In *Proceedings of the Second International Conference on Multiagent Systems (ICMAS-96),*
Kyoto, Japan, December 1996

Self-fulfilling Bias in Multiagent Learning

Junling Hu and Michael P. Wellman
Department of EECS, AI Laboratory
University of Michigan
Ann Arbor, MI 48109-2110 USA
{junling, wellman}@umich.edu

Abstract

Learning in a multiagent environment is complicated by the fact that as other agents learn, the environment effectively changes. Moreover, other agents' actions are often not directly observable, and the actions taken by the learning agent can strongly bias which range of behaviors are encountered. We define the concept of a *conjectural equilibrium*, where all agents' expectations are realized, and each agent responds optimally to its expectations. We present a generic multiagent exchange situation, in which competitive behavior constitutes a conjectural equilibrium. We then introduce an agent that executes a more sophisticated strategic learning strategy, building a model of the response of other agents. We find that the system reliably converges to a conjectural equilibrium, but that the final result achieved is highly sensitive to initial belief. In essence, the strategic learner's actions tend to fulfill its expectations. Depending on the starting point, the agent may be better or worse off than had it not attempted to learn a model of the other agents at all.

Introduction

Machine learning researchers have recently begun to investigate the special issues that multiagent environments present to the learning task. Recent workshops on the topic (Grefenstette & others 1996; Sen 1996). have helped to frame research problems for the field. Multiagent environments are distinguished in particular by the fact that as the agents learn, they change their responses, thus effectively changing the environment for all of the other agents. When agents are acting and learning simultaneously, their decisions affect (and limit) what they subsequently learn.

The changing environment and limited ability to learn the full range of others' behavior presents pitfalls for an individual learning agent. In this paper, we explore a simple multiagent environment representing a generic class of exchange interactions. One agent (called *strategic*) attempts to learn a model of the others' behavior, while the rest learn a simple reactive policy. We find the following:

1. The system reliably converges to an expectations equilibrium, where the strategic agent's model of the others is fulfilled, all the rest correctly anticipate the resulting state, and each agent behaves optimally given its expectation.

2. Depending on its initial belief, the strategic agent may be better or worse off than had it simply behaved reactively like the others.

The apparent paradox in this situation is that the learning itself is highly effective: the other agents behave exactly as predicted given what the agent itself does. The paradox is easily resolved by noting that the learned model does *not* correctly predict what the result would be if the agent selected an alternate action. Nevertheless, it is perhaps surprising how easy it is for the agent to get trapped in a suboptimal equilibrium, and that the result is often substantially worse than if it had not attempted to learn a model at all.

We refer to the above situation as *self-fulfilling bias*, because the revisions of belief and action by the agent reinforce each other so that an equilibrium is reached. Here bias is defined as in the standard machine learning literature—the preference for one hypothesis over another, beyond mere consistency with the examples (Russell & Norvig 1995). In reinforcement learning, the initial hypothesis is a source of bias, as is the hypothesis space (in multiagent environments, expressible models of the other agents). The combination of a limited modeling language (in our experiments, linear demand functions) with an arbitrarily assigned initial hypothesis strongly influences the equilibrium state reached by the multiagent system.

Most work on multiagent learning to date has investigated some form of reinforcement learning (Tan 1993; Weiß 1993). The basic idea of reinforcement learning is to revise beliefs and policies based on the success or failure of observed performance. When only policies are revised, the process can be viewed as hill-climbing search in policy space.

If the environment is well structured and agents have some knowledge about this structure, it would seem advantageous to exploit that knowledge. In a multiagent environment, such structure may help in learning about other agents' policy functions. In the experiments reported here, our strategic agent knows about the basic structure of the market system, and it uses that knowledge to form a model of other agents. Starting from an original model, the agent uses observations to update the model to increase accuracy. This process can be viewed as hill climbing in a space of agent models.

The technical question is whether this form of learning with limited information will converge to a correct model of the environment, and whether the learning agent will be better off using this model. Our theoretical and experimental investigations show that even when convergence to a "correct" model obtains, improvement in result does not always follow.

To our knowledge, the phenomenon of self-fulfilling bias as defined here has not been well studied in multiagent learning. Economists studying bidding games (Samples 1985; Boyle 1985) have noticed that biased starting bid prices strongly influence final bids. But these empirical findings have not been incorporated into a general framework in terms of learning. Machine learning researchers, on the other hand, directly address the general relationship of bias and learning, but not usually in the context of interacting rational agents.

Conjectural Equilibrium

Self-fulfilling bias arises from lack of information. When an agent has incomplete knowledge about the preference space of other agents, its interaction with them may not reveal their true preferences even over time.

This situation differs from the traditional game theory framework, where the joint payoff matrix is known to every agent. Uncertainty can be accommodated in the standard game-theoretic concept of *incomplete information* (Gibbsons 1992), where agents have probabilities over the payoffs of other agents. However, a state of complete ignorance about other agents' options and preferences can be expressed more directly, albeit abstractly, by omitting any direct consideration of interagent beliefs.

Consider an n-player game $G = (A, U, S, s)$. $A = \{A^1, \ldots, A^n\}$, where A^i is the action space for agent i. $U = \{U^1, \ldots, U^n\}$ is the set of agent utility functions. $S = S^1 \times \cdots \times S^n$ is the state space, where S^i is the part of state relevant to agent i. A utility function U^i is a map from the state space to real numbers, $U_i : S^i \to \Re$, ordering states by preference. We divide the state determination function s, into components, $s^i : A^1 \times \cdots \times A^n \to S^i$. Each agent knows only its own utility function, and the actions chosen by each agent are not directly observable.

Each agent has some belief about the state that would result from performing its available actions. Let $\tilde{s}^i(a)$ represent the state that agent i believes would result if it selected action a. Agent i chooses the action $a \in A^i$ it believes will maximize its utility.[1]

We are now ready to define our equilibrium concept.

Definition 1 *In game G defined above, a configuration of beliefs $(\tilde{s}^1, \ldots, \tilde{s}^n)$ constitutes a* conjectural equilibrium *if, for each agent i,*

$$\tilde{s}^i(a^i) = s^i(a^1, \ldots, a^n),$$

where a^i maximizes $U^i(\tilde{s}^i(a^i))$.

If the game is repeated over time, the agents can learn from prior observations. Let $a^i(t)$ denote the action chosen by agent i at time t. The state at time t, $\sigma(t)$, is determined by the joint action,

$$\sigma(t) = s(a^1(t), \ldots, a^n(t)).$$

We could incorporate environmental dynamics into the model by defining state *transitions* as a function of joint actions plus the current state. We refrain from taking this step in order to isolate the task of learning about other agents from the (essentially single-agent) problem of learning about the environment. In consequence, our framework defines a repeated game where agents are myopic, optimizing only with respect to the next iteration.

The dynamics of the system are wholly relegated to the evolution of agents' conjectures. At the time agent i selects its action $a^i(t)$, it has observed the sequence $\sigma(0), \sigma(1), \ldots, \sigma(t-1)$. Its beliefs, $\tilde{s}^i$, therefore, may be conditioned on those observations, and so we distinguish beliefs at time t with a subscript, $\tilde{s}^i_t$. We say that a learning regime *converges* if $\lim_{t\to\infty}(\tilde{s}^1_t, \ldots, \tilde{s}^n_t)$ is a conjectural equilibrium. Our investigation below shows that some simple learning methods are convergent in a version of the game framework considered above.

[1] A more sophisticated version of this model would have agents form probabilistic conjectures about the effects of actions, and act to maximize expected utility.

Note that our notion of conjectural equilibrium is substantially weaker than Nash equilibrium, as it allows the agent to be wrong about the results of performing alternate actions. Nash equilibria are trivially conjectural equilibria where the conjectures are consistent with the equilibrium play of other agents. As we see below, competitive, or Walrasian equilibria are also conjectural equilibria.

The concept of *self-confirming equilibrium* (Fudenberg & Levine 1993) is another relaxation of Nash equilibrium which applies to a situation where no agent ever observes actions of other agents contradicting its beliefs. Conjectures are on the play of other agents, and must be correct for all reachable information sets. This is stronger than conjectural equilibrium in two respects. First, it applies at each stage of an extensive form game, rather than for single-stage games or in the limit of a repeated game. Second, it takes individual actions of other agents as observable, whereas in our framework the agents observe only resulting state.

Boutilier (Boutilier 1996) also considers a model where only outcomes are observable, comparing the effectiveness of alternate learning mechanisms for solving multiagent coordination problems.

The basic concept of conjectural equilibrium was first introduced by Hahn, in the context of a market model (Hahn 1977). Though we also focus on market interactions, our basic definition applies the concept to the more general case. Hahn also included a specific model for conjecture formation in the equilibrium concept, whereas we relegate this process to the learning regime of participating agents.

Multiagent Market Framework

We study the phenomenon of self-fulfilling bias in the context of a simple market model of agent interactions. The market context is generic enough to capture a wide range of interesting multiagent systems, yet affords analytically simple characterizations of conjectures and dynamics. Our model is based on the framework of general equilibrium theory from economics, and our implementation uses the WALRAS market-oriented programming system (Wellman 1993), which is also based on general equilibrium theory.

General Equilibrium Model

Definition 2 *A* pure exchange economy, $E \equiv \{\langle X^i, U^i, e^i\rangle \mid i = 1, \ldots, n\}$, *consists of n consumer agents, each defined by:*

- *a* consumption set, $X^i \subseteq R^m_+$, *representing the bundles of the m goods that are feasible for i,*
- *a* utility function, $U^i : X^i \to \Re$, *ordering consumption bundles by preference, and*
- *an* endowment, $e^i \in R^m_+$, *specifying i's initial allocation of the m goods.*

The relative prices of goods govern their exchange. The *price vector*, $P \in R^m_+$, specifies a price for each good, observable by every consumer. A *competitive* consumer takes the price vector as given, and solves the following optimization problem.

$$\max_{x^i} U^i(x^i) \text{ s.t. } P \cdot x^i \leq P \cdot e^i. \quad (1)$$

That is, each agent chooses a consumption bundle x^i to maximize its utility, subject to the *budget constraint* that the cost of its consumption cannot exceed the value of its endowment.

A *Walrasian*, also called *competitive equilibrium*, is a vector $(P^*, (x^1, \ldots, x^n))$ such that

1. at price vector P^*, x^i solves problem (1) for each agent i, and
2. the markets clear: $\sum_{i=1}^n x^i = \sum_{i=1}^n e^i$.

It is sometimes more convenient to characterize the agents' actions in terms of *excess demand*, the difference between consumption and endowment,

$$z^i = x^i - e^i,$$

and to write the market clearing condition as $\sum_{i=1}^n z^i = 0$. The *excess demand set* for consumer i is $Z^i = \{z^i \in R^m \mid e^i + z^i \in X^i\}$.

A basic result of general equilibrium theory (Takayama 1985) states that if the utility function of every agent is quasiconcave and twice differentiable, then E has a unique competitive equilibrium.[2]

Observe that any competitive equilibrium can be viewed as a conjectural equilibrium, for an appropriate interpretation of conjectures. The action space A^i of agent i is its excess demand set, Z^i. Let the state determination function s return the desired consumptions if they satisfy the respective budget constraints with respect to the market prices, and zero otherwise. Utility function U^i simply evaluates i's part of the allocation. The agents' conjectures amount to accurately predicting the budget constraint, or equivalently, the prices. In competitive equilibrium, each agent is maximizing with respect to its perceived budget constraint, and the resulting allocation is as expected. Thus, the conditions for conjectural equilibrium are also satisfied.

[2]It is possible to express somewhat more general sufficient conditions in terms of underlying preference orders, but the direct utility conditions are adequate for our purposes.

Iterative Bidding Processes

In the discussion thus far, we have assumed that agents are *given* the prices used to solve their optimization problem. But it is perhaps more realistic for them to form their own expectations about prices, given their observations and other knowledge they may have about the system. Indeed, the dynamics of an exchange economy can be described by adding a temporal component to the original optimization problem, rewriting (1) as

$$\max_{x^i(t)} U^i(x^i(t)) \text{ s.t. } \tilde{P}^i(t) \cdot x^i(t) \leq \tilde{P}^i(t) \cdot e^i(t), \qquad (2)$$

where $x^i(t)$ denotes i's demand at time t, and $\tilde{P}^i(t)$ denotes its *conjectured* price vector at that time.[3]

A variety of methods have been developed for deriving competitive equilibria through repeated agent interactions. In many of these methods, the agents do not interact directly, but rather indirectly through auctions. Agents submit bids, observe the consequent prices, and adjust their expectations accordingly.

Different ways of forming the expected price $\tilde{P}(t)$ characterize different varieties of agents, and can be considered alternate learning regimes. For example, the *simple competitive agent* takes the latest actual price as its expectation,

$$\tilde{P}^i(t) = P(t-1). \qquad (3)$$

More sophisticated approaches are of course possible, and we consider one in detail in the next section.

In the classic method of *tatonnement*, for example, auctions announce the respective prices, and agents act as simple competitors. Depending on whether there is an excess or surfeit of demand, the auction raises or lowers the corresponding price. If the aggregate demand obeys *gross substitutability* (an increase in the price of one good raises demand for others, which hence serve as substitutes), then this method is guaranteed to converge to a competitive equilibrium (under the conditions under which it is guaranteed to exist).

The WALRAS algorithm (Cheng & Wellman 1996) is a variant of tatonnement. In WALRAS, agent i submits to the auction for good j at time t its solution to (2), expressed as a function of P_j, assuming that the prices of goods other than j take their expected values. In other words, it calculates a *demand function*,

$$x^i_j(\tilde{P}^i_1(t), \ldots, P_j, \ldots, \tilde{P}^i_m(t)).$$

[3]In the standard model, no exchanges are executed until the system reaches equilibrium. In so-called *non-tatonnement processes*, agents can trade at any time, and so the endowment e is also a function of time.

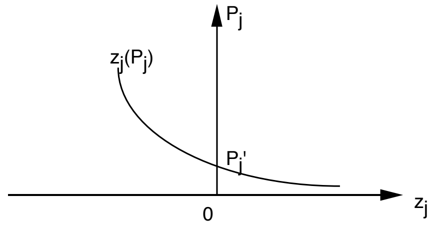

Figure 1: An aggregate excess demand curve for good j.

The bid they then submit to the auctioneer is their excess demand for good j,

$$z^i_j(P_j) = x^i_j(\tilde{P}^i_1(t), \ldots, P_j, \ldots, \tilde{P}^i_m(t)) - e^i_j(t).$$

The auctioneer sums up all the agents' excess demands to get an *aggregate excess demand* function,

$$z_j(P_j) = \sum_{i=1}^{n} z^i_j(P_j).$$

Figure 1 depicts an aggregate demand curve. We assume that $z_j(P_j)$ is downward sloping, the general case for normal goods. Given such a curve, the auctioneer determines the price P'_j such that $z_j(P'_j) = 0$, and reports this clearing price to the interested agents.

Given the bidding behavior described, with expectations formed as by the simple competitive agent, the WALRAS algorithm is guaranteed to converge to competitive equilibrium, under the standard conditions (Cheng & Wellman 1996). Such an equilibrium also represents a conjectural equilibrium, according to the definition above. Thus, the simple competitive learning regime is convergent, with respect to both the tatonnement and WALRAS price adjustment protocols.

Learning Agents

Agents *learn* when they modify their conjectures based on observations. We distinguish alternate learning regimes by the form of the conjectures produced, and the policies for revising these conjectures.

Competitive Learning Agents

An agent is *competitive* if it takes prices as given, ignoring its own effect on the clearing process. Formally, in our learning framework, this means that the conjectured prices $\tilde{P}$ do not depend on the agents' own actions—the excess demands they submit as bids. For example, the simple competitive agent described above simply conjectures that the last observed price is correct. This revision policy is given by equation (3).

Adaptive competitive agents adjust their expectations according to the difference between their previous expectations and the actual observed price

$$\tilde{P}^i(t) = \tilde{P}^i(t-1) + \gamma(P(t-1) - \tilde{P}^i(t-1)). \quad (4)$$

The learning parameter, γ, dictates the rate at which the agent modifies its expectations. When $\gamma = 1$, this policy is identical to the simple competitive agent's. Variations on this adaptation, for example by tracking longer history sequences, also make for reasonable conjecture revision policies.

Strategic Learning Agents

In designing a more sophisticated learning agent, we must take into account what information is available to the agent. In our market model, there are several relevant features of the system that an agent cannot directly observe:

- the endowments of other agents,
- the utility functions of other agents,
- the excess demands submitted by other agents, and
- the aggregate excess demand.

That is, the agents cannot observe preference, endowment, or the complete demand functions of other agents. What the agent does observe is the price vector. It also knows the basic structure of the system—the bidding process and the generic properties we assume about demand. In particular, it knows the following:

- Each local market clears at the price announced by the auctioneer. That is, the sum of agents' excess demands for good j is zero at each time t,

$$z_j^i(t) + \sum_{l \neq i} z_j^l(t) = 0.$$

- Aggregate excess demand is decreasing in price.

This fragmentary information is not sufficient to reconstruct the private information of other agents. In fact, it provides no individual information about other agents at all. The best an agent can do is to learn about the aggregate action it faces.

Because they know how the auctions work, the agents realize that they can affect the market price through their individual demands. A sophisticated agent, therefore, would take its own action into account in forming its expectation. Therefore, $\tilde{P}^i$ becomes a function of excess demand, $z^i(t)$, and then i's optimization problem is subject to a nonlinear budget constraint.

In our experiments with strategic learning, we adopt a simple model of an agent's influence on prices. Specifically, the agent assumes that its effect on price is linear for each good j,

$$\tilde{P}_j^i(t) = \alpha_j^i(t) + \beta_j^i(t) z_j^i(t). \quad (5)$$

As in our usual reinforcement-learning approach, the coefficients are adjusted according to the difference between the expected price and actual price:

$$\alpha_j^i(t+1) = \alpha_j^i(t) + \gamma_1(P_j(t) - \tilde{P}_j^i(t)), \quad (6)$$

$$\beta_j^i(t+1) = \beta_j^i(t) + \frac{\gamma_2}{z_j^i(t)}(P_j(t) - \tilde{P}_j^i(t)), \quad (7)$$

where γ_1 and γ_2 are constant coefficients.

Experimental Results

We have run several experiments in WALRAS, implementing exchange economies with various forms of learning agents. Our main experiments explored the behavior of a single strategic learning agent (as described above), included in a market where the other agents are simple competitors.

In the experiments, we generate agents with standard parametrized utility functions. Preference parameters and endowments are randomly assigned. The competitive agents have CES (constant elasticity of substitution) utility function—a common functional form in general equilibrium modeling. A standard CES utility function is defined as

$$U(x_1, \ldots, x_m) = (\sum_j a_j^{1-\rho} x_j^{\rho})^{\frac{1}{\rho}}.$$

In our experiments, we set $\rho = \frac{1}{2}$, and $a_j = 1$ for all j.

The strategic agent has a logarithmic utility function, $U(x_1, \ldots, x_m) = \sum_j \ln x_j$. This utility function is a limiting case of the CES form, with $\rho \to \infty$. The reason that we impose this special form on the strategic agent is simply for analytical convenience. The strategic agent's optimization problem is more complex because it faces a nonlinear budget constraint (i.e., its price conjecture is a function of its action), and the special form substantially simplifies the computation.

In our simulations, the competitive agents form conjectures by equation (3). The strategic agent forms conjectures given by equation (5), and it revises its conjecture given observations according to equations (6) and (7), with $\gamma_1 = \gamma_2 = \frac{1}{2}$.

Figure 2 presents a series of experimental results for a particular configuration with three goods and six agents. Each point represents the result from one experiment. The vertical axis represents the utility achieved by the strategic agent when the system reaches equilibrium. The horizontal axis represents the strategic agent's starting value of the β coefficient. For comparison, we also ran this configuration with the designated agent behaving competitively, i.e., forming expectations independent of its own behavior according, to the learning method (3). The utility thus achieved is represented by the horizontal line in the graph.

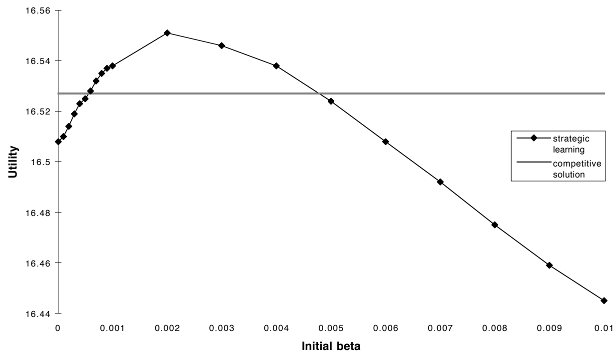

Figure 2: Utility achieved by the strategic learning agent, as a function of $\beta(0)$.

As Figure 2 demonstrates, the strategic agent can achieve higher or lower payoff than a similar competitive agent. For $\beta(0) \in [0.001, 0.004]$, the agent gets higher utility by learning the strategic model. Outside that region, the agent would be better off behaving competitively. (We also ran experiments for higher values of $\beta(0)$ than shown, and the trend continues.) Intuitively, the initial estimate of the agent's effect on prices moves it toward a demand policy that would fulfill this expectation.

The utility achieved by the other agents also crucially depends on the initial β of the strategic agent, as shown in Figure 3.

In the figure, the vertical axis represents the average of utilities achieved by all competitive agents in the system. (The difference in utility scale is due to the different functional form. Recall that utility values are ordinal only within a single agent.) The horizontal line represents the average of utilities of the same agents when the strategic agent behaving competitively. We can see that the smaller the initial β, the higher the average utility realized by the competitive agents. We currently lack an expressible intuition for the cause of this.

As we increase the number of competitive agents, the patterns of Figures 2 and 3 still hold. We also did experiments with multiple strategic agents in the

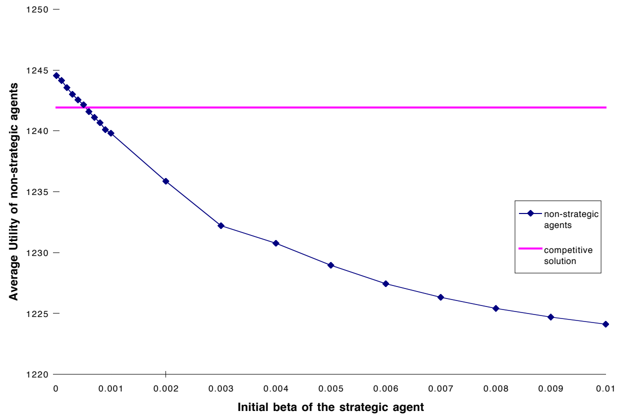

Figure 3: Average utility achieved by the competitive agents, as a function of the strategic agent's initial beta.

system. In all of our experiments, the system reliably converges to a conjectural equilibrium, although the particular equilibrium reached depends on the initial model of the strategic learning agents.[4]

Theoretical Analysis

Conjecture Functions

Our experimental analysis considered agents whose conjectures were either constant (competitive) or linear (strategic) functions of their actions. In this section, we provide some more general notation for characterizing the form of an agent's conjectures.

Definition 3 *The* conjecture function, $C^i : R^m \longrightarrow R_+^m$, *specifies the price system,* $C^i(z^i)$, *conjectured by consumer* i *to result if it submits excess demand* z^i.

The agent's conjecture about the resulting state, $\tilde{s}^i$, is that it will receive its demanded bundle if it satisfies its budget constraint. The actual resulting state is as demanded if the aggregate demands are feasible.

Definition 4 *The* market conjectural equilibrium *for an exchange economy is a point* $(C^1, \ldots, C^n)$ *such that for all* i, $\tilde{s}^i(z^i) = z^i$, *where*

1. $z^i = \arg\max U^i(z^i + e^i)$ *s.t.* $C^i(z^i) \cdot z^i = 0$, *and*

2. $\sum_i z^i \leq 0$.

Intuitively, $C^i(z^i) = P$, where P is the price vector determined by the market mechanism. However, nothing

[4]For configurations with only competitive agents (whether adaptive or simple), the system converges to the unique competitive equilibrium regardless of initial expectations.

in the definition actually requires that all agents conjecture the same price, though equivalent price conjectures with overall feasibility is a sufficient condition.

We can now characterize the existence of market conjectural equilibria in terms of the allowable conjecture functions.

Theorem 1 *Suppose E has a competitive equilibrium, and all agents are allowed to form constant conjectures. Then E has a market conjectural equilibrium.*

Proof: Let P^* be a competitive equilibrium for E. Then $C^i(z^i) = P^*$, $i = 1, \dots, n$, is a market conjectural equilibrium. □

Theorem 2 *Let E be an exchange economy, with all utility functions quasiconcave and twice differentiable. Suppose all agents are allowed to form constant conjectures, and at least one agent is allowed to form linear conjectures. Then E has an infinite set of market conjectural equilibria.*

Proof: Let P^* be a competitive equilibrium for E. Without loss of generality, let agent 1 adopt a conjecture of the form $C^1_j(z^1_j) = \alpha_j + \beta_j z^1_j$. Agent 1 is therefore strategic, with an optimal excess demand expressible as a function of α and β.[5] Let agents $i \neq 1$ adopt conjectures of the form $C^i_j(z^i_j) = P_j$. In equilibrium, the markets must clear. For all j,

$$z^1_j(\alpha, \beta) + \sum_{i=2}^{n} z^i_j(P) = 0.$$

We also require that agent 1's price conjecture for all goods j be equivalent to the other agents, $\alpha_j + \beta_j z^1_j = P_j$. We define a function

$$F(P, (\alpha, \beta)) = \begin{bmatrix} z^1_j(\alpha, \beta) + \sum_{i=2}^{n} z^i_j(P) \\ P_j - \alpha_j - \beta_j z^1_j \end{bmatrix},$$

and from above have $F(P, (\alpha, \beta)) = 0$ implies conjectural equilibrium. Since α, β, and P are each m-vectors with $m-1$ degrees of freedom, F represents the mapping $F : \Re^{m-1} \times \Re^{2(m-1)} \to \Re^{2(m-1)}$. The conditions on utility functions ensure that excess demand functions are continuous, and thus that F is continuously differentiable. The conditions also ensure the existence of a competitive equilibrium P^*, and therefore there is a point $(P^*, (P^*, 0))$ such that $F(P^*, (P^*, 0)) = 0$. Then by the Implicit Function Theorem (Spivak 1965), there exists an open set $\mathcal{P}$ containing P^* and an open set $\mathcal{B}$ containing $(P^*, 0)$ such that for each $P \in \mathcal{P}$, there is a unique $g(P) \in \mathcal{B}$ such that $F(P, g(P)) = 0$. All of these points $(P, g(P))$ constitute market conjectural equilibria for E. □

[5] Here we refer to the vectors $\alpha = (\alpha_1, \dots, \alpha_m)$ and $\beta_1, \dots, \beta_m$, since the excess demand for good j generally depends on conjectures about the prices for all goods.

Note that the conditions of Theorem 2 are satisfied by our experimental setup of the previous section. In that situation, the initial β determined which of the infinite conjectural equilibria was reached. Adding more strategic learning agents (those that could express non-constant conjecture functions) could only add more potential equilibria.

Learning and Convergence

The function C^i changes as consumer i learns about the effect of z^i on the price vector P. The strategic learning process given by equations (6) and (7) can be transformed into the following system of differential equations, assuming that we allow continuous adjustment. For all j,

$$\begin{aligned} \dot{\alpha}_j &= \gamma_1 (P_j - \alpha_j - \beta_j z_j), \\ \dot{\beta}_j &= \gamma_2 (P_j - \alpha_j - \beta_j z_j)/z_j. \end{aligned}$$

Note that all variables are functions of time. The z_j solve the strategic agent's optimization problem, thus each is a function α and β.

We assume that the market determines prices as a function of specified demands. In that case, we can express P_j as a function α and β as well.

Thus, the system of differential equations can be rewritten as

$$\begin{aligned} \dot{\alpha}_j &= \gamma_1 f_j(\alpha, \beta) \\ \dot{\beta}_j &= \gamma_2 f_j(\alpha, \beta)/z_j(\alpha, \beta)), \end{aligned}$$

where $f_j(\alpha, \beta) = (P_j(\alpha, \beta) - \alpha_j - \beta_j z_j(\alpha, \beta))$.

The equilibrium $(\bar{\alpha}, \bar{\beta})$ of this system is the solution of the following equations:

$$f_j(\alpha, \beta) = 0, \quad j = 1, \dots, m-1.$$

Since there are $m-1$ equations with $2(m-1)$ unknowns, the equilibrium is not a single point but a continuous surface, expressed as $(\bar{\alpha}, \bar{\beta}(\bar{\alpha}))$, where $\bar{\alpha} \in R^{m-1}$.

Although we do not yet have a proof, we believe that this learning process does converge, as suggested by our empirical results.

Welfare Implication

Our experiments demonstrated that a learning agent might rendered be better off or worse off by behaving strategically rather than competitively. However, the ambiguity disappears if it has sufficient knowledge to make a perfect conjecture.

If an agent's conjecture reflects full knowledge of the aggregate excess demand of the other agents, its utility from strategic behavior is at least as great as from competitive behavior.

Intuitively, if the agent makes a perfect conjecture, then it makes its choice based on the actual optimization problem it faces. Any other choice would either have lower (or equal) utility, or violate the budget constraint. However, when a strategic agent has imperfect information of the aggregate excess demand, for example a linear approximation, it may actually perform worse than had it used the constant approximation of competitive behavior.

Discussion

The fact that learning an oversimplified (in our case, linear) model of the environment can lead to suboptimal performance is not very surprising. Perhaps less obvious is the observation that it often leads to results worse than remaining completely uninformed, and behaving in a purely reactive (competitive) manner. Moreover, the situation seems to be exacerbated by the behavior of the agent itself, optimizing with respect to the incorrect model, and thus "self-fulfilling" the conjectural equilibrium.[6]

In our ongoing work on self-fulfilling bias, we are attempting to characterize more precisely the situations in which it can arise. In addition, we intend to explore techniques to overcome the problem. Random restart of the learning process is one straightforward approach, as is expanding the space of models considered (e.g., considering higher-order polynomials).

Another way to handle self-fulfilling bias is to transform this problem into a more traditional problem of decision under uncertainty. Agents that form probabilistic expectations may be less prone to get trapped in point equilibria. However, there is certainly a possibility of non-optimal expectations equilibrium even in this expanded setting.

A simple lesson of this exercise is that attempting to be a little bit more sophisticated than the other agents can be a dangerous thing, especially if one's learning method is prone to systematic bias. We hope to be able to provide more precise guidance for the design of strategic learning agents as a result of our further research.

[6] See (Kephart, Hogg, & Huberman 1989) for another setting where sophisticated agents that try to anticipate the actions of others often make results worse for themselves.

References

Boutilier, C. 1996. Learning conventions in multiagent stochastic domains using likelihood estimates. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, 106–114.

Boyle, K. 1985. Starting point bias in contingent valuation bidding games. *Land Economics* 61:188–194.

Cheng, J. Q., and Wellman, M. P. 1996. The WALRAS algorithm: A convergent distributed implementation of general-equilibrium outcomes. Submitted for publication.

Fudenberg, D., and Levine, D. K. 1993. Self-confirming equilibrium. *Econometrica* 61:523–545.

Gibbsons, R. 1992. *Game Theory for Applied Economists.* Princeton University Press.

Grefenstette, J. J., et al., eds. 1996. *AAAI Spring Symposium on Adaptation, Coevolution, and Learning in Multiagent Systems.* AAAI Press.

Hahn, F. 1977. Exercises in conjectural equilibrium analysis. *Scandinavian Journal of Economics* 79:210–226.

Kephart, J. O.; Hogg, T.; and Huberman, B. A. 1989. Dynamics of computational ecosystems. *Physical Review A* 40:404–421.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Samples, K. 1985. A note on the existence of starting point bias in iterative bidding games. *Western Journal of Agricultural Economics* 10:32–40.

Sen, S. 1996. IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems. *AI Magazine* 17(1):87–89.

Spivak, B. 1965. *Calculus on Manifolds.* Benjamin/Cummings.

Takayama, A. 1985. *Mathematical Economics.* Cambridge University Press.

Tan, M. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning.* Amherst, MA: Morgan Kaufmann.

Weiß, G. 1993. Learning to coordinate actions in multi-agent systems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 311–316.

Wellman, M. P. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research* 1:1–22.