A LEARNING ALGORITHM THAT MIMICS HUMAN LEARNING

by

W. Brian Arthur,

Stanford University *
and
Santa Fe Institute §

May 25, 1990

revised November 20, 1990

* Food Research Institute and Department of Economics, Stanford University, Stanford, CA 94305-6084,
§ Santa Fe Institute, 1120 Canyon Road, Santa Fe, NM 87501.

A LEARNING ALGORITHM THAT MIMICS HUMAN LEARNING

ABSTRACT

This paper explores the idea of calibrating a learning algorithm to match the way and rate at which human agents learn in a general, multi-choice setting. It develops a parametrized stochastic-learning algorithm, analyses its dynamics, and calibrates its parameters against human learning data from psychological experiments. The resulting calibrated algorithm appears to replicate human learning behavior to a high degree. It can therefore be used to replace the idealized, perfectly rational agents in appropriate neoclassical models with "calibrated agents" that represent *actual* human behavior.

The paper discusses the possibilities using the algorithm to represent human learning in normal-form stage games and in more general neoclassical models in economics. It explores the likelihood of convergence to long-run optimality, and to Nash behavior, and the "characteristic learning time" implicit in human adaptation in the economy.

A LEARNING ALGORITHM THAT MIMICS HUMAN LEARNING

by

W. BRIAN ARTHUR

INTRODUCTION

Although as economists we tend to believe that human rationality is bounded, still we know little about *how* rationality is bounded. Human decisions, we accept, sometimes fall short of perfect rationality; but we have little idea of where—on the line between perfect rationality and perfect absence of rationality—we should place the "dial of rationality"[1], and how we might build this dial-setting into actual theoretical models.

If there *is* a "dial of rationality" it is certainly not fixed at a constant level of performance. Human agents faced with a new economic environment improve their actions rapidly as they familiarize themselves with the setting and observe the results of their behavior. Thus, instead of asking where "rationality" might be fixed, we might ask whether there are models of *learning* or adaptation in economics whose parameters can be calibrated to match the way and rate at which human agents learn and adapt. If there are, we could replace the idealized, perfectly rational agents within standard theoretical models by more realistic, *calibrated learning agents* who replicate actual human behavior.

The idea that we might calibrate our learning models against actual human data, I believe, is timely. Modern machine-learning methods from computer science, (in particular John Holland's classifier systems) are beginning to introduce into economics the possibility of replacing the theoretical, perfectly-rational agents within standard neoclassical models with sophisticated learning algorithms—"artificially-intelligent agents"—that can gather information, improve their actions as they receive feedback from their environment, make sudden discoveries, and "learn to learn" at a meta-level.[2] It would seem sensible in constructing such algorithms to parametrize them and calibrate their parameters against human responses in

---

[1] The phrase is Frank Hahn's (personal communication). I thank in particular both John Holland and Richard Herrnstein whose ideas on adaptive behavior and learning did much to stimulate this paper. I also thank Kenneth Arrow, Vincent Crawford, Frank Hahn, David Lane, David Rumelhart, Andrzej Rusczynski and Tom Sargent for their insights and discussion, without implicating them in the views expressed here.

[2] For applications in economics see Marimon, McGrattan, and Sargent (1989), Arifovic (1990), and Miller (1989).

a corresponding context, whether measured in the actual economy or in an experimental economics laboratory. Neoclassical models containing accurately calibrated learning algorithms would then display behavior grounded upon *actual*, rather than idealized, human behavior.

Calibration would allow us to ask several questions that are not answerable under the assumption of a fixed, perfect rationality. We might want to ask whether actual human learning behavior—as captured by a calibrated model—converges to some standard outcome such as optimality, or Nash equilibrium.[3] We might ask, similarly, whether a given neoclassical model with calibrated agents will converge to some standard asymptotic pattern—a rational-expectations equilibrium, for example—or whether its behavior might fall short of this. We might want to to study the speed of adaptation and ask whether, under the calibrated parameters, behavior could be expected to converge *at all* in reasonable time. And we might want to study transient behavior, in particular the ephemeral patterns that might be ruled out by perfect rationality, but might nontheless appear temporarily, lock-in for a time, and disappear as the system "learns" its way past them. Some of these questions might be answerable by laboratory experiment, but the results of such experiments are not easily built in to theoretical models. Simple algorithms calibrated against experiments, on the other hand, may be useable as "learning modules" within larger, theoretical models to help answer these questions in more general settings.

Does there exist some yet-to-be-defined, specific algorithm of learning in economics that would be useable in all economic problems—an algorithm whose parameters, if we could estimate them, constitute universal constants of human behavior? I do not think so. Learning can differ considerably from one setting in economics to another. "Learning" about a job market may be a matter of probing a distribution of wages, searching over employers for better offers, and deciding upon a reasonable point to stop. "Learning" in setting a pricing policy may be a matter of implementing a policy, observing its consequences, adjusting the policy, trying alternatives, reobserving, and so on. Thus there are what we might call *contexts* of decision making in the economy, each with its distinctive style or variety of learning.[4] Different learning algorithms would be appropriate in these different contexts; and we would need a repertoire of algorithms to cover the various contexts that might arise.

What would it mean to calibrate an algorithm? In designing a learning system to represent human behavior in a particular context, we would be interested not only in reproducing human rates of learning, but also in reproducing the "style" in which humans learn, possibly even the ways in which they might depart from perfect rationality. The ideal, then, would not simply be learning curves that reproduce human learning curves to high goodness-of-fit, but more ambitiously, learning behavior that could pass the Turing test of being indistinguishable from human behavior with its foibles, departures and errors, to an observer

---

[3] The interesting recent work of Fudenberg and Kreps (1988) and Milgrom and Roberts (1989) runs parallel. This approach checks putative human learning behavior against particular, well-chosen sets to axioms, to test whether asymptotic behavior will result in standard outcomes. Other learning studies, for example Bray (1982), Marcet and Sargent (1989), view learning as the recursive estimation of parameters within well-defined neoclassical models.

[4] The psychological literature also recognizes many kinds of learning (Bower and Hilgard, 1981).

who was not informed whether the behavior was algorithm-generated or human-generated (Turing, 1956). Calibration ought not to be merely a matter of fitting parameters, but also one of building human-like qualitative behavior into the algorithm specification itself.

In this paper I explore the idea of calibrating a learning algorithm and discuss how we might use the resulting "calibrated agents" in appropriate neoclassical models. Section 2 develops a classifier-system-like learning algorithm for the simplest commonly recurring context that demands a learning approach in economics: that of economic agents repeatedly choosing among actions whose outcomes (or payoffs) are in advance to some degree unknown. I interpret this parametrized algorithm, analyze its transient behavior, and show that whether it converges to choosing the "optimal" action in the limit depends on its parameters' values. Section 3 calibrates the algorithm's parameters against available human data and shows that the algorithm can replicate human behavior to a high degree. But the resulting parameters, it turns out, lie outside the range that guarantees long-run optimality. In Section 4, I discuss this finding further: I show in fact that the likelihood of our "calibrated agents" achieving optimality depends on the degree to which the best action is easy to discriminate from the others. I also explore the possibilities of using this algorithm to represent learning agents in normal-form games and in more general neoclassical models in economics.

2. A PARAMETRIZED LEARNING ALGORITHM

Embedded within virtually all models in economics and game theory are agents making decisions. These decisions may have consequences that are either determinate or random, and that are in advance either known or unknown. The decisions may be made once-only or iterated; and they may be over actions on a continuum or in a discrete set. The class of models—the learning context—I will select here is that of agents choosing repeatedly among discrete actions with initially unknown, random consequences. This is the well-known multi-arm bandit problem in which the decision maker must choose one of N actions at each time, where the actions result in random payoffs or profits drawn from a stationary distribution that is unknown in advance. The agent—a firm, government agency, or consumer—might be faced with a choice among N alternative pricing schemes, or policy options, or research projects, or design features, or personnel policies, each with consequences that are poorly understood at the outset and that vary from "trial" to "trial." The agent chooses one of these at each time, observes its consequence or payoff, and over time updates his choice as a result.

Of course, there is no need that the alternative actions the agent chooses among are simple rules of thumb. Whether drawn from the actual economy or embedded in an economic model, actions may be highly sophisticated "subroutines" of behavior. The classic bandit problem is to design a learning algorithm or automaton that maximizes some criterion—such as expected discounted payoff or expected average profit (see Rothschild, 1974). Our problem is different. It is to design an algorithm or automaton that can be tuned to choose actions in this iterated choice situation the way humans would.

What makes this iterated choice problem interesting is the tension between *exploitation* of knowledge gained and *exploration* of poorly understood actions. For example one possible action—drilling for oil in a particular tract—may result when chosen in consistent payoffs clustered around $10M. An alternative—drilling in another tract—may pay zero the first ten times it is undertaken. But there may be 5% chance it pays $500M. There is no supposition that the agent—or any learning algorithm—knows this in advance; and it may take considerable exploration to find this out. Whether humans do enough exploration to eventually uncover and home in on the optimal action will be of interest to us in what follows.

We may think of "learning" in this iterated-choice context as updating the probabilities of choosing actions, as information on their consequences is received. An algorithm that does this is called a *stochastic learning* or *probability learning algorithm* in the psychology literature (Bush and Estes, 1959; Neimark and Estes, 1967) and a *learning automaton* in the machine-learning literature (Tsypkin, 1973; Narendra and Thathachar, 1989).

One possibility would be to draw upon one of the available stochastic learning algorithms in the psychological literature. These however are unsuitable for several reasons. They concentrate for the most part on two-choice problems where the consequences of the actions are the qualitative values "correct" or "incorrect." Payoffs that are monetary, and random, are difficult to cope with for these algorithms. And often they were designed to converge asymptotically to probability matching. Thus if one action pays one unit randomly 70% of the time and the other one unit randomly 30% of the time, they converge to triggering the two actions in the ratio 70:30. While there is some support for such behavior in some laboratory experiments,[5] it is unlikely that in the real economy, with actual profits at stake, decision makers would not go for an economic "edge" where it is obvious and easily learned. We do not want to enforce asymptotic optimality in the algorithm, but we do not want to rule it out at the beginning either.

A. The Algorithm

Consider a learning automaton that represents a single agent who can undertake one action of N possible actions at each time and who updates his probabilities of taking each action on the basis of the payoffs or outcomes he experiences. Action i brings reward $\Phi(i)$ which is unknown to the agent in advance, is positive, and in general is distributed randomly. The vector of rewards Φ has a stationary distribution.

The automaton—our artificial agent—"learns" via the following simple algorithm. It associates a vector of *strengths*, S_t, with the actions 1 through N, at each time t. C_t is the current sum of these

---

[5] Probability matching is discredited in psychology these days. As early as 1955 Goodnow showed that experimental subjects' asymptotic behavior depends heavily on whether they had been set a goal of being correct 100% of the time (in which case they attempt this by matching) or of maximizing the difference between correct responses and incorrect ones (in which case they eventually choose the better action consistently). And Gardner (1957) found that with more than two possible actions, subjects tended more to optimality than to probability-matching.

strengths (the component sum of S_t), and the initial strength vector S_0 is strictly positive. The vector p_t represents the agent's probabilities of taking actions 1 through N at time t.

At each time t, the agent:

1. Calculates the probability vector as the relative strengths associated with each action.
   That is it sets $p_t = S_t / C_t$

2. Chooses one action from the set according to the probabilities p_t and triggers that action

3. Observes the payoff received and updates strengths by adding the chosen action's j's payoff to action j's strength. That is, where action j is chosen, it sets the strengths to $S_t + \beta_t$ where $\beta_t = \Phi(j)e_j$; (e_j is the jth unit vector)

4. Renormalizes the strengths to sum to a pre-chosen constant.
   That is it sets $C_t = C$.

This algorithm has a simple behavioral interpretation.[6] We may think of the above strength vector as summarizing the current confidence the agent or automaton has learned to associate with actions 1 through *N*.[7] The confidence associated with an action increases according to the (random) payoff it brings in when taken. The automaton chooses its action with probabilities proportional to its current confidence in the *N* actions and learning takes place as these probabilities of actions are updated. The summed confidence in all actions is constrained to be constant. S_0, the initial confidence in the actions, represents prior beliefs, possibly carried over from past experience. The speed of learning in this algorithm will turn out to be proportional to 1/C. Hence C defines a one-parameter family that can be used to fit the algorithm to human behavior.

---

[6] It also bears similarities to the main models of stochastic learning in the psychological literature (see Sternberg, 1969). It is perhaps closest to Luce's (1959) "beta response-strength model," in which strengths are updated by multiplicative factors that depend on choice of action and its payoff (rather than additive factors as here). Alternatively, if we identify action-strengths with balls of different colors in an urn, it may be viewed as an urn scheme of the accumulation type (see Restle and Greeno, 1970; and also Arthur, Ermoliev and Kaniovski, 1987a and 1987b). Selection of an action then becomes equivalent to random choice of a color from the urn; new balls of the chosen action's color are added depending on the payoff received. Unlike other urn schemes in learning, however, the total number of balls is re-normalized at each step. It also resembles, in the way its dynamics work, the "linear models" of probability learning (Bush-Mosteller, 1955; Neimark and Estes, Chpt. 3, 1967) as we will see shortly.

[7] The idea of choice or response frequency as reflecting a latent property of *response strength* goes back to Tolman (1932) and to Hull (1943).

The algorithm also has a machine-learning interpretation. A Holland-type *classifier* is a condition/action couple (eg. "if object appears in left vision field / turn toward object"), where the action is allowed to be activated only if the condition is fulfilled.[8] If several classifiers have the same condition and that condition is fulfilled, they "compete" to be the one activated. Our algorithm can be viewed then as a set of N classifiers each competing to trigger its own action, where classifier j is the simple couple "if it is time to act / take action j." As is standard in classifier systems, strengths are associated with the classifiers; one classifier is triggered on the basis of current strengths; and the strength of the chosen classifier is updated by the associated reward.

It will sometimes be useful to work with a more general version of the algorithm. This replaces the renormalization *constant* in step four with a renormalization *sequence*, $C_t = C\,t^{\nu}$. The parameters C and ν, fixed in advance, now define a two-parameter family of algorithms that can be used to calibrate the automaton. C varies the speed as before, and ν now provides a deceleration or "annealing" term in the learning.

Notice that the algorithm is both non-linear and stochastic. It is non-linear in that actions that are frequently taken are further strengthened or reinforced, as in the classic Hebb's rule (Hebb, 1949). And it is stochastic in that actions are triggered randomly on the basis of current probabilities, and rewards are drawn randomly from a distribution. Heuristically, non-linearity allows for the exploitation of "useful" actions—ones that pay well tend to be strengthened early on and therefore to be heavily emphasized. And the stochastic property—triggering actions randomly on the basis of their strength—allows for exploration: if a little used action brings in a "jackpot" it may be strengthened sufficiently to become a frequent action.

B. Dynamics of the Learning Behavior.

To see more clearly how the algorithm works, I now turn to its dynamics. Begin with steps 3 and 4 of the algorithm. From these the strength vector is updated as

$$S_{t+1} = \frac{C_{t+1}}{C_t+B_t}\,(S_t + \beta_t) \qquad (1)$$

where the scalar random variable B_t is the component sum of the vector β_t.
We may rewrite this as

$$\frac{S_{t+1}}{C_{t+1}} = \frac{S_t}{C_t+B_t} + \frac{\beta_t}{C_t+B_t}$$

---

[8] In a Holland *classifier system*, classifiers are concatenated into an interdependent network, with actions taken serving as conditions for triggering choice among further, dependent actions. Classifier systems are powerful: they are computationally complete and so they can model anything a digital computer can. For classifier systems see Holland *et al.* (1986), Holland (1986), and Goldberg (1989).

$$= \frac{(C_t+B_t)S_t}{(C_t+B_t)C_t} - \frac{B_tS_t}{(C_t+B_t)C_t} + \frac{\beta_t}{C_t+B_t} . \qquad (2)$$

Recall that actions are chosen according to current relative strength, that is $S_t/C_t = p_t$. Denoting the random variable $(C_t+ B_t)^{-1}$ by α_t, we may therefore rewrite (2) simply as [9]

$$p_{t+1} = p_t + \alpha_t \{\beta_t - B_t p_t\} . \qquad (3)$$

Now define the function f(p) as $E[\beta(p) - Bp \mid p]$, the conditional expectation of the change in the action strengths given the action probabilities p, where the expectation is taken both over the distribution of each action's reward and with respect to the randomly chosen actions. Write the expected reward $E[\Phi(j)]$ as $\phi(j)$. Now, given action j, the expectation of $\beta - Bp$ is the vector $\phi(j)\,(e(j) - p)$. Action j is triggered with probability p(j). The function f is therefore given by

$$f(p) = \sum_j \phi(j)\,(e(j) - p)p(j) . \qquad (4)$$

Note that f is continuous. Now define the random vector $\xi_t(p_t)$ as

$$\xi_t(p_t) = \beta_t - B_t p_t - f(p_t) . \qquad (5)$$

(By the definition of f, the conditional expectation $E[\xi_t \mid p_t]$ is zero.)
We can then finally rewrite the algorithm's dynamics (4) as

$$p_{t+1} = p_t + \alpha_t f(p_t) + \alpha_t \xi_t(p_t) . \qquad (6)$$

Alternatively, from (4) the j-th component of the expected motion vector f(p) is $p(j)\{\phi(j) - \sum_i \phi(i)p(i)\}$, so that the jth component of p_t updates as

$$p(j)_{t+1} = p(j)_t + \alpha_t\, p(j)_t [\,\phi(j) - \sum_{i=1}^{N} \phi(i)p(i)_t\,] + \alpha_t\, \xi(j)_t .$$

Noting that p_t remains in the interior of the simplex, and writing $\xi(j)/p(j)$ as $\zeta(j)$, we have

$$\frac{p(j)_{t+1} - p(j)_t}{p_t(j)} = \alpha_t [\,\phi(j) - \sum_{i=1}^{N} \phi(i)p(i)_t + \zeta(j)_t\,] . \qquad (6')$$

---

[9] This scheme falls into the class of linear-reward-inaction schemes (Narendra and Thathachar, 1989).

We now have two representations for the dynamics—the transient learning behavior of the automaton. Equation (6) tells us that the action probabilities are updated at each time by an "expected motion" vector $f(p_t)$, together with an unbiased "perturbation" term ξ_t. And version (6′) tells us that the growth rate of the probability of choosing action j is driven by the difference between *its* expected payoff and the weighted-average expected payoff for *all* actions at the current probabilities p, plus unbiased noise. The step size of the algorithm, α_t, is given in both cases by $(Ct^{\nu}+ B_t)^{-1}$; it is random and is of the order $O(t^{-\nu})$. The overall rate of learning therefore increases both with larger step size and—as is realistic—with larger differences in expected payoff among the actions.

The dynamics in (6) take the form of a *stochastic approximation*, with state vector p, driving function f, and random step-size vector α_t (see Nevelson and Hasminskii, 1973). And the alternative version (6′) takes the form of a *replicator system* with noise. Viewed either way, the limiting behavior of the process will depend on the equivalent deterministic system (eg. (6) without the ξ_t term) and the rate ν at which the step size falls off.

C. *Asymptotic Optimality and Exploration*

One question we want to ask of human learning is whether it is likely to achieve asymptotic optimality by converging over time to activating only the choice with highest expected payoff. Put another way, we want to ask whether human learning explores alternative choices sufficiently to be sure to concentrate eventually on the best option. We can settle this by deriving theoretically the range of parameters under which the automaton does sufficient exploration to achieve asymptotic optimality and later comparing the human, calibrated parameters to see if they fall into this range.

It is certainly not clear *a-priori* whether the algorithm I have described learns over time to put all its "confidence" in the optimal action *k* and activate it only in the limit, or alternatively locks in to actions that are second- or third-best. It shows two apparently contradictory tendencies. On the one hand, its *expected* motion of learning is always in the direction of maximal-payoff action *k*: the *k*th component $p_t(k)$ in (6′) shows expected change $\alpha_t\, p_t(k)[\phi(k) - \Sigma_i \phi(i)p_t(i)]$ at time *t* and this is always positive. (See Figure 1.) On the other, it is subject to positive feedback in that high-payoff action *j*, triggered early, may gain in strength and therefore be triggered ever more frequently until it dominates. The maximal-payoff action *k* may then be insufficiently explored and become shut out.

It turns out that which of these tendencies dominates depends on the rate of decrease of the step size—that is, on the parameter ν. If the step-size remains of constant order ($\nu = 0$), then inferior action *j*, if emphasized early, may indeed build up sufficient strength to shut *k* out. In this case, action *j*'s strength—and therefore its probability of activation—may rise rapidly enough that with some positive probability after a finite time alternatives like *k* may cease to be triggered at all. In a sense here the algorithm is learning too fast. Learning may thus fall into the "gravitational orbit" of a non-optimal but reasonable action without escaping. If on the other hand ν is sufficiently large ($\nu = 1$) this does not happen. Here the

normalizing sequence $C_t = C.t$ increases linearly, and the step size falls at the corresponding rate 1/t. [10] This keeps action *k* "in the game" by delaying rapid lock-in to a non-optimal action and thus retains exploration for an arbitrarily long time. Even if *j* dominates early, sooner or later *k* will be triggered, further explored, and eventually take over.

Thus, whether long-run optimality is guaranteed depends on whether the value of ν is large enough. I state this more precisely in the following two theorems. Assume (A1): the random payoffs Φ are bounded above and below by positive constants and (A2) expected payoffs are unequal.

Theorem 1. Suppose A1, and that $0 < \nu < 1$. Then the vector sequence p_t converges to non-maximal-payoff vertex j of the simplex of probabilities S^N, with positive probability.

Theorem 2. Suppose A1 and A2, and that $\nu = 1$. Then the vector sequence p_t converges to optimal-payoff vertex *k* of the simplex of probabilities S^N, with probability one.

These results rely on probabilistic limit arguments that I relegate to an appendix.

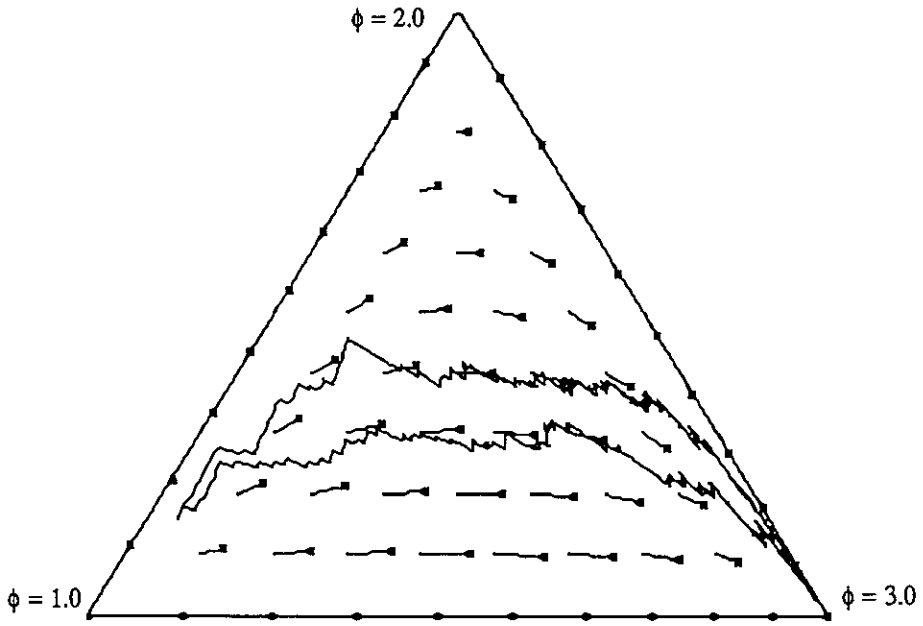

Fig 1. Expected Motions and two Learning Trajectories for a Three-Action Problem

These theorems give qualitative results, but they do not say anything about the practical probabilities of locking in to the wrong action. Here the other parameter, C, becomes important.

---

[10] This "cooling" of motion or step size is common in stochastic optimization to ensure optimality.

Obviously, non-optimality is more likely if the difference in expected payoffs between k and its next-best alternatives is small and differences are hard to discern. It is less likely the larger the constant C. Larger C means smaller step size; and with small enough steps attraction to a non-optimal action may occur slowly enough to allow "discovery" and exploitation of the optimal action. Also, practically speaking, ν need not be as large as 1 for optimality to occur with probability close to 1. If action k is more than 10% better than other actions, above $\nu = 0.5$ or so the probability of locking in to inferior actions becomes insignificant.

Example. Figure 1 shows a example of this learning automaton in practice. Three actions 1, 2, and 3 provide expected rewards of 1.0, 2.0 and 3.0, respectively. The figure maps the vector field of expected motions and shows two sample learning trajectories. Both trajectories have been started at a fairly extreme point (at bottom left) corresponding to high initial confidence in the poorest action 1, and where the activation probability of the best action 3 is low. For a short while the high-probability actions 1 and 2 compete, but then the automaton begins to "discover" action 3, and with its higher average reward its frequency of activation rapidly rises. Eventually the main competition is between actions 2 and 3 with convergence to almost 100% activation of 3 after about 200 steps. In practice, the algorithm would presumably start with equal probabilities, and convergence would then be considerably faster.

4. CALIBRATION AGAINST HUMAN SUBJECTS

Our next step is to calibrate the parameters C and ν against data on human learning. We are interested in three things here: the degree to which the calibrated algorithm represents human behavior; whether the measured value of ν lies within the range that guarantees asymptotically optimal choices; and the general characteristics of learning—such as speed and ability to discriminate—that the calibrated values imply.

To calibrate the algorithm I use the results of a series of two-choice bandit experiments conducted by Laval Robillard at Harvard in 1952-1953 using students as subjects (see Bush and Mosteller, 1955).[11] Robillard set up seven experiments, each with its own payoff structure, and allocated groups of ten subjects to each. Each subject, in his or her particular experiment, could choose action A or B in one hundred trials. In designating the experiments, 50 : 0 denotes that action A delivers one unit payoff randomly with probability .5, action B one unit payoff with probability zero; 80 : 40 denotes payoffs with probabilities .8 and .4 respectively; and so on. Table 1 summarizes the results. It shows the proportion of A-choices in each sequential block of ten trials, averaged over the group of ten subjects for that experiment.

---

[11] Data on humans choosing repeatedly among competing alternatives are scarce. Bush and Mosteller (1955) discuss the available human (and animal) multi-choice experiments. These experiments have gone out of fashion among psychologists, and no recent, more definitive results appear to be available.

A comment or two on this data set. The antiquity of these data should not bother us too much—we can presume that human behavior has not changed in the last four decades. But there are several shortcomings. Ideally I would prefer to calibrate on experiments that show wider variation in the expected

Experiment:	50 : 0	50 : 0	50 : 0	30 : 0	80 : 0	80 : 40	60 : 30
Trial Block							
0—9	0.52	0.51	0.49	0.56	0.59	0.50	0.49
10—19	0.63	0.54	0.58	0.57	0.77	0.59	0.58
20—29	0.69	0.67	0.67	0.55	0.88	0.71	0.62
30—39	0.63	0.59	0.59	0.63	0.88	0.64	0.51
40—49	0.64	0.66	0.63	0.60	0.86	0.63	0.51
50—59	0.75	0.66	0.64	0.66	0.91	0.63	0.61
60—69	0.76	0.77	0.71	0.65	0.92	0.53	0.57
70—79	0.85	0.70	0.73	0.65	0.89	0.71	0.57
80—89	0.87	0.83	0.72	0.65	0.88	0.73	0.65
90—99	0.90	0.83	0.81	0.66	0.89	0.70	0.55

Table 1. "Two-Armed Bandit" data on human learning obtained by Robillard.

value of rewards, longer trial lengths to allow for reasonable convergence of choice probabilities (if it is present), and more than just two alternatives. Further, I would want real, not token, monetary reward to be at stake.[12] In spite of these limitations, the experiments appear to have been conducted and documented with care, and I will use them as an expedient, until I possess proper experimental economic data. The resulting calibration should be interpreted as a good indication of human behavior in situations of choice, rather than a definitive statement.

Our artificial agents produce stochastic sequences of choices or frequencies of choosing action A, and so do the Robillard subjects. Goodness of fit (under any appropriate criterion) for fixed parameters is therefore a random variable. The object in calibration is then to choose parameters that maximize *expected* goodness of fit. I proceed therefore by allowing "groups of artificial agents" (computer runs of the algorithm) to reproduce the equivalent of Robillard's data for fixed values of C and ν. I then choose the parameters C and ν to fit the seven experiments taken together, by minimizing the expected sum of errors squared between the automaton-generated frequencies and the corresponding human frequencies for each particular experiment, totaled over the seven experiments.

[12] Robillard offered actual monetary reward (1 c and 5 c per correct response) for the second and third experiments only. These showed no observable difference in learning over the first experiment. In the calibration I have thus simply assigned payoff utility 1 to a "correct" action, and 0 to an "incorrect" action.

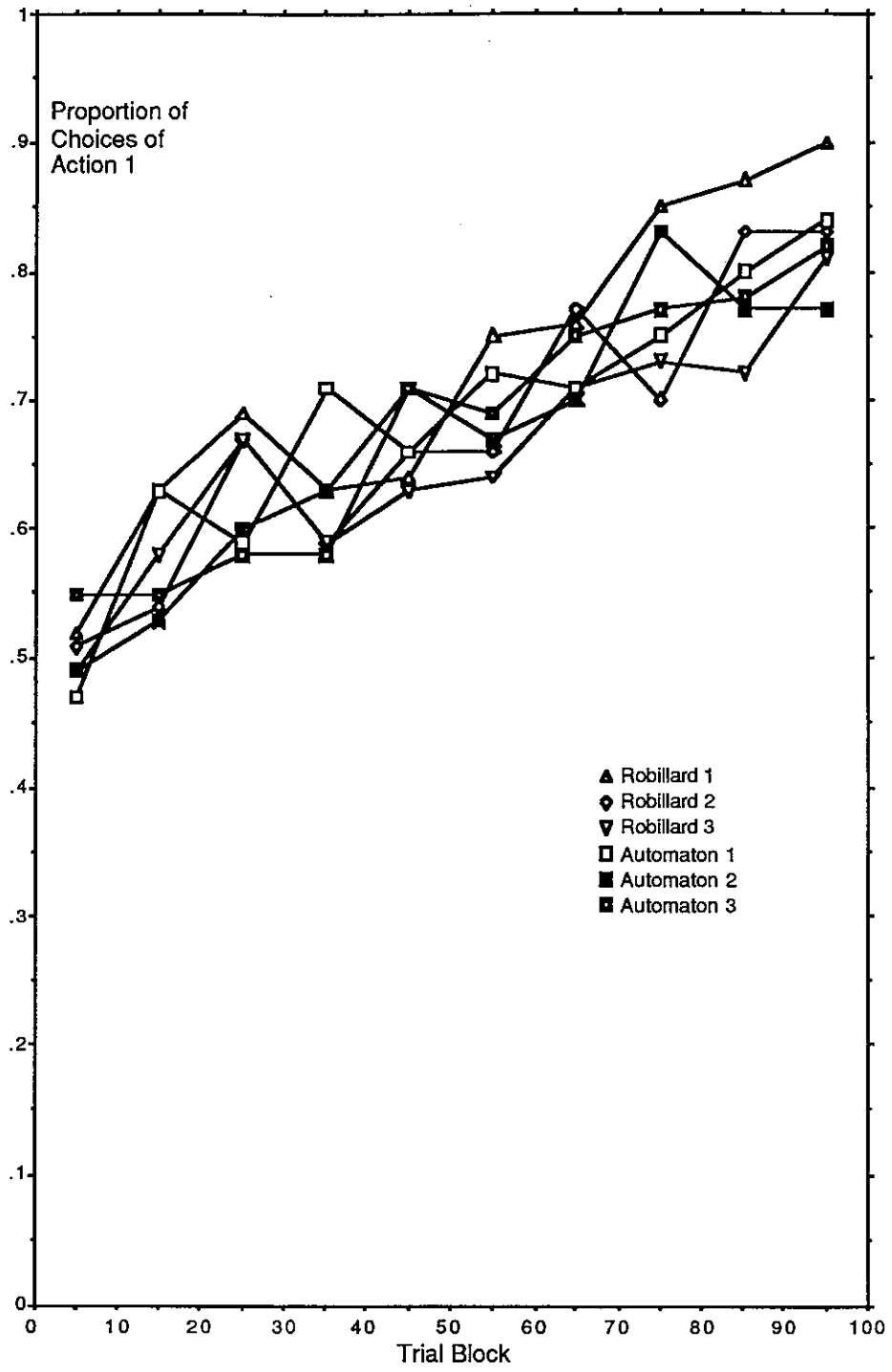

Fig. 2. Calibrated learning automata versus Robillard's human subjects. Choice frequencies in the three 50:0 experiments.

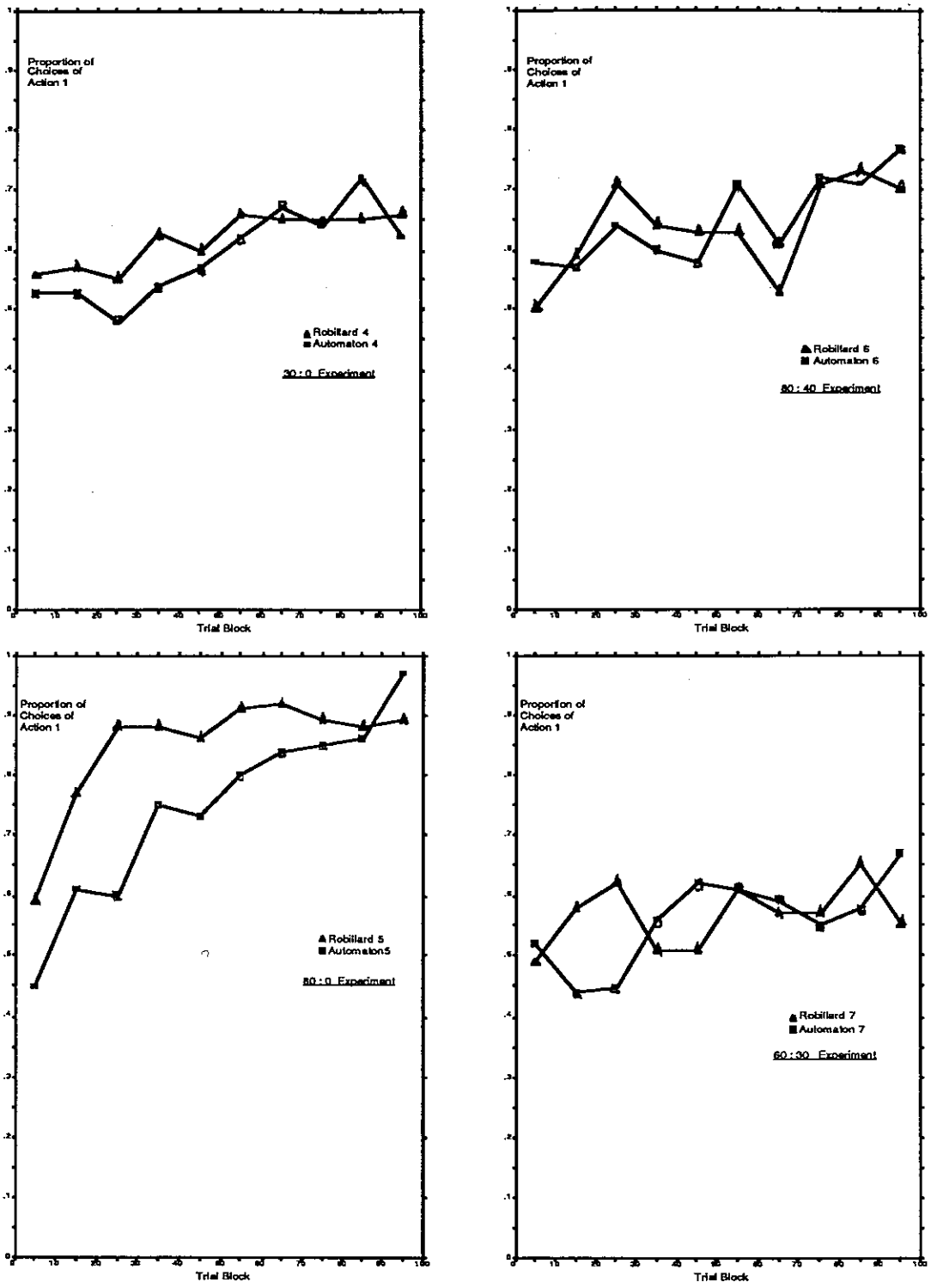

Fig 3. Automata versus Robillard subjects in the other four experiments.

This operation of choosing C and ν to fit the artificial agents' learning to the human learning data is a relatively simple numerical stochastic optimization problem. It results in C = 31.1 and ν = 0.00. Note immediately that ν lies in a region where optimality is far from guaranteed.

Figures 2 and 3 plot artificial agents' learning against human subjects', using these calibrated values. More specifically, for a representative seven-experiment set of artificial agent choices, it plots the average frequencies of activation of action *A* by seven groups of ten automata against the corresponding frequencies from Robillard's seven groups of ten human subjects. ("Representative" here means that the automata seven-experiment choices show goodness of fit to the human data approximately equal to the *expected* goodness of fit for the C and ν values used.) The results, judged by eye, are encouraging. The automata learning plots show roughly the same trend and variation as the human plots in each of the experiments. The parameter values are calibrated over all seven experiments together, and not for each one separately; yet the algorithm does not "average" among the experiments. Instead it varies, the change in choice frequency speeding up when reward differences are easy to discern (as in the 80 : 0 case), slowing down when discernment is difficult (the 60 : 30 case), closely tracking the behavior of the human subjects.

Are there systematic differences that distinguish the human learning from the calibrated automata learning curves? Consider the hypothesis that choice-frequencies identical to those of Robillard's subjects could be generated by our calibrated automata. To test this we can generate for each experiment 100 automata trajectories, measure the error-squared distance between pairs taken at random and compute the distribution, mean and variance of these pair-distances. We can similarly, for each experiment, compute the mean distance between the human (Robillard) trajectory and 100 corresponding, randomly generated automata ones. We would reject the hypothesis if the human trajectories lie on average far from the automata ones—on the tail of the automata pair-distance distribution.

Experiment	#1. 50:0	#2. 50:0	#3. 50:0	#4. 30:0	#5. 80:0	#6. 80:40	#7. 60:30
t-statistic	0.58	– 0.21	– 0.07	– 0.76	6.75	0.86	1.14

Table 2. t-distance of Robillard learning trajectories from randomly-generated automata learning trajectories.

Table 2 shows the t-statistic for the Robillard-automata pair-distances as compared to the inter-automata pair-distance distribution. It shows that six of the seven Robillard trajectories fall well within the distribution of automata trajectories and that the overall calibration is a good fit. Experiment 5 however is an outlier. Here humans learn faster than the calibrated algorithm. The reason is that its 80:0 payoff scheme has a close to deterministic outcome and for deterministic payoffs humans appear to speed up learning once they become convinced that actions produce the same payoff each time they are undertaken.[13] Such meta-learning might be built in by making C endogeneous but at the cost of complicating the algorithm.

---

[13] This is confirmed in the psychological experiments with deterministic outcomes of Goodnow (p.295, Bush and Mosteller, 1955) For Goodnow's non-deterministic payoff experiments our Robillard-calibrated artificial agents show very similar fits, out of sample, to those in Figures 2 and 3.

I would expect too that learning may also speed up somewhat—the parameter C might fall—as the payoffs at stake increased from a few dollars to several millions of dollars and casual decision-making

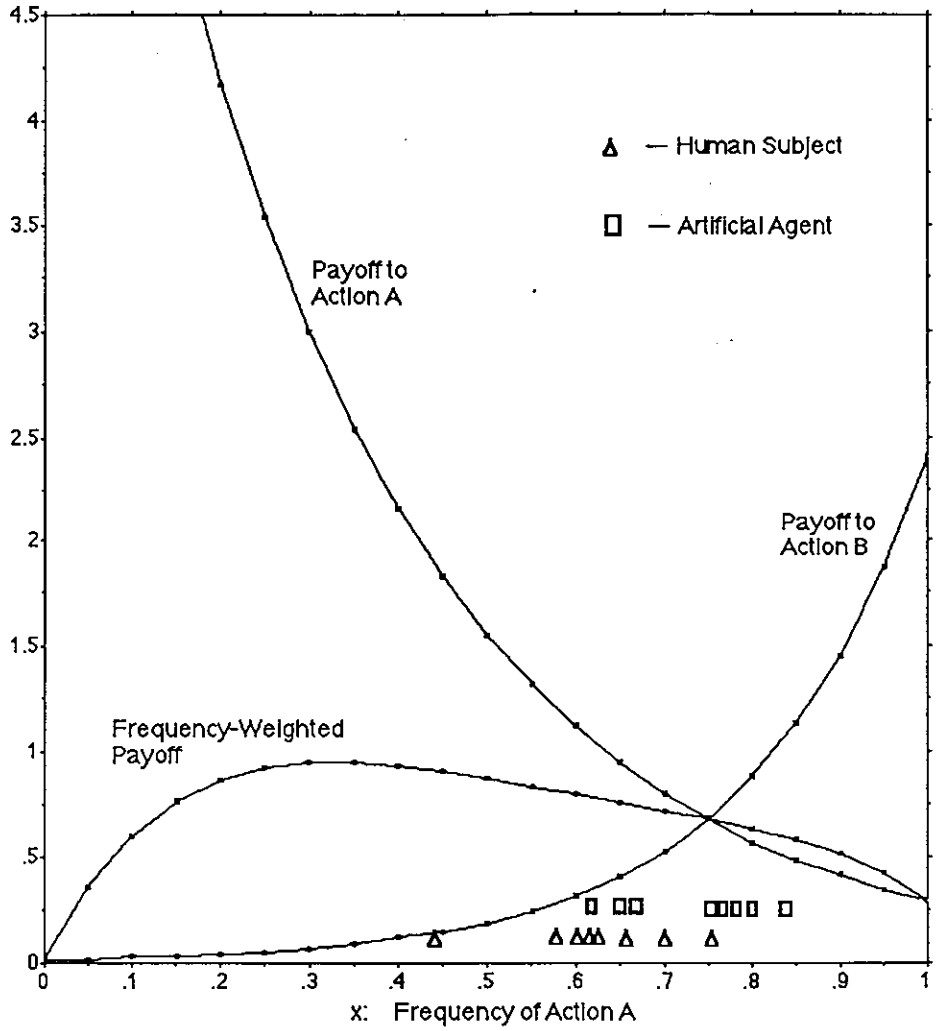

Fig. 4. Frequency Dependent Payoffs in the Herrnstein *et al.* Experiments.

behavior gave way to more calculated behavior. C would likely not fall too far—ignorance that must be narrowed remains ignorance, at any scale—but certainly it would be worth estimating at the high range of payoff to see how it might vary.

More convincing than statistical tests of fit would be tests of whether the algorithm can replicate human behavior in quite different choice problems than those for which it was calibrated. A different problem, with a different style of outcome (but still within our multi-choice context) is provided by recent experiments of Herrnstein, Prelec, Loewenstein, and Vaughan (1990). In these the payoff distributions are no longer fixed, but depend instead on the frequency of actions taken. These are of interest to us because

they show a well-documented characteristic behavior called *melioration* (Herrnstein, 1979; Herrnstein and Vaughan, 1980) and converge to an interior, non-optimal set of frequencies of choice.

In the Herrnstein *et al.* experiments there are two choices and their payoffs are frequency dependent. The payoff to choice *A* is $\Phi_A = 3^{(1.9 - 3x)}$ and to *B* is $\Phi_B = 3^{(0.8 - 4.6(1-x))}$, where x is the frequency of *A*-choices in the last twenty trials, as shown in Figure 5. These payoff functions are deterministic, but unknown in advance to the subject in the experiment. Above a frequency of 75% *A*-choices, option *B* pays better; below 75% option *A* pays better.

Human subjects, Herrnstein *et al.* find, tend to choose the action with the higher payoff at their current frequency—they "meliorate." Here this implies convergence to choosing *A* 75% of the time. Optimizing behavior on the other hand would maximize frequency-weighted payoff—it would set x, the frequency of A-choices, to maximize $(x\Phi_A + (1-x)\Phi_B)$. This implies choosing *A* about 33% of the time.

Figure 4 shows the frequency of *A*-choices in the last 50 of 400 trials, for eight Herrnstein subjects (plotted as triangles at the foot of the graph) and for eight of our calibrated agents (plotted as squares). The human subjects tend to meliorate, with a possible bias in the direction of optimality. The calibrated agents replicate this behavior. They also meliorate, but with no bias toward asymptotic optimality. That our algorithm picks up this characteristic of human behavior is not altogether surprising. Both human and artificial agents are carrying out local search in the frequency-dependent case, on the efficacy of choices at the current frequency of choice. Thus they deviate in the same direction, coming much closer to melioration than optimality.

Findings like this give us confidence that we *can* indeed replicate human behavior for a particular decision context with a simple parametrized learning algorithm.

4. Asymptotic Optimality, Nash Equilibria, and Other Questions

Let us now return to some of the questions asked at the start. In the basic multi-choice problem where payoffs are independent of frequency, what can we say about the chances of long-run optimality? We know from the zero calibrated value for ν that we can not expect optimality in all occasions where learning takes place. But what does this mean in practice? To the degree that the calibrated algorithm represents actual human learning, we can use computer experiments with the algorithm to test how often we might expect human decisions to lock in to less-than-optimal actions.

Consider one such experiment, a decision problem in which there are six choices, with action 1 the maximal choice. Expected payoffs fall off geometrically from action 1 to 6, and payoffs are distributed uniformly, from 0.5 to 1.5 times expected value for each of the actions. For rewarding the actions, average expected payoff over the six actions is scaled to unity. Obviously when expected payoffs are close, with all payoffs varying randomly, the optimal action becomes hard to discriminate. This difficulty of discrimination can be varied in our experiment by a parameter λ, the (common) ratio of expected payoffs between action 1 and action 2, action 2 and action 3, and so on, from a range of 0.7 (easy) to .98 (very difficult).

Table 3 shows the percentage of times action 1 dominated in 100 experiments at the end of 600 trials, using the calibrated algorithm. Human choice—if captured by the calibration—indeed "discovers" and exploits the optimal action until it becomes difficult to discriminate. When the best action ceases to be more than 10% better than the next best, chance variations in payoff cause performance to lock in to a less-than-optimal outcome in a significant percentage of experiments. (Even then, when it is on average only about 2% better than its next best competitor ($\lambda = .98$), action 1 dominates more than twice as often as any other.) Of course when all actions are the same ($\lambda = 1$), action 1 emerges about as often as the others. Whether the optimal action dominates in the long run then, depends on how difficult the choice problem is.

λ	.70	.72	.74	.76	.78	.80	.82	.84	.86	.88	.90	.92	.94	.96	.98	1.0
freq%	100	100	99	99	98	96	94	94	86	85	77	70	60	43	38	15

Table 3. Frequency of convergence to the optimal action at varying difficulty (λ) of discrimination.

It might be objected that the algorithm I have calibrated is to some degree ad-hoc—that this finding is merely an artifact of the algorithm. Would it be robust across other algorithmic specifications? I believe so. The reason is that what is crucial to the emergence of the optimal action is a slowing down in speed of convergence, so that learning has time to explore and "discover" the action with largest expected value. The data—not the algorithm—show that this slowing down does not occur. I would therefore expect the result that finding that the chances of long-run optimality is a function of the difficulty of the problem to be validated under other well-fitting specifications.

A similar finding carries over to the question of whether human agents are likely to converge to a Nash equilibrium in an iterated game. Think now of our calibrated agents representing human agents learning or adapting within a normal-form, stage game (Arthur,1989). The agents can observe their own actions and random payoffs, but are not particularly well informed of other players' actions and payoff functions. Each agent then faces a multi-choice bandit problem as before and so our learning context carries over to this wider problem.[14] An example might be oligopolistic firms choosing among pricing policies in a decentralized market on the basis of observed end-of-quarter profit.

For some games of this type learning behavior may not converge at all. The fact that learning agents change their choice-probabilities or strategy-profiles as other agents change *their* strategy-profiles may cause probability vectors to cycle. This is a familiar result in the adaptive theory of games. In games where learning *does* converge, to the degree that the calibration captures actual human behavior, we can assert that Nash is likely but not guaranteed (see Arthur 1989 for details). More precisely, as before, the likelihood of convergence to Nash depends on the difficulty of discrimination among the action payoffs.

---

[14] Of course, in this case where the payoff environment is no longer stationary; each agent's payoff distribution slowly changes as other agents change their choice probabilities. We would have to check that the calibration above carries over to this case. I suspect it would: the rate of learning would be little affected providing players' payoff distributions changed slowly.

The reason of course is that, once again, where discernment of the optimal (best-reply) action is difficult, agents may come to concentrate their play on "good" actions that are working well, and not explore less known but potentially superior alternatives. Individual behavior may then not be best-reply; and combined behavior may therefore not be Nash. Although in examples with closely clustered payoffs, Nash is not guaranteed, in the simpler games that I have tested with calibrated agents, Nash behavior always emerges.

How might we use calibrated agents to represent actual human adaptive behavior in other standard neoclassical models? As one example, consider a Lucas (1978) stock market. Here there are a constant number of units of a single stock, an alternative financial instrument that pays rate β, and a random dividend. Agents buy or sell the stock in discrete periods. Lucas shows that under rational expectations, there is a price equilibrium at fundamental value. We might ask whether this equilibrium is likely to be reached by *actual* human buying and selling behavior. Arthur, Holland, Palmer and Tayler (1990) explore this question using calibrated agents in an adaptive version of this market. Each agent is represented by a collection of classifiers, of the form "If the current price falls in the range x to $x + z$ / *buy* one unit of stock" or "If the current price falls in the range x to $x + z$ / *sell* one unit of stock." A specialist sets the price each period with the object of clearing the market. Starting from random behavior—50% chance of choosing "buy" or "sell"—we find that the calibrated agents learn to buy and sell stock appropriately as the price is below and above fundamental value. Within about one hundred and fifty buying-selling periods the price indeed converges to small fluctuations around fundamental value. However, we also find that small speculative bubbles sometimes occur, in the form of localized price "supports" and "ceilings"—a hint that under realistic learning technical analysis may emerge.

5. CONCLUSIONS

This paper has explored the possibility of calibrating learning models against experimental data, with the object of replacing the idealized, perfectly rational agents of standard neoclassical models with learning agents whose choice behavior replicates *actual* human behavior. The data used for calibration come from the psychology literature and are not perfectly suited for economic purposes; so the results here should be regarded as preliminary. But they suggest that we can indeed design "artificially intelligent" agents or learning automata that replicate human behavior in the repeated multi-choice problem to a high degree. The artificial agents' rate of learning varies with the difficulty of discerning expected payoff in the way the human rate varies. And in the case where payoffs depend on the frequency of choice, the artificial agents "meliorate" as humans consistently do. The calibrated iterated-choice algorithm developed here reduces to a well-defined stochastic process that can be inserted into theoretical models.

However, to the degree it replicates human behavior, the calibration shows that human learning can lock in to an inferior choice, and that this is prone to happen where payoffs to choices are closely clustered, random, and therefore difficult to discriminate among. This sort of finding is unfamiliar in economics, where our habit of thinking is that if there were a better alternative, it would be chosen. But

the data show that learning-steps do not slow down sufficiently to guarantee discovery of better alternatives. These may remain less understood and eventually ignored in the exploitation of "good" actions that pay off well early on. [15] If this finding holds up under better data sets (as I suspect it will) it implies that the question of whether human behavior adapts its way to an optimal steady-state, or a Nash outcome admits no simple yes or no answer. The answer depends on the problem: If alternatives are distinct and clearly different, we can expect standard outcomes. But where discrimination is difficult, non-standard outcomes—not too distant from optimality—are possible. This type of result is familiar in psychology (for a recent instance see Bailey and Mazur, 1990), where it has long been accepted that there are thresholds, beyond which better alternatives become difficult to discriminate.

Practically speaking, this possible lack of optimality would not matter a great deal—it would merely detract somewhat from efficiency—if decision problems throughout the economy were independent. But often they are not. Earlier actions may determine the options or alternative decisions available later. And knowledge gained in earlier problems may be carried into later, similar ones in the form of expectations or prior beliefs—a phenomenon called *transfer* in psychology. There is thus an "ecology" of decision problems in the economy, with earlier patterns of decisions affecting subsequent decisions. This interlinkage would tend to carry sub-optimality through from one decision setting to another. The overall economy would then follow a path that is partly decided by chance, is history-dependent, and is less than optimal.

How long it takes to uncover "reasonable" or optimal choices depends upon the degree to which payoffs are tightly clustered. But as a rule of thumb for actions with random payoffs that are unknown in advance, we should not expect behavior to have settled down much before forty to a hundred or more trials. In turn this implies that there is a *characteristic learning time* or "relaxation time" for human decisions in the economy that depends on the frequency of observed feedback on actions taken and on the payoff structure of the problem itself. There is also, of course, a time-frame or horizon over which the economic environment of a decision problem stays relatively constant. For some parts of the economy, this learning time or learning horizon may be shorter than the problem horizon, and we can expect these parts to be at equilibrium—albeit a slowly changing one. For other parts, learning may take place more slowly than the rate at which the problem shifts. These parts would be always transient—always tracking changes in their decision environment. They by contrast would not be at equilibrium.[16]

---

[15] This type of result is well known in "bandit" models with discounting. The calibration shows that in human behavior it carries over even under the benign assumptions of no discounting and no initial costs for exploration. We could of course stipulate that humans optimize under some criterion with an implicit discount rate. But this "discount rate" would then appear to be independent of time; I see little merit to going in this direction.

[16] And not at optimality either, if decisions had not yet "discovered" the optimal action. Note also that one part of the economy in learning motion may form the environment for another part. Thus transience may beget transience, and lack of equilibrium can percolate across the economy.

APPENDIX

Consider the two-parameter learning algorithm with dynamics described by (4), (5), and (6). Recall that starting strengths S_0 are positive. Assume: (A1) the random payoffs Φ are uniformly bounded above and below; and (A2) actions have unequal expected payoffs.

I now show the that if $0 \leq \nu < 1$, the process may possibly converge to a non-optimal action.

Theorem 1. *Assume* A1 *and that* $0 \leq \nu < 1$. *Then the vector sequence* p_t *may converge to triggering action j only (to vertex* j *of the simplex of probabilities* S^N*) with positive probability.*

Proof. Given A1, there is positive probability that the sequence reaches the interior of an arbitrary ε_1-neighborhood of inferior vertex e_j, in finite time. It is then sufficient to show that once the process is in this neighborhood there is positive probability that *only* action j will be triggered from that time on. I show this by straight computation.

Let $A_M(j)$ be the event that p lies within the ε_1-neighborhood of e_j and that *j* only is triggered from time *M* onward.

$$P\{A_M(j)\} = \prod_M^\infty p_s(j)$$

Now let $p_s(j) = 1 - a_s(j)$. Since $a_s(j) > 0$, for the infinite product $\prod (1 - a_s(j))$ to converge it is sufficient that the series $\sum a_s$ converge. In this case, the event $A_M(j)$ will have positive probability, and p_t will converge to sub-optimal vertex e_j with positive probability.

We therefore investigate the convergence of the series $\sum a_s$. Let action j receive random reward Φ. From (3)

$$p_{s+1}(j) = p_s(j) + (Cs^\nu + \Phi)^{-1} \Phi (1 - p_s(j))$$

so that

$$a_{s+1}(j) = a_s(j) (1 - \Phi (Cs^\nu + \Phi)^{-1})$$

Thus the ratio of succeeding terms in the *a*-series is

$$\frac{a_{s+1}(j)}{a_s(j)} = (1 - \Phi (Cs^\nu + \Phi)^{-1})$$

Note immediately that if $\nu = 0$, the fact that Φ is bounded below ensures that the *a*-series decreases at least as fast as a geometric series and therefore converges. Thus, where $\nu = 0$, the sequence p_t may converge to sub-optimal action *j* with positive probability.

Now let $0 < \nu < 1$. Denote the lower bound on Φ by r. Then, where time $s > C/r$, the ratio

$$1 - \frac{\Phi}{(Cs^\nu + \Phi)} < 1 - \frac{r}{Cs^\nu} < 1 - \frac{1}{s^\nu} = 1 - s^{-\nu}.$$

The convergent series $\sum s^{-(1+\varepsilon)}$ has ratio $(1- s^{-1})^{1+\varepsilon}$. By comparison, since $\nu < 1$, the *a*-sequence is then also convergent. Therefore, where $0 < \nu < 1$, the stochastic sequence p_t may converge to sub-optimal action *j* with positive probability. The theorem follows. ◊

I now consider $\nu = 1$. This implies that the step size falls at the rate 1/t. The process in (6) is therefore a standard stochastic approximation process on the unit simplex, with expected motion toward one point, vertex *k*. The only complication is that there are zeros at all vertices of the simplex, and no expected motion toward *k* on the subsimplices where p(k) = 0. I proceed via two lemmas, the first one showing that step size rate 1/t is slow enough to rule out convergence to a non-optimal vertex.

Lemma 1. *Suppose* A1 *and* $\nu = 1$. *Then the algorithm converges to a non-optimal action j (vertex j of the simplex,* $j \neq k$*) with probability zero.*

Proof. Consider the stochastic process (6) with driving function f as in (4) and perturbations ξ_t as in (5), starting at point p_0. At non-maximal vertex j, where $p = e_j$, $f(e_j) = 0$. The matrix of derivatives at e_j has an eigenvector $e_k - e_j$ with associated eigenvalue $\phi_k - \phi_j$ which is positive. Therefore the point e_j is linearly unstable. Because the payoffs are bounded above and below, the order of the step-size $\alpha_t = (Ct^{\nu} + B)^{-1}$ is t^{-1}. The perturbations $\xi_t(p) = \beta_t - B_t p_t - f(p_t)$ are bounded above uniformly on the simplex, and below outside a neighborhood of the vertices. We may now invoke the argument of Pemantle (1988; Theorem 1). This amounts to using probabilistic inequalities together with the above conditions to show that (i) whatever the starting point p_0, at some finite time τ, p_τ will lie outside an ε–neighborhood of e_j with probability greater than 1/2; and (ii) the probability the process enters an $\varepsilon/2$ neighborhood subsequently is less than c. A simple tail argument then shows that (i) and (ii) together rule out convergence to e_j. Therefore, $\text{Prob}\{p_t \to e_j\} = 0$. ◊

The next lemma shows that a process of the type we are dealing with cannot stay forever in a region with non-zero expected motion, provided a suitable Lyapunov function exists.

Lemma 2. *Let G be an open set and suppose there exists a nonnegative function* V(p) *defined on the domain* $t \geq 0$, $p \in G$, *such that* $E[V(p_{t+1}) - V(p_t) \mid p_t] \leq -\alpha_t$ *const, where* α_t *is a sequence such that*

$$\alpha_t > 0, \qquad \sum_{t=0}^{\infty} \alpha_t = \infty.$$

Then the process p_t *exits* G *in a finite time with probability one.*

Proof. This is Theorem 2.5.1 of Nevelson and Hasminskii (1976). The argument amounts to showing that if the process stayed in the domain, the cumulated increments of $V(p_t)$ would drive V negative with probability one, which contradicts the positivity of V. ◊

Theorem 2. *Assume* A1, A2 *and* $\nu = 1$. *Then the vector sequence* p_t *converges to the maximal expected-payoff vertex of the simplex with probability one.*

Proof. Relabel the vertices in order of expected payoff, so that vertex 1 is maximal, vertex 2 is second best, and so on. Define non-negative functions $V_1, V_2, \ldots, V_N$ (that map S^N into R) as $V_1 = 1-p(1), V_2 = 1-p(2), \ldots, V_N = 1-p(N)$. Also define $W_2, \ldots, W_N$ as $W_j = V_1 + V_2 + \ldots + V_j$. We first show that $V_1(p_t)$ is a supermartingale. From (6′) we have

$$E[V_1(t+1) - V_1(t) \mid p_t] = -E[p_{t+1}(1) - p_t(1) \mid p_t]$$

$$= -\alpha_t p_t(1) [(\phi(1) - \Sigma_i \phi(i) p_t(i)] \ . \qquad (7)$$

Since $\phi(1) > \phi(i)$ for all $i \neq 1$, $E[V_1(t+1) - V_1(t) \mid p_t] \leq 0$. Thus the sequence $\{V_1(p_t)\}$ is a supermartingale on S^N bounded below by zero and is therefore convergent. $W_j(p_t)$ is also a supermartingale. To show this think of actions 1 to j as the composite action j′. This has combined probability $p'_t(j) = p_t(1) + \ldots + p_t(j)$ and, if chosen, expected payoff $\phi'_j = [p_t(1)\,\phi(1) + \ldots + p_t(j)\phi(j)]/p'_t$. Thus j′'s expected payoff $\phi'(j)$ is a convex combination of the payoffs $\phi(1)$ to $\phi(j)$, which by A2 is strictly greater than the payoffs $\phi(j+1)$ through $\phi(N)$. We then have

$$E[W_j(t+1) - W_j(t) \mid p_t] = -E[(p'_{t+1}(j) - p'_t(j)) \mid p_t]$$

$$= -\alpha_t p'_t(j)[\phi'(j) - \Sigma_i \phi(i) p_t(i)]$$

$$= -\alpha_t p'_t(j) \ [\phi'(j) - \{p'_t(j)\phi'(j) + \sum_{i=j+1}^{N} \phi(i) p_t(i)\} \ ] \leq 0. \qquad (8)$$

W_j is therefore a supermartingale bounded below by $j-1$ and is convergent with probability one. From the convergence of $V_1(t)$ and $W_2(t)$ follows the convergence of $V_2(t)$; from this and the convergence of $W_3(t)$ follows the convergence of $V_3(t)$. Proceeding this way, all V_j converge. It follows that the sequence $\{p_t\}$ converges to a limit random vector γ with probability one. Now, Lemma 1 shows that, with probability one, γ cannot be a non-optimal vertex point e_j, $j \neq 1$. Suppose then γ is a non-vertex point h. Let the set U_h be an open ε-neighborhood of this point. Then there exists a finite time t′, for which $p_t \in U_h$ from t′ onward. Let j be the index of the first non-zero component of h. Then within U_h, the expected payoff of the composite action j′ strictly exceeds that of the other actions by a constant c; and by (8), $E(W_j(t+1) - W_j(t) \mid p_t) < -\alpha_t$ const for $t > t'$. The step size α_t is positive, and by A1, with $\nu = 1$, the summation $\Sigma_t \alpha_t > \Sigma_t (Ct + \underline{B})^{-1} = \infty$. But then U_h, W, and α_t fulfill the requirements of Lemma 2, and the process p_t for $t > t'$, must exit U_h in finite time with probability one. This contradicts $\gamma = h$. It follows with probability one that the sequence $\{p_t\}$ can converge only to the maximal vertex e_1. The theorem is proved. ◊

REFERENCES

Arthur, W.B. 1989. "Nash-Discovering Automata for Finite Action Games." Mimeo. Santa Fe Institute.

Arthur, W.B., Ermoliev, Yu. M. and Kaniovski, Yu.M. 1987a. "Nonlinear Urn Processes: Asymptotic Behavior and Applications." WP-87-85, IIASA, Laxenburg, Austria.

Arthur, W.B., Ermoliev, Yu. M. and Kaniovski, Yu.M. 1987b. "Urn Schemes and Adaptive Processes of Growth," *Kibernetika*, 23, 49-58.

Arthur, W.B., Holland, J.H. and Palmer, R. "Adaptive Behavior in the Stock Market." Paper in progress. Santa Fe Institute.

Bailey, J.T. and J.E. Mazur. 1990. "Choice Behavior in Transition: Development of Preference for the Higher Probability of Reinforcement." *Journal of Experimental Analysis and Behavior.*

Bower, G.H. and E.R. Hilgard. 1981. *Theories of Learning.* 5th Ed. Prentice-Hall: N.J.

Bray, M.M. 1982. "Learning, Estimation, and the Stability of Rational Expectations." *Journal of Econ. Theory.* 26, 318-339.

Bush R. R. and F. Mosteller. 1955. *Stochastic Models for Learning.* Wiley. New York.

Estes, W.K. 1950. "Toward a Statistical Theory of Learning." *Psychological Review.* 57, 94-107.

Fudenberg D. and D.M.Kreps. 1988. "Learning, Experimentation, and Equilibrium in Games." Mimeo. MIT.

Gardner, R.A. 1957. "Probability-Learning with Two and Three Choices." *American Journal of Psychology.* 70, 174-185.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, Mass.: Addison-Wesley.

Goodnow, J.J. 1955. "Determinants of Choice-Distribution in Two-Choice Situations." *American Journal of Psychology.* 68, 106-116.

Hebb, D. O. 1949. *The Organization of Behavior.* Wiley, New York.

Herrnstein, R. 1979. "Derivatives of Matching," *Psychological Review.* 86, 486-495.

Herrnstein, R. and W. Vaughan. 1980. "Melioration and Behavioral Allocation," in *The Allocation of Individual Behavior*, J.E. Straddon (ed.). 143-176. New York. Academic Press.

Herrnstein, R., D. Prelec, Loewenstein, and W. Vaughan (1990). Paper in progress.

Holland, J.H. 1986. "Escaping Brittleness: The Possibilities of General Purpose Machine Learning Algorithms Applied to Parallel Rule-based Systems." In R. Michalski, J. Carbonell, and T. Mitchell (eds.) *Machine Learning: An Artificial Intelligence Approach*, Vol 2. Kaufman, Los Altos, Calif.

Holland, J.H., K.J. Holyoak, R.E.Nisbet, and P.R. Thagard. 1987. *Induction: Process of Inference, Learning, and Discovery.* MIT Press.

Hull, C.L. 1943. *Principles of Behavior.* New York: Appleton-Century Crofts.

Luce, R.D. 1959. *Individual Choice Behavior: a Theoretical Analysis.* New York: Wiley.

Lucas, R.E. 1978. "Asset Prices in an Exchange Economy." *Econometrica*, 46, 1429-1445.

Marcet, A. and T. Sargent. 1989. "Convergence of Least Squares Learning Mechanisms in Self-Referential Linear Stochastic Models." *Journ. Econ. Theory*, 48, 337-368.

Marimon R., E. McGrattan, and T. Sargent. "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents." Santa Fe Institute, Paper 89-004.

Milgrom, P. and Roberts, J. 1989. "Adaptive and Sophisticated Learning in Repeated Normal Form Games." Mimeo, Stanford.

Narendra K., and M.A.L. Thathachar. 1989. *Learning Automata: An Introduction.* Prentice-Hall, Englewood Cliffs, New Jersey.

Nevelson M.B., and R.Z. Hasminskii. *Stochastic Approximation and Recursive Estimation.* Vol. 47. American Math. Soc. Providence.

Pemantle, R. 1988. "Nonconvergence to Unstable Points in Urn Models and Stochastic Approximations." Mimeo. Statistics Department, U.C. Berkeley.

Restle F. and J.G. Greeno. 1970. *Introduction to Mathematical Psychology.* Addison-Wesley.

Rothschild, M. 1974. "A Two-Armed Bandit Theory of Market Pricing," *Journal of Economic Theory*, 9, 185-202.

Sternberg S. 1963. "Stochastic Learning Theory," in *Handbook of Mathematical Psychology.* R.D. Luce, R.R. Bush, E.Galanter, Eds. New York: Wiley.

Tsetlin M.L. 1973. *Automaton Theory and Modeling of Biological Systems.* Academic Press. New York.

Tsypkin, Ya.Z. 1973. *Foundations of the Theory of Learning Systems.* Academic Press. New York.

Turing, A.M. 1956. "Can a Machine Think?" in *The World of Mathematics*, J.R. Newman, Ed. New York: Simon and Schuster, 4, 2009-2123.