p_1 \cdots p_n)$, that tells him which bid she might pick given the set of bids p_i submitted by all sellers s_i. The seller s_j uses this model to determine which of his bids will bring him higher profit, by first finding the set of bids he can make that will win:

$$P' = \{p_j | p_j \in P, C(s_1 \cdots s_j \cdots s_n, p_1 \cdots p_j \cdots p_n) = j\} \tag{8}$$

And from these finding the one with the highest profit:

$$p^* = \arg_{p \in P'} \max(p - c_s^g) \tag{9}$$

4 Equilibrium Analysis

The reader might already have guessed that this model has, in general, no equilibrium solution. However, an equilibrium analysis does help us to predict the main attractors. Let's assume that there is an equilibrium at price p — given this, it must also be true that every seller who can bid at that price does so. That is, those sellers whose cost is strictly greater than p, since all others would make zero or negative profit at this price. For any price p, we can define this set as S_p (omitting the g superscript for clarity):

$$S_p = \{s | s \in S \wedge c_s < p\}$$

The utility each seller s will get at equilibrium price p is

$$U_s(p) = \frac{\text{Profit}}{\text{Number of bids at price } p} = \frac{p - c_s}{|S_p|} \tag{10}$$

We can see that the utility the sellers get increases linearly with price, therefore, a Pareto optimal price equilibrium is quickly ruled out. The buyers' utility function, however, does offer some hope:

$$U_b(p) = V_b(p, \mu_b(p)) \tag{11}$$

Where $\mu_b(p)$ is the average quality buyer b can expect at equilibrium price p. It is calculated from the quality $q_b(s)$ that b expects, on average, to get from seller s.

$$\mu_b(p) = \frac{1}{|S_p|} \sum_{s \in S_p} q_b(s) \tag{12}$$

When the buyers have different value functions then, we can create a utility function that takes the average of all buyers' utilities, weighted by their purchasing rate β_b, where $\sum_{b \in B} \beta_b = 1$.

$$U_B(p) = \sum_{b \in B} \beta_b U_b(p) = \sum_{b \in B} \beta_b V_b(p, \mu_b(p)) \tag{13}$$

This equation will usually have a maximum price:

$$p^* = \arg_{p \in P} \max U_B(p) \tag{14}$$

When p^* exists we can use this "equilibrium" price as the main attractor of the system. We will notice that the prices tend to hover at or just above this price. Unfortunately, this point is not a stable equilibrium in those cases where the buyer gives a higher value to quality than to price (i.e. $k > l$), for some range of prices. In these cases, it is very probable that the value the buyer is getting at p^* is less than the value she would be getting if one (just one) of the sellers offering higher quality, sold her the good at a slightly higher price. Since the agents are always exploring, this is likely to eventually happen. When it does, the buyer will be drawn to this new price because of its high expected value. Eventually, however, the other sellers (with lower quality) will notice this and start selling at the higher price. This will, in turn, lower the value the buyer is getting at this price and she will go back to a lower price.

The sellers are always trying to push the price higher. We find that they are more successful at it when the buyers don't mind so much. That is, let $p^+ = \arg_{p \in \{P - p^*\}} \max U_B(p)$ be the second-best price, then the smaller $p^+ - p^*$ is, the more likely it is that the price will go to p^+. The behavior of the system is expected to be qualitatively different for populations where there is one undisputed p^*, versus those where there is competitions among different prices.

5 Simulations

Since there is no obvious way to analytically determine how different populations of agents would interact and, of greater interest to us, how much better (or worse) the agents with deeper models would fare, we decided to implement a society of the agents described above and ran it to test our hypotheses. We have, so far, concentrated in examining populations that

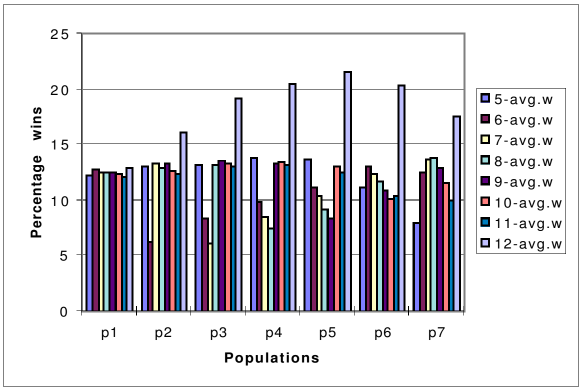

Figure 2: Percentage of times that each of the sellers wins the auctions in the different populations. Seller with 1-level models is #12 (i.e. 12-avg.w). He wins more in all populations, except p1.

combine zero-level buyers (i.e. they have no models) with zero and one-level sellers. In these populations, we vary the number of agents, and change the qualities (as assessed by the buyers) that different sellers return. There is a small amount of random noise (plus or minus one) added to the quality assessment functions as a way to model the uncertainty inherent is assessing quality from just one instance of a good. The value function we used for all the buyers was $V(p,q) = 3q - p$, with $P = \{1,\dots,20\}$ and $Q = \{1,\dots,20\}$. The minimum learning rate α was .1, the minimum explore rate ϵ was .5, the annealing rate was $\gamma = .99$ for both. Slight variations in these parameters do not significantly change the results presented here.

5.1 Some test results

These are the results from tests using seven different agent populations. The results shown are averages over 100 runs, each of 10000 auctions, where an auction is one execution of the protocol shown in Figure 1. All populations have 5 buyers (numbered 0-4) with value function $V(p,q) = 3q - p$ and eight sellers (numbered 5-12), the first 7 have 0-level models and the last seller (#12) has 1-level models. The first population $p1$ has all sellers with $c = q = 2$, $p2$ changes the second seller to have $c = q = 3$, $p3$ further changes the third seller to have $c = q = 4$, and so on until the seventh. The first (#5) and last (#12) sellers always have $c = q = 2$.

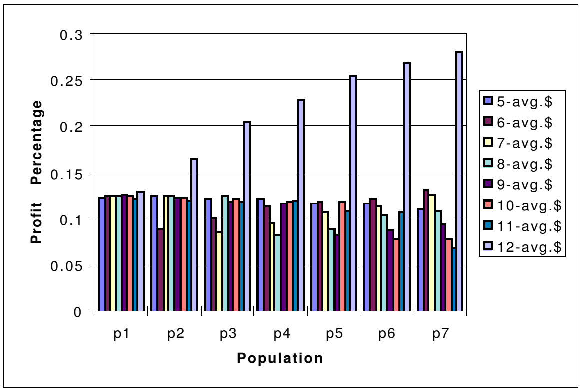

Figure 3: Percentage of the total money that each seller makes (i.e. their profit). Seller with 1-level models is #12 (i.e. 12-avg.w).

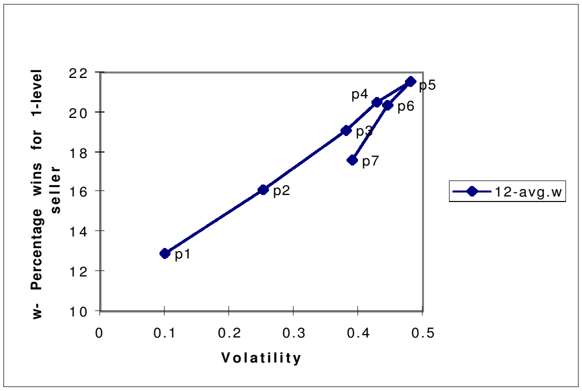

Figure 4: Scatter plot of volatility versus the percentage of time that the 1-level seller wins (w).

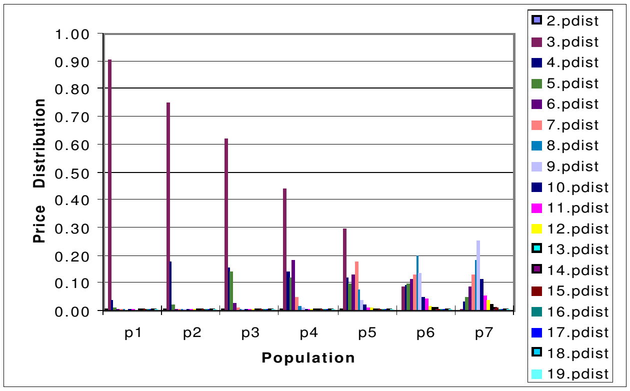

Figure 5: Average price distribution for the different populations. The highest peak in p1–p5 corresponds to price 3.

Figure 2 shows the percentage of times that each seller won the auctions. We can see that the 1-level seller always wins a higher percentage of the time, and that his winning percentage seems to reach a peak in population $p5$ and then decreases. In other words, as the population becomes more heterogeneous his fraction of winning increases until some maximum and then starts to decrease again. The reason why this was happening was not clear to us until we decided to look at the *volatility* of the prices. Still, we must point out, even as the seller loses more of the actions, its profits are still accruing, as shown in Figure 3.

We define volatility as the number of times the price changes over the total number of auctions[5]. Figure 4 shows a scatter plot of the volatility as it is correlated to the percentage of times that the 1-level seller wins (w). We see a line that goes up and reaches a peak at $p5$ and then comes back down, with a slightly different slope. What happens is that populations $p2$ through $p5$ have two competing "equilibrium" prices[6], while $p1$ and $p6-p7$ have only one. We have found that the volatility v is correlated to w differently for these two different populations. That is, if $w = m \cdot v + b$, then m and b take on different values for each population type. Figure 5 shows the price distributions for these runs. It can be seen how, in some populations like

[5] In economics volatility is defined as the actual price changes over the total number of auctions. However, we are not concerned with the magnitude of the changes, only with the actual price movement.

[6] We won't show the math, but this can easily be verified by plugging in the appropriate values into the equations of Section 4.

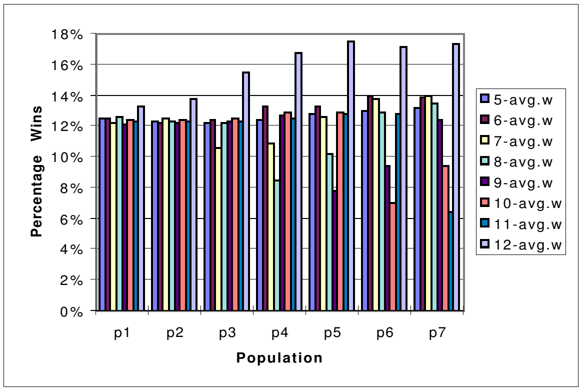

Figure 6: Percentage of times that each of the sellers wins the auctions in the different populations. Seller with 1-level models is #12 (i.e. 12-avg.w).

$p4$ and $p5$, the price distribution gives away the fact that there are two competing equilibrium prices. In other populations this is not as clear but can still be deduced.

Figures 6–8 show similar results for a similar series of populations except that in this case $p1$ has $c = q = 8$ and successive populations have members with decreased $c = q$, (i.e. $p7$ has seller 2 with $c = q = 7$, and so on...). That is, whereas our first experiment changed the population by adding higher cost/quality agents, this examples introduces lower cost/quality agents. We see a similar correlation between volatility and wins by the 1-level seller, except that in this case the discontinuity is not due to change in the population, since all these populations have one undisputed p^*. It is instead due to the fact that the 1-level seller has $c = q = 8$ and, therefore, will not bid lower than $p = 9$, so when the price distribution starts to drop below this point it can not take advantage of the increased volatility in that area. If we adjust the volatility by multiplying it times the percentage of time the price is above 8, then we get a better correlation.

6 Lessons

Here we present some of the main lessons we have learned from our work, so far. These are rather general and qualitative in nature but we hope to quantify some of these results and arrive at some formulas we can use to precisely predict when it is better to use deeper models.

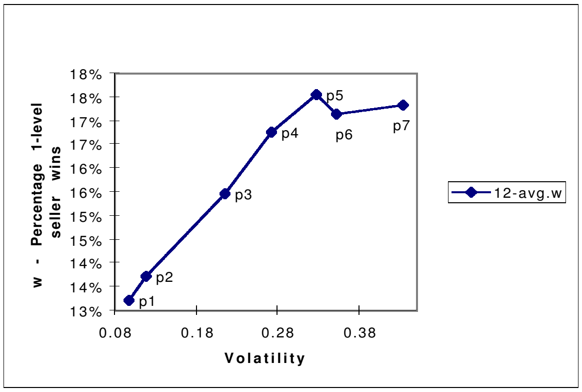

Figure 7: Scatter plot of volatility versus the percentage of time that the 1-level seller wins (w).

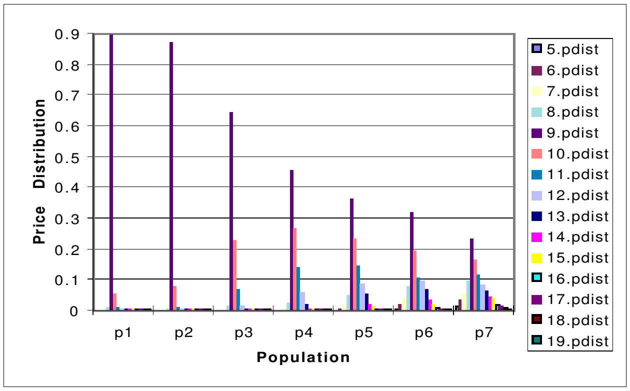

Figure 8: Average price distribution for the different populations. The highest peak corresponds to price 9.

There is usually no equilibrium One of the few times when we can be sure that there is an equilibrium is when all of the buyers prefer lower priced goods, regardless of the quality delivered. This case is the same as a traditional economic market, where if two producers produce the same good then, by definition, these goods are all identical and completely substitutable. The buyers simply pick the lowest price they get and the sellers sell at the lowest price they can sell, while still drawing a small profit. In such a system, sellers will be tempted to find ways of reducing their costs (perhaps by sacrificing quality), in order to lower their prices. This process, without the addition of some costly accreditation system, is destined to eventually lead all sellers to sell only the lowest priced goods of the lowest possible quality.

If, on the other hand, we assume that buyers value quality, then we find ourselves in a system with no clear equilibrium, unless all the sellers are selling the same quality. As the sellers start to offer different qualities, the system start to oscillate around (and not just between) the two points p^* and p^+.

Having low costs really pays off Given our previous explanation, it should come as no surprise that, in general, sellers with low cost have higher profits. These sellers can sell low when the buyers are buying low, and they can raise their prices and "pretend" to be high quality sellers. This deception does not really hurt them (given that the buyers have 0-level models); it simply drives prices down where they can still bid. It is the high quality sellers that are really hurt by this behavior and, therefore, would want to differentiate themselves from others. Such differentiation would be achieved if buyers had 1-level models.

Volatility This measure, it turns out, is a very effective tool for predicting the usefulness of 1-level models. In general, we find that 1-level modeling reaps higher rewards when the volatility of the system is high, and low rewards when volatility is low. The reason is that the 1-level modelers have, by definition, a better ability to predict what others will do. 0-level sellers are slower to respond to changes in the market. They take longer to find the new "stable" price.

Volatility alone is sometimes not enough to make correct predictions. We need to separate the populations into two categories, those with one undisputed p^*, and those where price fluctuates around p^* and p^+. Since a particular agent does not know what everybody's value function and costs are, we cannot expect him to determine what p^* and p^+ are for the population he is in. He can, however, observe the behavior of the system and try to guess the type of population he is in. Namely, he will observe the price distribution and if it shows only one peak, this is evidence that he is in a population with only one undisputed p^*. If it shows two or more peaks, then chances are he is in a population with dueling equilibria. Once an agent knows what population he is in then he can determine the exact correlation between volatility and the usefulness of using 1-level models.

We want to make clear a small caveat, which is that the volatility that is correlated to the usefulness of doing 1-level modeling, is the volatility of the system *with* the agent already doing 1-level modeling. Fortunately, having one agent change from 0-level to 1-level does not, in general, affect the volatility too much. Therefore, the volatility before the change can be used as a good predictor of the volatility after the change to 1-level models. This is important to us because we wish to develop methods and an agent, which is currently using 0-level models, can use to determine if it would be profitable to start using 1-level models.

It was also observed that, even within the same population, different volatilities are present at different times. The predictive power of the volatility is still evident, even at this finer level, since these volatilities were also correlated the usefulness of 1-level models.

7 Future Work

There is a lot of work left to be done within the framework presented here. We are conducting tests on many different types of agent populations, including 2-level sellers and 1-level buyers, in the hopes of getting a better understanding of how well different agents fare in the different populations. Once we have this understanding, we can take into account the costs associated with deeper models and try to come up with an algorithm or, at least, some criteria that agents can use to decide what modeling depth is the best to use.

We are also considering the expansion of the model with the possible additions of agents that can both buy and sell, and sellers that can return different quality goods. Both of these extension endeavor to bring our model closer to the reality of a multi-agent system. In the same vein, we are already implementing some of the agent capabilities shown in this paper, into the UMDL [2] multi-agent system.

In the long run, another offshoot of this research could be a better characterization of the type of environments and how they allow/inhibit "cheating" behavior on different agent populations. That is, we saw how, in our economic model, agents are sometimes rewarded for behavior that does not seem to be good for the community as a whole. The rewards, we are finding, start to diminish as the other agents become "smarter". It would be very useful to characterize the type of environments and agent populations that, combined, foster such antisocial behavior (see [9]), especially as interest in multi-agent systems grows.

References

[1] George A. Akerlof. The market for 'lemons': Quality uncertainty and the market mechanism. *The Quaterly Journal of Economics*, pages 488–500, August 1970.

[2] Daniel E. Atkins, William P. Birmingham, Edmund H. Durfee, Eric J. Glover, Tracy Mullen, Elke A. Rundensteiner, Elliot Soloway, José M. Vidal, Raven Wallace, and Michael P. Wellman. Toward inquiry-based education through interacting software agents. *IEEE Computer*, 29(5), May 1996. `http://www.computer.org/pubs/computer/dli/r50069/r50069.htm`.

[3] Robert Axelrod. *The Complexity of Cooperation*, chapter Evolving New Strategies, pages 10–29. Princeton University Press, 1997.

[4] Edmund H. Durfee, Piotr J. Gmytrasiewicz, and Jeffrey S. Rosenschein. The utility of embedded communications and the emergence of protocols. In *Proceedings of the 13th International Distributed Artificial Intelligence Workshop*, 1994.

[5] Joshua M. Epstein and Robert L. Axtell. *Growing Artifical Societies: Social Science from the Bottom Up.* Brookings Institution, 1996.

[6] Piotr J. Gmytrasiewicz. On reasoning about other agents. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents*

Volume II, Lecture Notes in Artificial Intelligence, pages 143–155. Springer-Verlag, 1996.

[7] Piotr J. Gmytrasiewicz and Edmund H. Durfee. A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 125–132, 1995.

[8] Tracy Mullen and Michael. P. Wellman. Some issues in the design of market-oriented agents. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents Volume II*, Lecture Notes in Artificial Intelligence, pages 283–298. Springer-Verlag, 1996.

[9] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, Massachusetts, 1994.

[10] Stuart Russell. Rationality and intelligence. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 950–957, 1995.

[11] Stuart Russell and Eric Wefald. *Do The Right Thing*. The MIT Press, Cambridge, Massachusetts, 1991.

[12] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.

[13] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[14] Milind Tambe and P. S. Rosenbloom. Architectures for agents that track other agents in multi-agent worlds. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents Volume II*, Lecture Notes in Artificial Intelligence, pages 156–170. Springer-Verlag, 1996.

[15] José M. Vidal and Edmund H. Durfee. Task planning agents in the UMDL. In *Proceedings of the Fourth International Conference on Information and Knowledge Managment (CIKM) Workshop on Intelligent Information Agents.*, 1995. `http://jmvidal.ece.sc.edu/papers/tpa/`.

[16] Christopher J. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.