Informationally Decentralized Learning Algorithms for Finite-Player, Finite-Action Games of Incomplete Information

William A. Brock
Ramon Marimon
John Rust
Thomas J. Sargent[1]

April, 1988[2]

Abstract:

This paper is an exploratory analysis of the convergence of three informationally decentralized learning algorithms for finite-action, finite-player games of incomplete information. The algorithms involve repeated plays of a static "one-shot" game to allow players to "learn" about the unknown preferences and beliefs of their opponents. The first algorithm, the linear reward-inaction algorithm (LRI), posits that players choose actions according to a probability distribution which is modified after each play of the game according to an exogenous linear updating rule. The LRI algorithm can only converge to vertices of the simplex, but whether or not such vertices can be guaranteed to coincide with pure strategy Nash equilibria is an open question. The second algorithm is a simple "Bayesian" learning algorithm where each player chooses an action to maximize his single-stage posterior expected utility, conditioned on observations of opponents' plays in previous plays of the game. We show that this algorithm is equivalent to an Euler method for numerically integrating a certain differential equation. Numerical examples suggest that the algorithm is not globally convergent- in particular it appears that it cannot converge to a mixed strategy equilibrium. The third algorithm is a "modified Bayesian" learning algorithm in which players generally select the action which maximizes their posterior expected utility, but occasionally experiment by randomly choosing "non-optimal" actions. We show that a single play of this experimentation game can be interpreted as a game of incomplete information, and demonstrate that if the modified Bayesian algorithm converges, it must converge to a locally stable Bayesian Nash equilibrium (BNE) of the incomplete information game. Further, we show that as a parameter σ governing the degree of incomplete information goes to 0, the set of BNE converges to a subset of the NE of the complete information game, including mixed strategy equilibria. Numerical examples show that the while the algorithm does indeed converge to mixed strategy equilibria, the rate of convergence is very slow.

[1] University of Wisconsin-Madison, University of Minnesota, University of Wisconsin and Institute for Empirical Macroeconomics, Federal Reserve Bank of Minneapolis, and Hoover Institution, Stanford University.

[2] This is a very preliminary first draft. Since co-authors have not had time to incorporate their views, all errors, misstatements and crazy conjectures are solely the responsibility of Rust. We appreciate any comments, but please do not quote.

A practical limitation of the concept of Nash equilibrium (NE) is the assumption that players have common knowledge of each others' objective functions. In games played in "real life" situations this assumption is almost never satisfied, yet there is mounting experimental evidence with human subjects (e.g. Smith, 1982) that even without such common knowledge, realized outcomes are surprisingly close to the Nash equilibrium outcome- especially when the games are repeated a sufficient number of times to give the players a chance to "learn". Harsanyi's notion of "Bayesian Nash equilibrium" (BNE) attempts to deal with the problem of incomplete information by assuming that players know only their own objective functions but have prior probability distributions over a pre-defined space of possible objective functions of their opponents. However this only pushes the information problem into a higher dimensional space since Bayesian equilibrium depends on the assumption that players have common knowledge of each others' prior probability distributions. It seems likely that players of real life games will be as ignorant of their opponents' prior beliefs as they are of their objective functions. However even though the common knowledge assumptions are not literally true of real life games, it need not imply that game theory is irrelevant as a positive theory of strategic behavior. There may exist relatively simple learning strategies or learning algorithms that allow players to learn the relevant parts of their opponents' objectives and beliefs, leading to decentralized coordination of behavior and convergence to a Nash equilibrium in repeated plays of the game. Ideally one would like a learning theory which leads to Nash equilibrium behavior in the limit, while at the same time providing a theory of disequilibrium behavior along the path to convergence. At the very minimum, a candidate theory should satisfy three criteria: 1) it should be privacy-preserving recognizing the absence of common knowledge about other players' objective and beliefs, 2) it should be decentralized recognizing the absence of a single agent or "auctioneer" to coordinate the independent actions of the players, and 3) it should be globally convergent recognizing that the players' lack of knowledge may initially lead them to use strategies which may be far from the equilibrium strategies.

These are strong requirements and indeed we do not yet know whether there even exists an algorithm which satisfies all three properties. The related literature on stability of general equilibrium is not very encouraging. For example, the well known counter examples of Scarf show that Walrasian tatonnement fails to be globally (or even locally) convergent. The "instability theorem" of Jordan (1986) goes even further and shows that any continuous strategy adjustment rule for any game-form which Nash implements the Walarasian equilibrium is unstable in the sense that there exist economies for which the adjustment rule fails to converge to the competitive equilibrium, including the instability of tantonnement as a special case. These problems have lead Hahn to conlude in his (1982) survey "a great deal of skilled and sophisticated work has gone into the study of processes by which an economy could attain an equilibrium... While some special models exist, we shall have to conclude that we still lack a satisfactory discriptive theory of the invisible hand". These results are discouraging given the relatively simple structure of economic equilibria, and the fact that they satsify certain desirable properties such as Pareto efficiency (which make such equilibria equivalent to solutions of a certain optimization problem). Nash equilbria are generally not Pareto efficient, so in some sense the problem of finding a informationally decentralized, globally convergent learning rule may be even harder in this case. However while the theoretical obstacles may be large, we believe that there must exist some, possibly very sophisticated, convergent learning process, otherwise how can one explain the widespread convergence to Nash equilibrium in experimental games?

This paper is a modest initial attempt to analyze learning algorithms for finite player, finite action games. We present three learning algorithms which are consistent with Hurwicz's (1960) notion of "informational decentralization" (criteria 2 and 3 above). All of the algorithms generate a sequence of probability distributions for players' actions, and can be viewed as nonlinear stochastic difference equations on the unit simplex. Section 5 presents the linear reward-inaction algorithm (LRI) which has a long history in the literature on cybernetics and learning in biological systems (Bush and Mosteller (1958), Mendel and Fu (1970), and Tsetlin (1973)). Recently Narendra and Wheeler (1986) used the LRI algorithm to establish convergence to a unique pure strategy Nash equilibrium in a class of identical payoff automata games. Narendra and Wheeler proved that by proper choice of a fixed "stepsize parameter λ, one can guarantee convergence to the equilibrium point with probability $1-O(\lambda)$. In section 5 we apply the LRI algorithm to non-identical payoff automata games. We argue that Narendra and Wheeler's proof of convergence, generalizes directly to this case. Unfortunately, their proof is apparently incorrect since the LRI algoirthm must converge to a vertex of the simplex with probability 1, but a non-identical payoff game need not have a pure strategy equilibrium point. Whether or not Narendra and Wheeler's theorem itself is incorrect remains an open question. In section 3 we present a natural "Bayesian" learning algorithm whereby each player chooses a strategy which maximizes his expected utility given his current posterior distribution on the strategies of his opponents. Since the posterior is based only on the observed actions of the other players, the Bayesian learning algorithm is informationally decentralized. We show that the Bayesian learning algorithm can be viewed as an Euler-type method for solving a certain differential equation. We conjecture, but have not yet formally proven, that the Bayesian learning algorithm is locally convergent to a strict pure strategy equilibrium (provided one exists). We also conjecture, but have not proven, that the Bayesian algorithm cannot converge to non-strict pure strategy equilibria or mixed strategy equilibria. Thus, if no strict pure strategy equilibria exist, the algorithm will "wander" forever. In section 4 we present a modified Bayesian learning algorithm which allows the player to "experiment" with new strategies. We relate this "experimentation game" to a game of incomplete information including the original complete information game as a special case. In section 2 we define the incomplete information game, characterize the set of Bayesian Nash equilibria (BNE), and show that as the "experimentation parameter" σ governing the degree of incomplete information tends to zero, the set of BNE of the game of incomplete information converge to a subset of Nash equilibria (NE) of the game of complete information (including mixed strategy equilibria). Using these results, we show in section 4 that for fixed σ, if the modified Bayesian learning algorithm converges with positive probability, it must converge to a locally stable BNE of the game of incomplete information. If the limiting complete information game has a strict pure strategy equilibrium, then for sufficiently small σ the incomplete information game must have a locally stable BNE in a neighborhood of the pure strategy NE. Combining the above results, our results show that by choosing the parameter σ sufficiently small, the modified Bayesian learning algorithm can converge to a point arbitrarily close to a Nash equilibrium point of the original complete information game, including mixed strategy equilibrium points. Thus, by allowing the agents to experiment with possibly non-optimal actions, we can induce enough smoothness into the problem to obtain convergence to mixed strategy equilibrium points.

The idea behind the convergence proofs is to show that the sequence of action probabilities asymptotically follow trajectories of a certain ordinary differential equation (ODE) whose zeros correspond to equilibria of the game. In the case of the Bayesian learning algorithms, we use a theorem of Ljung (1977) to show that the only possible limit points are the locally stable stationary points of the ODE. In the case of the Bayesian learning algorithm the ODE only has zeros at vertices of the simplex, implying convergence to pure strategy equilibria but not mixed strategy equilibria. The ODE for the modified Bayesian algorithm does have zeros in the interior of the simplex, even in the limit as $\sigma \to 0$. Numerical examples show that indeed, the algorithm converges to mixed strategy equilibria. Unfortunately, we have not yet discovered, sufficient conditions to guarantee global convergence with probability 1, so we can't rule out the possiblity that the modified Bayesian algorithm could wander forever. What we need are conditions that 1) guarantee global asymptotic stability to at least one zero of the ODE (or at a minimum, to guarantee that the invariant set of the ODE equals the stable manifold of at least one equilibrium point), and 2) that the algorithm will visit a domain of attraction of a locally stable fixed point sufficiently often to be "sucked in". In analogy to simulated annealing algorithms, we need the algorithm to "bounce around" a lot initially so that it can escape vectorfields of unstable regions and get caught in a domain of attraction of a fixed point, just as simulated annealing algorithm bounces out of vectorfields leading to local minima to eventually get caught in the domain of attraction of the global minimum point. The degree of experimentation, as indexed by σ, provides the noise or "heat" which can lead the algorithm to bounce into a domain of attraction of a fixed point. Based on our computer experiments, we find that by starting out the algorithm with σ large and letting it decrease sufficiently slowly (at rate $1/\sqrt{T}$), we obtain convergence. This version of the modified learning algorithm can be viewed as a stochastic version of a homotopy "path-following" algorithm for computing fixed points. Homotopy path-following algorithms are globally convergent under a "full-rank" assumption (Garcia and Zangwill, 1983). Thus we conjecture that if the certain full rank and stability conditions can be established, we can guarantee global convergence with probability arbitrarily close to 1 by starting the modified algorithm with a value of σ sufficiently large and letting it approach 0 sufficiently slowly (at rate $1/\sqrt{T}$ or $1/\log(T)$). The existing literature on learning algorithms for games is small. Most of the work (Crawford (1974), Lakshmivarahan and Narendra (1982)) focuses on learning algorithms for two-person zero sum games or cooperative games. Narendra and Wheeler (1986) used a simple linear reward-inaction algorithm (LRI) and demonstrated its convergence in a class of identical payoff automata games with unique pure strategy equilibria. Specifically, Narendra and Wheeler showed that by proper choice of a fixed "stepsize" parameter λ, one can guarantee convergence to the equilibrium point with probability $1-O(\lambda)$.

Although it is globally convergent and informationally decentralized, the LRI algorithm has two major shortcomings: 1) it cannot converge to a mixed strategy equilibrium, and 2) it is only minimally consistent with individually rational behavior. Recent work by Fudenberg and Kreps (1988) studies local stability of NE for non-identical payoff games under adjustment or "behavioral rules" similar in spirit to ones used here, but they focus on establishing local stability rather than global stability. In addition, Fudenberg and Kreps focus on the issue of how players come to have common knowledge of their opponents' choice of strategies, rather than how players come to have common knowledge of their opponents' objective functions. Thus, Fudenberg and Kreps assume that the players "know the structure of the game and their opponents' payoffs, but they are uncertain about how their opponents will play" (p. 2). Besides the LRI algorithm, the only other informationally decentralized, globally convergent algorithm we know of is due to Rosen (1965) who developed a "gradient algorithm" for a class of continuous action concave games with unqiue pure strategy equilibria. We are not aware of any informationally decentralized learning algorithms that are globally convergent to mixed strategy equilibria. However even if we succeed in proving global convergence of the modified Bayesian algorithm, we are still far away from having a "successful" theory of learning behavior because all of these algorithms appear to converge far too slowly to be consistent with observed trajectories in human experiments. Specifically, although all the algorithms are able to get into a domain of attraction of a fixed point fairly rapidly, once in a neighborhood the convergence is very slow[1] The problem with the slow local convergence seems to be that 1) the Bayesian procedure for updating beliefs "learns" far too slowly in comparison to what apparently occurs in games with human subjects, and 2) the algorithms use no curvature information of the fixed point mapping.

[1] The rapid initial convergence suggests that these algorithms might offer practical methods for generating starting values for traditional solution algorithms such as Newton's method.

information that is responsible for the rapid local convergence rates of Newton and quasi-Newton methods.

2. Computing Nash Equilibria as Limits of Bayesian Nash Equilibria

The learning algorithms we present can be viewed as stochastic fixed point algorithms. Our discussion in the introduction suggests that fixed points corresponding to mixed strategy equilibria of complete information games may be irregular, i.e. they may not be sufficiently smooth functions of the underlying parameters to insure global convergence. This section paves the way for our subsequent results by showing how the addition of a small amount of incomplete information can smooth out the model, guaranteeing convergence to a mixed strategy equilibrium. Normally one thinks of games of incomplete information as being more complicated objects to solve than games of complete information, so it is ironic that such a trick actually makes it easier to compute an equilibrium of the original game. More formally, we present a class of finite player, finite action games of complete information, and a closely related class of games of incomplete information indexed by a parameter σ. When $\sigma=0$, the two games coincide. We establish that the Bayesian Nash equilibria of the incomplete information games are regular in the sense of Debreu (1976) and Dierker (1970), there are an odd number of such equilibria, and that set of Bayesian equilibria converge to a subset of the set of Nash equilibria of the complete information game as $\sigma\to0$. To simplify notation we present notation for games with only two players, but all our results appear to generalize in a straightforward manner to the case of N-player games.

Consider first the notation for the game of complete information. Player 1 chooses one of N_1 possible actions, and player 2 choose one of N_2 possible actions. When player 1 chooses action i, $i=1,\dots,N_1$, and player 2 chooses action j, $j=1,\dots,N_2$, the players receive rewards $u_1(i,j)$ and $u_2(i,j)$, respectively. We have a game of complete information provided that it is common knowledge that both players are rational and that both know (N_1,N_2,u_1,u_2). Actions (i^*,j^*) constitute a pure strategy Nash equilibrium if

$$
\begin{aligned}
i^{*} &\in \operatorname*{argmax}_{1\leq i\leq N_1} \; u_1(i,j^{*}) \\
j^{*} &\in \operatorname*{argmax}_{1\leq j\leq N_2} \; u_2(i^{*},j).
\end{aligned}
\tag{1}
$$

Let S^N denote the N-1 dimensional simplex, i.e. $S^N=\{p\in R^N \mid p\geq 0,\ \Sigma_i\, p(i)=1\}$. Probability distributions $(p_1,p_2)\in S^{N_1}\times S^{N_2}$ constitute a *mixed strategy Nash equilibrium* if

$$
\begin{aligned}
i \in \mathrm{supp}(p_1) &\Rightarrow i \in \operatorname*{argmax}_{1\leq d\leq N_1} \sum_{k=1}^{N_2} u_1(d,k)p_2(k) \\
j \in \mathrm{supp}(p_2) &\Rightarrow j \in \operatorname*{argmax}_{1\leq d\leq N_2} \sum_{k=1}^{N_1} u_2(d,k)p_1(k),
\end{aligned}
\tag{2}
$$

where $\mathrm{supp}(p_1)=\{i \mid p_1(i)\neq 0\}$. While pure strategy equilibria need not exist. Nash's (1950) theorem guarantees that at least one mixed strategy equilibrium always exists.

Now consider a closely related game of incomplete information. As before there are only two players who have N_1 and N_2 possible actions, respectively. Let $\eta_1\in R^{N_1}$ and $\eta_2\in R^{N_2}$ represent private information of the players and let $\sigma\geq 0$ be a fixed scalar parameter. Suppose that when the players choose actions (i,j) they receive rewards $u_1(i,j)+\sigma\eta_1(i)$ and $u_2(i,j)+\sigma\eta_2(j)$. Although player 1 does not know the type η_2 of his opponent, he has a prior probability density $q_1(\eta_2)$ over the possible types of player 2. Similarly player 2 has prior probability density $q_2(\eta_1)$. We have a game of incomplete information provided that it is common knowledge that each player is rational (i.e. chooses an action to maximize his expected utility given his prior) and both players know $(N_1,N_2,u_1,u_2,q_1,q_2)$. Decision rules $d_1(\eta_1)$ and $d_2(\eta_2)$ constitute a *Bayesian Nash equilibrium* provided

$$
\begin{aligned}
d_1(\eta_1) &= \operatorname*{argmax}_{1\leq i\leq N_1} \int_{\eta_2} [u_1(i,d_2(\eta_2))+\sigma\eta_1(i)]q_1(\eta_2)d\eta_2 \\
d_2(\eta_2) &= \operatorname*{argmax}_{1\leq j\leq N_2} \int_{\eta_1} [u_2(d_1(\eta_1),j)+\sigma\eta_2(j)]q_2(\eta_1)d\eta_1.
\end{aligned}
\tag{3}
$$

Define the (best) response probabilities $\Pi_1(p_2,\sigma)\in S^{N_1}$, $\Pi_2(p_1,\sigma)\in S^{N_2}$ by

$$\Pi_1(p_2,\sigma)(i) = \int_{\eta_1} I\{i = \operatorname*{argmax}_{1\le k\le N_1} \sum_{j=1}^{N_2} u_1(k,j)p_2(j)+\sigma\eta_1(k)\}q_2(\eta_1)d\eta_1 \tag{4}$$

$$\Pi_2(p_1,\sigma)(j) = \int_{\eta_2} I\{j = \operatorname*{argmax}_{1\le k\le N_2} \sum_{i=1}^{N_1} u_2(i,k)p_1(i)+\sigma\eta_2(k)\}q_1(\eta_2)d\eta_2.$$

Implicit in (4) is the assumption that there are no ties, i.e. the argmax is uniquely attained. This will be true provided the players have atomless priors, or in measure-theoretic terms, if q_1 and q_2 are absolutely continuous. The following assumption puts some additional restrictions on the players' prior beliefs to guarantee that the integrals in (3) and (4) are well defined.

(A1) q_1 and q_2 are absolutely continuous probability densities on R^{N_1} and R^{N_2} with unbounded support and finite first moments.

Direct computation of the Bayesian Nash equilibrium defined in (3) seems to be a formidable task since it defines the equilibrium decision rules (d_1,d_2) as a fixed point in the infinite-dimensional space of pairs of functions (f_1,f_2) mapping $R^{N_1}xR^{N_2}$ into $\{1,\ldots,N_1\}x\{1,\ldots,N_2\}$. A shortcut is to calculate the equilibrium via response probabilities. Define a fixed point (p_1,p_2) in the product space $S^{N_1}xS^{N_2}$ by

$$\begin{aligned} p_1 &= \Pi_1(p_2,\sigma) \\ p_2 &= \Pi_2(p_1,\sigma). \end{aligned} \tag{5}$$

Brouwer's theorem guarantees that a fixed point must exist since the simplex is a compact, convex set and the response probabilities are continuous functions of (p_1,p_2). Note that if the random variables η_1 and η_2 have unbounded support by (A1), the fixed point in (5) will always be an interior point of $S^{N_1}xS^{N_2}$ provided $\sigma>0$. This provides an "inward pointing" condition used to establish that (5) has an odd number of fixed points. Intuitively, (p_1,p_2) are the "reduced-form" probability distributions over each agent's actions induced by the decision rules (d_1,d_2) and the priors (q_1,q_2). The fixed point condition (5) says that the players' action probabilities must be mutual best responses. Given the fixed point (p_1,p_2) Theorem 1 shows that it is a simple matter to recover the underlying Bayesian Nash equilibrium decision rules.

Theorem 1 Suppose (A1) holds. Let (p_1,p_2) be a fixed point of (5) and define the decision rules d_1 and d_2 by

$$d_1(\eta_1) = \underset{1 \leq k \leq N_1}{\operatorname{argmax}} \sum_{j=1}^{N_2} u_1(k,j)p_2(j)+\sigma\eta_1(k)$$

(6)

$$d_2(\eta_2) = \underset{1 \leq k \leq N_2}{\operatorname{argmax}} \sum_{i=1}^{N_1} u_2(i,k)p_1(i)+\sigma\eta_2(k).$$

Then (d_1,d_2) are Bayesian Nash equilibrium decision rules.

The great advantage of computing the BNE via (5) is not only that the fixed point problem has been reduced to a finite-dimensional simplex, but also that the fixed points of (5) are regular. This is a consequence of the Williams-Daly-Zachary theorem (for statement and proof, see McFadden, 1981) which shows that under (A1), the response probability functions defined in (4) are continuously differentiable functions of (p_1,p_2,σ). Differentiability implies that the fixed point can be computed as a zero of the function $F(p,\sigma)$ defined by

$$(7) \qquad F(p_1,p_1,\sigma) = \begin{bmatrix} \Pi_1(p_2,\sigma) - p_1 \\ \Pi_2(p_1,\sigma) - p_2 \end{bmatrix}$$

In particular, one can compute the fixed point by Newton's method provided the $(N_1+N_2)x(N_1+N_2)$ Jacobian matrix

$$\partial F(p,\sigma)/\partial p = \begin{bmatrix} -I & \partial\Pi_1(p_2,\sigma)/\partial p_2 \\ \partial\Pi_2(p_1,\sigma)/\partial p_1 & -I \end{bmatrix} \tag{8}$$

is non-singular in a neighborhood of a fixed point. Note that the off-diagonal blocks of the Jacobian, $\partial\Pi_1(p_2,\sigma)/\partial p_2$ and $\partial\Pi_2(p_1,\sigma)/p_1$, have columns that sum to zero due to the adding up restrictions placed on (Π_1,Π_2) as elements of a simplex. Since the set of invertible matrices is an open, dense set, it seems reasonable to suppose that if the Jacobian were not invertible at a particular value of σ, a small perturbation of σ will make it invertible. We conjecture (on the basis of extensive computer calculations), but have not yet been able to prove, the following result:

Conjecture 1: For almost all $p_1 \in S^{N_1}$ and $p_2 \in S^{N_2}$ and $\sigma>0$, $\partial F(p,\sigma)/\partial p$ is invertible.

The regularity and inward pointing condition immediately imply the following results:

Theorem 2: (Lefschetz Fixed Point Theorem) If all fixed points of (5) are regular, then (5) has an odd number of isolated fixed points.

Theorem 3: (Implicit Function Theorem) In a neighborhood about any regular fixed point, there exist continuous functions $p_1(s)$, $p_2(s)$ satisfying:

$$\begin{aligned} 0 &= \Pi_1(p_2(s),s) - p_1(s) \\ 0 &= \Pi_2(p_1(s),s) - p_2(s) \end{aligned} \tag{9}$$

for all s in a neighborhood of $\sigma>0$.

Theorem 3 suggests that the limit points (p_1^*, p_2^*) given by $p_1^*=\lim_{\sigma\to 0}p_1(\sigma)$ and $p_2^*=\lim_{\sigma\to 0}p_2(\sigma)$, will be Nash equilibrium points of the complete information game. Some care is required to show this, however. Note first that Theorem 3 implies that $p_1(s)$ and $p_2(s)$ are continuous functions of s *in a neighborhood of $\sigma>0$*. This neighborhood need not necessarily contain the limit point 0. Secondly it is not immediately obvious that the limiting points, if they exist, will in fact be Nash equilibria of the complete information game. Third, the response functions $\Pi_1(p_2,\sigma)$ and $\Pi_2(p_1,\sigma)$, appear to converge pointwise to vertices of S^{N_1} and S^{N_2} as $\sigma\to 0$. It is not clear, therefore, how it is possible that $p_1(\sigma)$ and $p_2(\sigma)$ can converge to mixed strategy equilibria of the complete information game. Actually, convergence to vertices only occurs at points p_1 and p_2 at which the argmax in (2) is unique.

Lemma 1 Suppose $p_2^*=\lim_{\sigma\to 0}p_2(\sigma)$ is such that for some $1\le i\le N_1$

(10) $$\sum_{j=1}^{N_2} u_1(i,j)p_2^*(j) > \sum_{j=1}^{N_2} u_1(k,j)p_2^*(j)$$

for $k\neq i$, $1\le k\le N_1$. Then

(11) $$\lim_{\sigma\to 0} \Pi_1(p_2(\sigma),\sigma) = e_i = \lim_{\sigma\to 0} \Pi_1(p_2^*,\sigma)$$

where e_i is the i^{th} unit vector.

Lemma 2 Suppose $p_2^*=\lim_{\sigma\to 0}p_2(\sigma)$ is such that action k is non-optimal. then

(12) $$\lim_{\sigma\to 0} \Pi_1(p_2(\sigma),\sigma)(k) = 0 = \lim_{\sigma\to 0} \Pi_1(p_2^*,\sigma)$$

The proof of both lemmas is a simple application of the Lebesgue Dominated Convergence theorem. If p_2 is such that two distinct actions i, j attain the argmax, then Lemma 1 does not hold. In this case $\lim_{\sigma\to 0}\Pi_1(p_2(\sigma),\sigma)$ depends on the distributuion q_1, the precise way in which $p_2(\sigma)$ approaches its limit p_2^*, and possibly on how σ approaches 0 as well. For example, if q_1 is a multivariate extreme value distribution, then the response probability Π_1 is given by then well-known multinomial logit formula:

$$(13) \qquad \Pi_1(p_2,\sigma)(i) = \frac{\exp\{V_i/\sigma\}}{\sum_{j=1}^{N_1} \exp\{V_j/\sigma\}}$$

where

$$(14) \qquad V_j = \sum_{k=1}^{N_2} u_1(j,k)p_2(k).$$

Suppose that $V_i=V_j > V_k$, for $k \neq i,j$. Then (14) implies

$$(15) \qquad \lim_{\sigma\to 0} \Pi_1(p_2,\sigma)(m) = \begin{cases} 1/2 & \text{if } m=i=j \\ 0 & \text{otherwise.} \end{cases}$$

Thus in this case the limiting response probabilities assigns equal probability to the utility-maximizing actions among which player 1 is indifferent. Suppose that the limiting complete information game has a mixed strategy equilibrium (p_1^*, p_2^*) where player 1 plays actions i, j with probabilities $p_1^*(i)=.4$, $p_1^*(j)=.6$. Then the pointwise convergence result in (15) seems to rule out convergence to a mixed strategy equilibrium since $\lim_{\sigma\to 0}\Pi_1(p_2^*,\sigma)(i)=.5\neq p_1^*(i)$. Note, however, that when there is no unqiue argmax it does not follow that $\lim_{\sigma\to 0}p_2(\sigma)=p_2^*$ implies $\lim_{\sigma\to 0}\Pi_1(p_2(\sigma),\sigma)=\lim_{\sigma\to 0}\Pi_1(p_2^*,\sigma)$. For example, suppose that in the example in equation (15) that $p_2(\sigma)$ is such that $V_j=V_i+a\sigma$ and $V_i > V_k$ for all $k \neq i,j$. Then it is easy to see that $\lim_{\sigma\to 0}\Pi_1(p_2(\sigma),\sigma)=\exp\{a\}/[1+\exp\{a\}]$ even though $\lim_{\sigma\to 0}\Pi_1(p_2^*,\sigma)=1/2$. These examples suggest that convergence to a mixed strategy equilibrium is a very delicate matter. However computer experiments show that in fact BNE of the incomplete information game do converge to a mixed strategy equilibrium of the complete information game with no special problem. The following Theorem provides some insight on why this occurs.

Theorem 4 Let (p_1^*, p_2^*) be a cluster point of $(p_1(\sigma), p_2(\sigma))$ as $\sigma \to 0$. Then (p_1^*, p_2^*) is either a pure or mixed strategy equilibrium of the game of complete information.

Proof Since the cartesian product of simplices is compact, we know that at least one cluster point (p_1^*, p_2^*) exists for any sequence of σ's. Thus we are free to choose a subsequence $\{\sigma_t\}$ with $\sigma_t \to 0$ such that $\lim_{t\to\infty}(p_1(\sigma_t), p_2(\sigma_t)) = (p_1^*, p_2^*)$. There are four cases to consider depending on whether the best response functions defined in (2) have unique argmax's when evaluated at (p_1^*, p_2^*). Suppose that at (p_1^*, p_2^*) the argmax in (2) is unique for both players. Then Lemma 1 immediately implies that $p_1^* = \lim_{t\to\infty}\Pi_1(p_2(\sigma_t), \sigma_t) = \lim_{t\to\infty}\Pi_1(p_2^*, \sigma_t) = e_{i^*}$, where $i^* = \text{argmax}_i \Sigma_k u_1(i,k) p_2^*(k)$. Similarly we have $p_2^* = \lim_{t\to\infty}\Pi_2(p_1(\sigma_t), \sigma_t) = \lim_{t\to\infty}\Pi_2(p_2^*, \sigma_t) = e_{j^*}$ where $j^* = \text{argmax}_j \Sigma_k u_2(j,k) p_1^*(k)$. It follows immediately from the definition (1) that $(p_1^*, p_2^*) = (e_{i^*}, e_{j^*})$ is a pure strategy Nash equilibrium point of the complete information game. Now consider the case where at (p_1^*, p_2^*), both players have multiple actions that attain the argmax in (2). Then lemma 2 implies that $p_1^*(k)=0$ and $p_2^*(k')=0$ for any actions k, k' of players 1 and 2 that do not attain the respective argmax's in (2). This immediately implies that p_1^* and p_2^* put all their mass on the set actions that attain the argmax in (2). But this is exactly the condition (3) defining a mixed strategy equilibrium of the complete information game. The proofs of the other two cases (where one player has a unique argmax and the other has multiple argmax's at (p_1^*, p_2^*)), is similar to the above arguments. ∎

Let BNE(σ) denote the set of Bayesian Nash equilibria of the incomplete information game with parameter σ, and let NE denote the set of Nash equilibia of the complete information game. Let $\lim_{\sigma\to 0}$BNE(σ) be the set of all cluster points of

$(p_1(\sigma_t), p_2(\sigma_t))$ for all sequences $\{\sigma_t\}$ with $\sigma_t \to 0$. Then we have:

Corrollary $\lim_{\sigma \to 0} BNE(\sigma) \subseteq NE$.

Under what conditions does $\lim_{\sigma \to 0} BNE(\sigma) = NE$? Since for each $\sigma > 0$ the set $BNE(\sigma)$ consists of an odd number of isolated points in the interior of $S^{N_1} x S^{N_2}$, it seems clear that this will hold in the limit as well. This suggests that in order for $\lim_{\sigma \to 0} BNE(\sigma) = NE$ we will need to restrict our attention to complete information games with only a finite number of isolated equilibria. Actually, we need an additional restriction: each pure strategy equilibrium point (i^*, j^*) must be a strict Nash equilibrium, i.e. i^* and j^* are the unique argmax's for players 1 and 2.

Definition We say that the complete information game is regular if 1) each mixed strategy equilibrium point (p_1^*, p_2^*) is locally isolated, and 2) each pure strategy equilibrium point is a strict Nash equilibrium.

If a particular game is not regular, an arbitrarily small perturbation of the player's payoffs will make it regular: thus regularity is a generic property.

Conjecture 2: If the complete information game is regular, $\lim_{\sigma \to 0} BNE(\sigma) = NE$.

Conjecture 3: If $\lim_{\sigma \to 0} (p_1(\sigma), p_2(\sigma)) = (p_1^*, p_2^*)$ is a pure strategy equilibrium, then it is a strict Nash equilibrium.

The conjectures 2 and 3 arose from numerical experiments with a simple 2x2 game with payoff matrix:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 1 \\ \hline .5 & 1.5 & 1 & 1.5 \end{array}\right]$$

The game has two pure strategy equilibria (1,1) and (2,2) and no mixed strategy equilibria. Notice that in this case there are an even number of equilibria and the (2,2) equilibrium is not strict. In order to see what is going on in the incomplete information game, note that we can collapse the fixed point condition (5) into a single equation:

$$p_1 = \Pi_1(\Pi_2(p_1,\sigma),\sigma). \tag{16}$$

Intuitively, (16) says that an equilibrium point p_1^* must be a best response to players two's best response to p_1^*. Using the adding up restriction that $p_1(1)+p_1(2)=1$, it suffices to plot only the first coordinate of p_1 and Π_1 in figure 1 below. Thus, the numerical results using extreme value priors for η_1 and η_2, show that for any $\sigma > 0$ there is only one Bayesian Nash equilibrium of the incomplete information game and it converges to the (1,1) equilibrium as $\sigma \to 0$. It is quite clear that the Bayesian equilibrium will never converge to the (2,2) equilibrium: since player 1 is indifferent between actions 1 and 2 if player 2 takes action 2, player two will impute response probabilities of (.5 .5) for player 1, but given any randomization between actions 1 and 2, player 2 will find higher expected utility to taking action 1. Thus it will be impossible for the Bayesian Nash equilibrium to converge to the (2,2) equilibrium.

Finally, it is useful to illustrate the above convergence results in the case of limiting mixed strategy equilibria. Consider first the following 2x2 game with payoff matrix

$$\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 3 & 3 \end{array}\right]$$

Notice that this is a regular game with 3 equilibria: 2 pure strategy equilibria (1,1) and (2,2) and a mixed strategy equilibrium (p_1^*,p_2^*) with $p_1^*=p_2^*=(3/4,1/4)$. Figure 2 shows that in this case there are 3 BNE. The pointwise limit of

$\Pi_1(\Pi_2(p_1,\sigma),\sigma)$ is a step function with a discontinuity at the mixed strategy equilibrium point $p_1(1)=3/4$. If we take the convex hull of the left and right limit points of the limit function, we get an upper hemicontinuous correspondence that defines player 1's best response correspondence in the limiting complete information game. This best reponse correspondence has three fixed points corresponding to the 3 Nash equilibria of the game. As $\sigma \to 0$, $\Pi_1(\Pi_2(p_1,\sigma),\sigma)$ undergoes homotopic deformation to its discontinuous limit. Even though the limit function is discontinuous and has no interior fixed points (corresponding to the mixed strategy equilibrium), nevertheless it is still the case that the middle BNE converges to the mixed strategy equilibrium (3/4,1/4).

Now consider the following 2x2 game with payoff matrix

$$\left[\begin{array}{cc|cc} 1 & 2 & 4 & 1 \\ \hline 2 & 1 & 1 & 2 \end{array}\right].$$

Notice that this is a regular game with no pure strategy equilibria and a unique mixed strategy equilibrium $p_1^*=(1/2,1/2)$, $p_2^*=(3/4,1/4)$. Figure 3 shows that the corresponding incomplete information game has a unique equilibrium and that it converges to the unique complete information mixed strategy equilibrium as $\sigma \to 0$.

These results suggest a practical "path following" algorithm for computing mixed strategy equilibria: start with $\sigma_o>0$ and compute an equilibrium of the incomplete information game using Newton's method. Then compute the NE as the limit of the sequence $(p_1^*(\sigma_t),p_2^*(\sigma_t))$ where σ_t is a sequence tending to zero. The path following algorithm uses the solution $(p_1^*(\sigma_t),p_2^*(\sigma_t))$ as a starting point for computing the equilibrium $(p_1^*(\sigma_{t+1}),p_2^*(\sigma_{t+1}))$ at step t+1. Theorem 3 guarantees that this path is smooth, and that by choosing σ_{t+1} sufficiently close to σ_t we can guarantee that $(p_1^*(\sigma_t),p_2^*(\sigma_t))$ is in a domain of attraction of $(p_1^*(\sigma_{t+1}),p_2^*(\sigma_{t+1}))$ so the convergence of Newton's method is guaranteed, provided the Jacobian matrix (8) is nonsingular for all $\sigma>0$. Since the path following algorithm exploits smoothness and the quadratic rate of convergence of Newton's method, we expect it to be much faster than standard combinatorial fixed point algorithms such as Scarf's algorithm and its variants. The only difficult problem is to find conditions to guarantee that(8) is invertible for all (p_1,p_2,σ). The invertibility and the implicit function theorem implies a monotonic path $\{p_1(s),p_2(s)\}$ which can be followed to a solution $(p_1(0),p_2(0))$. If the Jacobian become singular a some $\sigma>0$, then Garcia and Zangwill derive a "homotopy differential equation" which essentially re-parameterizes the problem to allow the path following algorithm to reverse itself in σ-space and still converge to a solution $(p_1(0),p_2(0))$. Unfortunately a path following alogrithm based on solution to the homotopy differential equations does not appear to have a simple interpretation in terms of a learning algorithm with experimentation.

3. Convergence of the Bayesian Learning Algorithm

The "Bayesian" learning algorithm is also known as "solution by fictious play" is known to have susequences which converge to NE in 2-person zero-sum games (cf. Owen, 1982). The algorithm works as follows. The one-shot game between players 1 and 2 is repeated T times. In plays 1,...,T player 1 observes actions $i_2(1),....i_2(T)$ of player 2 and player 2 observes actions $i_1(1),....,i_1(T)$ of player 1. Let $d_1(t)=e_{i_1(t)}$ and $d_2(t)=e_{i_2(t)}$, where e_i is the i^{th} unit vector. At play t with data $\{d_1(1),....,d_1(t-1)\}$, player 2 forms an estimate (histogram) of player 1's action probability vector p_1 as follows

$$(17) \qquad \hat{p}_1(t-1) = \sum_{s=0}^{t-1} d_1(s)/t.$$

Similarly player 1 forms an estimate of player 2's action probability vector p_2

$$(18) \qquad \hat{p}_2(t-1) = \sum_{s=0}^{t-1} d_2(s)/t.$$

Here $d_1(0)\epsilon\, \text{int}(S^{N_1})$ and $d_2(0)\epsilon\, \text{int}(S^{N_2})$ are initial values which can be interpreted as "initial beliefs", or alternatively as the expected values of each player's prior probability distributions $f_1(p_2)$, $f_2(p_1)$ over the possible values of their opponents' action probability vector. Specifically, if player 1 has a Dirichlet prior over S^{N_2}

$$(19) \quad f_1(\tilde{p}_2 | d_2(0)) = \frac{\Gamma(d_2(0)(1)+ \ . \ . \ . + d_2(0)(N_2))}{\Gamma(d_2(0)(1)) \ . \ . \ . \ \Gamma(d_2(0)(N_2))} \text{ x } p_2(1)^{[d_2(0)(1)-1]} \ . \ . \ . \ p_2(N_2)^{[d_2(0)(N_2)-1]},$$

then

$$(20) \qquad E\{\tilde{p}_2 | d_2(0)\} = d_2(0).$$

Similarly if player 2 has a Dirichlet prior over S^{N_1},

$$(21)\quad f_2(\tilde{p}_1 | d_1(0)) = \frac{\Gamma(d_1(0)(1) + \ . \ . \ . + d_1(0)(N_1))}{\Gamma(d_1(0)(1)) \ . \ . \ . \ \Gamma(d_1(0)(N_1))} \times p_1(1)^{[d_1(0)(1)-1]} \ . \ . \ . \ p_1(N_1)^{[d_1(0)(N_1)-1]},$$

then

$$(22)\qquad E\{\tilde{p}_1 | d_1(0)\} = d_1(0).$$

Since Dirichlet is conjugate to the multinomial distribution, the posterior distributions

$$f_1(\tilde{p}_2 | d_2(0),\ldots,d_2(t-1))$$

$$f_2(\tilde{p}_1 | d_1(0),\ldots,d_1(t-1))$$

are Dirichlet with expected values given by:

$$(23)\qquad E\{\tilde{p}_2 | d_2(0),\ldots,d_2(t-1)\} = \hat{p}_2(t-1)$$

$$(24)\qquad E\{\tilde{p}_1 | d_1(0),\ldots,d_1(t-1)\} = \hat{p}_1(t-1).$$

Thus the updating rules (17) and (18) can be interpreted as posterior means generated by a Bayesian learning procedure. Given the players' posterior beliefs (of which the posterior means are sufficient statistics), each player chooses an action to maximize expected utility

$$(25)\qquad i_1(t) = \operatorname*{argmax}_{1 \le i \le N_1} \sum_{j=1}^{N_2} u_1(i,j)\hat{p}_2(t-1)(j), \quad d_1(t) = e_{i_1(t)}$$

$$(26) \qquad i_2(t) = \underset{1\le j\le N_2}{\operatorname{argmax}} \sum_{i=1}^{N_1} u_2(i,j)\hat{p}_1(t-1)(i), \qquad d_2(t) = e_{i_2(t)}.$$

In the case an argmax in (25) or (26) is not unique, we assume players use some tie-breaking rule to determine a unique chosen alternative (such as randomization) so that $\forall t>0$ $d_1(t)$ and $d_2(t)$ are vertices of S^{N_1} and S^{N_2}. Now, re-write (17) and (18) recursively as

$$(27) \qquad \hat{p}_1(t) = \hat{p}_1(t-1) + \frac{1}{t+1} [\ d_1(t) - \hat{p}_1(t-1)\ ]$$

$$(28) \qquad \hat{p}_2(t) = \hat{p}_2(t-1) + \frac{1}{t+1} [\ d_2(t) - \hat{p}_2(t-1)\ ].$$

The Bayesian learning algorithm consists of the system (25), (26), (27), and (28), a deterministic system of nonlinear difference equations. Since the product of simplices is compact, the sequence $\{\hat{p}_1(t),\hat{p}_2(t)\}$ must have a convergent subsequence. Suppose it has a cluster point (p_1^*,p_2^*). Then from (25) and (26), it is easy to see that (p_1^*,p_2^*) is a Nash equilibrium point: if $i\varepsilon\text{supp}(p_1^*)$, then i must be an argmax of (25) infinitely often, so it must be an argmax in the limit against p_2^*. Similarly if $j\varepsilon\text{supp}(p_2^*)$, then j must be an argmax of (26) infinitely often, so it must be an argmax in the limit against p_1^*. But this is precisely the condition (2) for (p_1^*,p_2^*) to be a Nash equilibrium point.

Lemma: Any cluster point of the modified Bayesian learning algorithm is a Nash equilibrium.

This is an extremely weak convergence result since the sequence need not stay at (p_1^*,p_2^*), but continually cycle through the simplex. Given a fixed tie-breaking rule, we will almost never expect the sequence (27) and (28) to converge to a mixed strategy equilibrium point (p_1^*, p_2^*), since the probability distribution over the elements in the set of argmax's need not correspond to the distribution required to support the mixed strategy equilibrium. Thus, we conjecture that the Bayesian learning algorithm will not converge to a mixed strategy equilibrium point, expect in the event of a stochastic tie-breaking rule that happens to coincide with the Nash equilibrium probability distributions.

The Bayesian learning algorithm can be interpreted as using Euler's method (with decreasing stepsize h=1/(t+1)) to solve the ODE

$$(29) \qquad \begin{bmatrix} dp_1(t)/dt \\ dp_2(t)/dt \end{bmatrix} = \begin{bmatrix} d_1(p_2(t)) & - & p_1(t) \\ d_2(p_1(t)) & - & p_2(t) \end{bmatrix} \equiv F(p)$$

where we use the (inconsistent) notation $d_1(p_2(t))$ and $d_2(p_1(t))$ to emphasize the fact that the action probabilities depend on t only through $p_1(t)$ and $p_2(t)$ (see (25) and (26)).

Theorems from numerical analysis on the convergence of Euler's method show that with decreasing stepsize h=1/(1+t) the difference between the trajectories of Euler's method (27) and (28) and the actual solution to the ODE (29) are O(h) at a predetermined point t_0. However such theorems tell us little about the asymptotic behavior of the trajectories as $t_0 \to \infty$. Although non-stochastic, the system (25), (26), (27) and (28) fits within the class of recursive stochastic algorithms whose asymptotic behavior was analyzed by Ljung (1977), who handles non-stochastic systems as a special case. Ljung's Theorem 1 establishes that under weak regularity conditions, if the ODE (29) has an invariant set D_c with domain of attraction D_A, and if the trajectories of the learning algorithm visit a closed subset of D_A infinitely often, then $\hat{p}(t)=(\hat{p}_1(t),\hat{p}_2(t))$ converges to D_c as $t \to \infty$. Recall that the invariant set D_c is the set of absorbing states of the ODE, i.e. it is the set of points D_c such that once a trajectory p(t) of (29) enters D_c it never leaves D_c. An obvious subset of D_c are the equilibrium points, i.e. the zeros of F(p). Clearly any zero of F(p) is a pure strategy NE of the complete information game: since $d_1(t)$ and $d_2(t)$ are vertices, it is obvious that no mixed strategy equilibria can be a zero of F(p). Ordinarily, the zeros of F(p) are strict pure strategy equilibria, but it may happen that by coincidence of the tie-breaking rule a non-strict pure strategy NE may be a zero. Thus, the invariant set D_c to which the learning algorithm converges (assuming that the assumptions of the Ljung's theorem can be verified) includes pure strategy NE of the complete information game. The problem is that D_c may include other points which are not equilibria, such as closed orbits or limit cycles. One needs a "global asymptotic stability" condition to guarantee that D_c consists only of zeros of F(p). Since we have not been able to define a suitable Liapunov function to guarantee global stability, we proceeded to analyze a series of examples on the computer.

Consider the 2x2 game matrices presented in section 2. Recall that the first example had the payoff matrix

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 1 \\ \hline .5 & 1.5 & 1 & 1.5 \end{array}\right]$$

with one strict pure strategy equilibrium, one non-strict pure strategy equilibrium, and no mixed strategy equilibria. As expected, Table 1 shows that the algorithm converges to the pure strategy equilibrium. Starting the learning algorithm with initial conditions $d_1(0)$ and $d_2(0)$ arbitrarily close to the (1, 1.5) equilibrium were not successful: the algorithm invariable converged to the (1,2) equilibrium (note that the tie-breaking rule used in this case was to to choose alternative 1 in case both alternatives yielded the same expected utility). Table 2 reports numerical results for the learning algorithm on the game with payoff matrix

$$\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 3 & 3 \end{array}\right]$$

which has two strict pure strategy equilibria and a mixed strategy equilibrium at $p_1^*=p_2^*=(3/4,1/4)$. Depending on initial conditions we were able to generate trajectories for $\{p(t)\}$ that converged to either the (1,1) equilibrium or the (3,3) equilibrium, but the algorithm never converged to the mixed strategy equilibrium. Finally, consider the game with payoff matrix

$$\left[\begin{array}{cc|cc} 1 & 2 & 4 & 1 \\ \hline 2 & 1 & 1 & 2 \end{array}\right]$$

which has no pure strategy equilibria and a unique mixed strategy equilibrium $p_1^*=(1/2,1/2)$ and $p_2^*=(3/4,1/4)$. As can be seen from Table 3, the learning algorithm appeared to drift about a neighborhood of the equilibrium (p_1^*,p_2^*), but even after hundreds of thousands and even millions of iterations, it showed no signs of actually converging to the equilibrium.

Overall the numerical results demonstrate the slow rate of convergence of Bayesian learning, taking tens of thousands of iterations to converge to a small neighborhood of a strict equilibrium. It is evidence that the Bayesian procedure for updating beliefs may be somewhat "stupid" in the sense that after viewing thousands of observations of one'sopponent taking the same action, the algorithm is unwilling to take of leap of faith and conclude that the opponent is actually playing the action with probability 1.

4. Convergence of a Modified Bayesian Learning Algorithm

Section 3 showed that the simple Bayesian learning may converge to a strict pure strategy NE if one exists, but it is an open question whether it can converge to a mixed strategy equilibrium (although the last numerical example of section 3 suggests that the algorithm may "hover about" a mixed strategy equilibrium without ever completely converging to it). Ljung's theorems indicate that the only possible limit points of the learning algorithm are the locally stable zeros of the ODE dp(t)/dt=F(p(t)) defined in (29). Since mixed strategy equilibria cannot be zeros of F(p), and since F is not differentiable at a mixed strategy equilibrium point, Ljung's theorem appears to rule out the possibility of convergence to a mixed strategy equilibrium. Unfortunately, Ljung's theorem 3 cannot be used because it assumes that F(p) is continuously differentiable in a neighborhood of any limit point p*. The analysis of section 2 indicates that by adding a small amount of incomplete information indexed by σ, one could obtain equilibria p* which are arbitrarily close to equilibria of the complete information game as $\sigma \to 0$, and which are all regular equilibria. This implies that the corresponding differential equation $F(p,\sigma)$ defined in (7) is continuously differentiable in a neighborhood of any fixed point. Thus Ljung's theorem 3 applies, and we can conclude that if the modified Bayesian learning algorithm (to be defined below) converges with positive probability, it must converge to a locally stable zero of $F(p,\sigma)$, i.e. a Bayesian Nash equilibrium of the incomplete information game defined in section 2. Section 2 also suggested a "path following" algorithm that leads to a NE of the complete information game as $\sigma \to 0$, including mixed strategy equilibria. This suggests running the modified Bayesian learning algorithm allowing σ to approach zero as a function of t will cause the learning algorithm to follow a path to a NE, including mixed strategy equilibria. Numerical examples demonstrate that this is indeed the case, but unfortunately, it appears very difficult to prove that such an algorithm must be globally convergent with probability 1.

The modified Bayesian learning algorithm uses the same updating rules for the probability estimates $\hat{p}_1(t)$ and $\hat{p}_2(t)$ given in (17) and (18), but now the agents' decisions are random variables $\tilde{i}_1(t)$ and $\tilde{i}_2(t)$ defined by

$$(33) \qquad \tilde{i}_1(t) = \operatorname*{argmax}_{1\le i\le N_1} \sum_{j=1}^{N_2} u_1(i,j)\hat{p}_2(t-1)(j)+\sigma\tilde{\eta}_1(t)(i), \quad \tilde{d}_1(t) = e_{\tilde{i}_1(t)}$$

$$(34) \qquad \tilde{i}_2(t) = \operatorname*{argmax}_{1\le j\le N_2} \sum_{i=1}^{N_1} u_2(i,j)\hat{p}_1(t-1)(i)+\sigma\tilde{\eta}_2(t)(j), \quad \tilde{d}_2(t) = e_{\tilde{i}_2(t)}$$

where $\sigma>0$ is a fixed parameter and $\tilde{\eta}_1(t)$ is an IID draw at play t from q_1 and $\tilde{\eta}_2(t)$ is an IID draw from q_2. Under (A1) of section 2 the argmax's in (33) and (34) are unique with probability 1 so the decision rules $\tilde{d}_1(t)$ and $\tilde{d}_2(t)$ are well-defined and are elements of $S^{N_1}xS^{N_2}$ with probability 1 without the need for exogenous tie-breaking rules. One interpretation of (33) and (34) is that each player maximizes expected utility, but each accounts for private information $\tilde{\eta}_1(t)$ and $\tilde{\eta}_2(t)$ that randomly affects their decisionmaking each period.[*] Alternatively, one could think (33) and (34) as defining a choice process whereby the players try to maximize expected utility on average, but occasionally "experiment" by randomly choosing alternative actions. Note that each replication of this "experimentation game" is not the same as repeated plays of the incomplete information game of section 2, since the IID assumption implys that new "types" are randomly drawn in each replication.

The modified Bayesian learning algorithm defined by equations (27), (28), (33) and (34) is a nonlinear system of stochastic difference equations, a system which also fits within the class of recursive stochastic algorithms analyzed by Ljung (1977). Ljung's results suggest that the asymptotic behavior of the modified Bayesian learning algorithm will be governed by the ODE

$$(35)\qquad \begin{bmatrix} dp_1(t)/dt \\ dp_2(t)/dt \end{bmatrix} = \begin{bmatrix} \Pi_1(p_2(t),\sigma) & - & p_1(t) \\ \Pi_2(p_1(t),\sigma) & - & p_2(t) \end{bmatrix} \equiv F(p,\sigma).$$

We know from section 2 that for almost all $\sigma>0$, $F(p,\sigma)$ has an odd number of locally isolated zeros and each zero is a BNE of the game of incomplete information. Further we know that Π_1 and Π_2 are continuously differentiable functions of p_1 and p_2, so Ljung's Theorem 3 applies.

Definition: A BNE p^* is locally stable if $F(p^*)=0$ and the Jacobian $\partial F(p^*,\sigma)/\partial p$ has eigenvalues all of whose real parts are negative.

Theorem 5: If the Modified Bayesian algorithm converges with positive probability, it must converge to a locally stable BNE of the game of incomplete information.

Theorem 5 leaves open the possibility that the modified Bayesian algorithm may not converge at all. There are three situations under which this can occur: 1) $F(p,\sigma)$ may not have any locally stable zeros, 2) the invariant set D_c of the ODE (35) may not equal the stable manifold of the set of zeros of $F(p,\sigma)$ (i.e. the equilibria may not be globally asymptotically stable), and 3) even if the equilibria are globally asymptotically stable, the algorithm may not visit a domain of attraction of D_c inifinitely often (and will thus fail to be "sucked in"). If is tempting to search for a Liapunov function $V(p)$ similar to the one constructed in the case of the Bayesian learning algorithm, thus guaranteeing global asymptotic stability of (35). Unfortunately the Lefschetz Index theorem suggests that any such effort is bound to fail for the following reason. Existence of a Liapunov function $V(p)$ implies that each zero of $F(p,\sigma)$ is locally stable. However since eigenvalues come in conjugate pairs, this would imply that each equilibria has the same index (since the determinant of the Jacobian equals the product of its eigenvalues). However the Lefschetz Index Theorem would then imply that BNE are always unique, but we know from the numerical examples of section 2 that this is not the case. Therefore it immediately follows that at least one BNE is locally unstable. We formalize this argument as

Lemma 3: At least one BNE must be locally unstable.

We do not yet know whether there are sufficient conditions guaranteeing that at least 1 BNE is locally stable. However even if we were to establish this, we would still face the difficult task of showing that the invariant set D_C of (35) did not contain closed orbits that could capture {p(t)} and lead to nonconvergence of the process. Finally even if we could demonstrate this, we would still have to prove that {p(t)} visited a domain of attraction of a stable BNE infinitely often.

At this point there is little else we can say from a theoretical level about the convergence of the modified Bayesian learning algorithm. To get some insight on its properties we resort to numerical computation of several examples. Table 10 compares the trajectories of the modified Bayesian learning algorithm to the trajectories of the Newton path-following algorithm when σ is allowed to tend to zero according to the sequence $\sigma(t)=5/\sqrt{t}$. The table shows approximate aggreement of the trajectories, tending to support our claim that the modified Bayesian learning algorithm can be viewed as a stochastic path-following alogithm. After 30,000 iterations with $\sigma(30000)=.0288$, both algorithms yielded probability vectors close to a mixed strategy equilibrium $p_1^*=(.5,0,.5)$, $p_2^*=(1/3,1/3,1/3)$. Note, however, that the off-diagonal blocks of the Jacobian are tending to infinity since the best response correspondence is non-differentiable at a mixed strategy point. These growing off-diagonal parts induce growing oscillatory characteristic roots of the Jacobian matrix making convergence increasing difficult as $\sigma(t)\rightarrow0$ even though the real parts of the characteristic roots are converging to -1. Table 11 provides an illustration of possible non-convergence of the modified learning algorithm arising from problems of oscillatory roots that occur when the initial value of σ is chosen too small. Here $\sigma(1)=.5$, and the algorithm showed no signs of converging even after 30,000 iterations. The trajectories appeared to cycle about the non-strict pure strategy equilibrium point, gradually approaching it, but being suddenly repelled if it approached to closely. Table 12 shows the corresponding trajectory when the initial value of σ is sufficiently high to avoid getting trappped into cycles. After 60,000 iterations the algorithm appears to have settled down to the unique mixed strategy equilibrium of the game.

5. Convergence of the LRI Learning Algorithm

The linear reward-inaction algorithm has been described in many places (see, e.g. Lakshimvarahan , (1981)), and since this section closely follows Narendra and Wheeler (1986), we will only provide only a brief sketch of the algorithm and the convergence theorem and refer the reader to the above references for further details. The LRI algorithm is an adjustment rule for stochastic automata games. The one shot game is played repeatedly, and at play t players 1 and 2 take actions $i_1(t)$ and $i_2(t)$ as random draws from probability distributions $p_1(t) \in S^{N_1}$ and $p_2(t) \in S^{N_2}$, respectively. Suppose we normalize the players' payoffs so that for all i,j we have $0 \le u_1(i,j) \le 1$ and $0 \le u_2(i,j) \le 1$. Define $d_1(t)=e_{i_1(t)}$ and $d_2(t)=e_{i_2(t)}$, i.e. $d_1(t)$ and $d_2(t)$ are the vertices of S^{N_1} and S^{N_2} corresponding to actions $i_1(t)$ and $i_2(t)$. The LRI algorithm is a simple linear rule for updating the players' action probabilities given by

$$p_1(t+1) = p_1(t) + \lambda u_1(i_1(t),i_2(t))[d_1(t) - p_1(t)]$$

(36)

$$p_2(t+1) = p_2(t) + \lambda u_2(i_1(t),i_2(t))[d_2(t) - p_2(t)]$$

where $\lambda \in (0,1)$ is a fixed stepsize parameter. It follows from (36) that $p_1(t+1) \in S^{N_1}$ and $p_2(t+1) \in S^{N_2}$ for any $i_1(t)$, $i_2(t)$. According to (36), if player 1 takes action $i=i_1(t)$ at play t then at play t+1 his probability of taking action i always increases, and the probability of taking action $j \neq i$ decreases to make $p_1(t+1)$ sum to 1. The magnitude of the increase in the probability of taking action i, $[p_1(t+1)(i)-p_1(t)(i)]$, depends on the size of the realized reward $r=u_1(i_1(t),i_2(t))$: if r is close to 0 then the probability of taking action i is increased very little, if r is close to 1 it is increased a lot. Thus LRI automata respond positively to good payoffs, but do not react adversely to bad payoffs in the sense of reducing the probability of taking action i. Other learning schemes, such as the linear reward-penalty scheme, force the player to reduce his probability of taking action i when he receives a bad payoff. Note that while LRI automata respond to payoffs, they do so only in a very indirect way: they are not "rational" in the sense of choosing actions to maximize so objective function. The learning algorithm can be interpreted as a strategy followed by relatively stupid players, who choose randomly but show some myopic response to favorable outcomes. Note also that the LRI algorithm satisfies the requirement of informational decentralization: neither player needs to know the payoff function of his opponent.[1]

The convergence analysis of the LRI algorithm begins with the observation that (36) induces a Markov process on $S^{N_1}xS^{N_2}$ with stationary transition probabilities. Thus, $\{p_1(t),p_2(t)\}$ is a Markov process whose absorbing states consist of the vertices of $S^{N_1}xS^{N_2}$. It is easy to see from (36) that unless u_1 and u_2 are identically 0, no point of the interior of $S^{N_1}xS^{N_2}$ is absorbing. Thus, to prove convergence of the LRI algorithm we need to show that the process eventually hits one of the absorbing vertices with probability 1. This result follows from the theory of compact Markov processes:

Theorem 5 With probability 1 the Markov process $\{p_1(t),p_2(t)\}$ defined by (36) converges to a vertex of $S^{N_1}xS^{N_2}$.

Proof: The result follows immediately from Theorem 4.3 of Norman (1972) provided $\{p_1(t),p_2(t)\}$ is a compact Markov process. Since $S^{N_1}xS^{N_2}$ is a compact state space, a sufficient condition for $\{p_1(t),p_2(t)\}$ to be compact is that the process is distance diminishing. To verify this condition in the present case, write the algorithm (36) in vector form as $p(t+1)=T(p(t),\lambda,x(t))$, where $x(t)=(i_1(t),i_2(t))$. Then $\{p_1(t),p_2(t)\}$ is distance diminishing (i.e.. T is a stochastic contraction) if

[1] In fact, the players need not even know their own payoff functions. All the players need to know is their realized payoffs $u_1(i_1(t),i_2(t))$ and $u_2(i_1(t),i_2(t))$.

(37) $$\sup_{\substack{p^1 \neq p^2 \\ x,\ \lambda}} \| T(p^1,\lambda,x) - T(p^2,\lambda,x) \| < \| p^1 - p^2 \| .$$

It is straightforward to verify from (36) that T indeed satisfies the distance diminishing condition, so it follows that $\{p_1(t),p_2(t)\}$ converges to a vertex of $S^{N_1}xS^{N_2}$ with probability 1. ∎

It follows immediately that the LRI algorithm is only capable of converging to pure strategy Nash equilibrium points, and in fact, it is only capable of converging to strict pure strategy NE. In order to show that $\{p_1(t),p_2(t)\}$ converges to a strict NE, Narendra and Wheeler (1986) showed that by choosing the stepsize parameter λ sufficiently small, the trajectories of $\{p_1(t);p_2(t)\}$ will follow the trajectories of a certain ODE, whose stable stationary points are the vertices of $S^{N_1}xS^{N_2}$ corresponding to the strict NE. Define the function W(p) mapping $S^{N_1}xS^{N_2}$ into itself by:

(38) $$W(p) = E\{p(t+1)-p(t) \mid p(t)=p\}/\lambda .$$

From the definition of the LRI algorithm in (36) it follows that the vertices of $S^{N_1}xS^{N_2}$ are zeros of W(p). Let f(t) be the solution to the ODE $df(t)/dt=W(f(t))$ subject to the initial condition $f(0)=p\in int(S^{N_1}xS^{N_2})$. Then following arguments of Lakshmivarahan (1981) one can show that there exist constants k_1 and k_2 such that

(39) $$E\{p(n) - f(n\lambda)\} = k_1\lambda$$
$$E\{[p(n) - f(n\lambda)]^2\} = k_2\lambda, \quad n=0,1,2,\ldots,\ p(0)=p.$$

and where $f(n)=f(n\lambda)$. Since $\{p(n)\}$ converges to a vertex with probability 1, (39) implies that the ODE $df(t)/dt=W(f(t))$ is globally asymptotically stable. Narendra and Wheeler define a vertex e* to be a stable stationary point is

$$(40) \qquad \partial W(p)(i)/\partial p(i) < 0 \quad \forall i.$$

The crux of Narendra and Wheeler's convergence result is the (tedious, but straightforward) demonstration that e* is a vertex of $S^{N_1}xS^{N_2}$ corresponding to a strict Nash equilibrium if and only if e* is a stable stationary point of the ODE. Narendra and Wheeler take this result as implying that when the stepsize λ is sufficiently small, the trajectories of $\{p_1(t),p_2(t)\}$ are arbitrarily close to the trajectories of the ODE $df(t)/dt=W(f(t))$, which can only converge to a locally stable zero of W, the strict pure strategy equilibrium point.

However there is a paradox in the Narendra Wheeler proof: their stability arguments do not depend at all on the fact that the game they originally analyzed was an identical payoff game with a unique pure strategy equilibrium point. Thus, if their convergence proof is correct, one can apply it to establish the convergence of the LRI algorithm in non-identical payoff games. However here is where the paradox arises: consider the behavior of the LRI algorithm in a non-identical payoff game with no pure strategy equilibria. Their stability argument then shows that each vertex is locally unstable. Thus by choosing λ sufficiently small, the trajectories of $\{p_1(t),p_2(t)\}$ will follow the ODE trajectories arbitrarily closely, and hence will never converge to a vertex. However this contradicts Thereom 5 which shows that with probability 1, the trajectories must converge to a vertex.

The problem in the Narendra-Wheeler argument seems to be their definition of "stability" (40). We can seen no reason why (40) should imply global asymptotic stability (or even local stability), and conversely, why the reverse inequality should imply that the vertex is locally unstable. If there is no necessary connection between condition (40) and stability. there is no guarantee that trajectories of the ODE converge only to pure strategy equilibrium points. implying that the LRI algorithm could fail to learn even in identical payoff games.

In extensive computer simulations, the LRI algorithm usually converged to a pure strategy equilibrium in non-identical payoff games, but occasionally it would get trapped at a non-equlibrium vertex. Such examples are not proof that the LRI algorithm can fail to converge, however, because one can always argue that by choosing a smaller stepsize parameter λ the new trajectories might converge to the Nash equilibrium. Although the Narendra-Wheeler proof seems to be incorrect, it is an open question whether their Theorem is true or false.

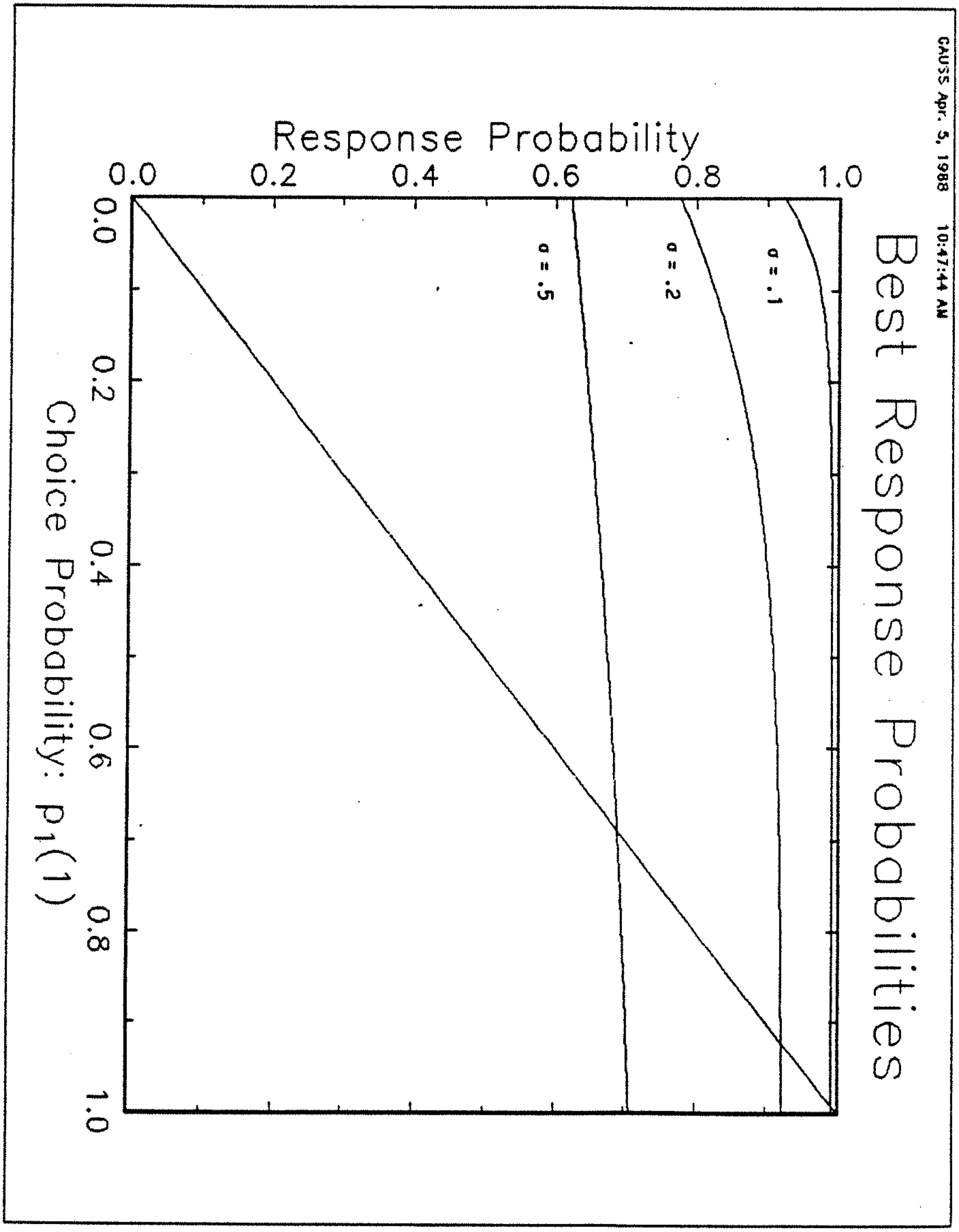
GAUSS Apr. 5, 1988 10:47:44 AM
Best Response Probabilities
Response Probability
0.0
0.2
0.4
0.6
0.8
1.0
σ = .1
σ = .2
σ = .5
0.0
0.2
0.4
0.6
0.8
1.0
Choice Probability: p1(1)

FIGURE 1

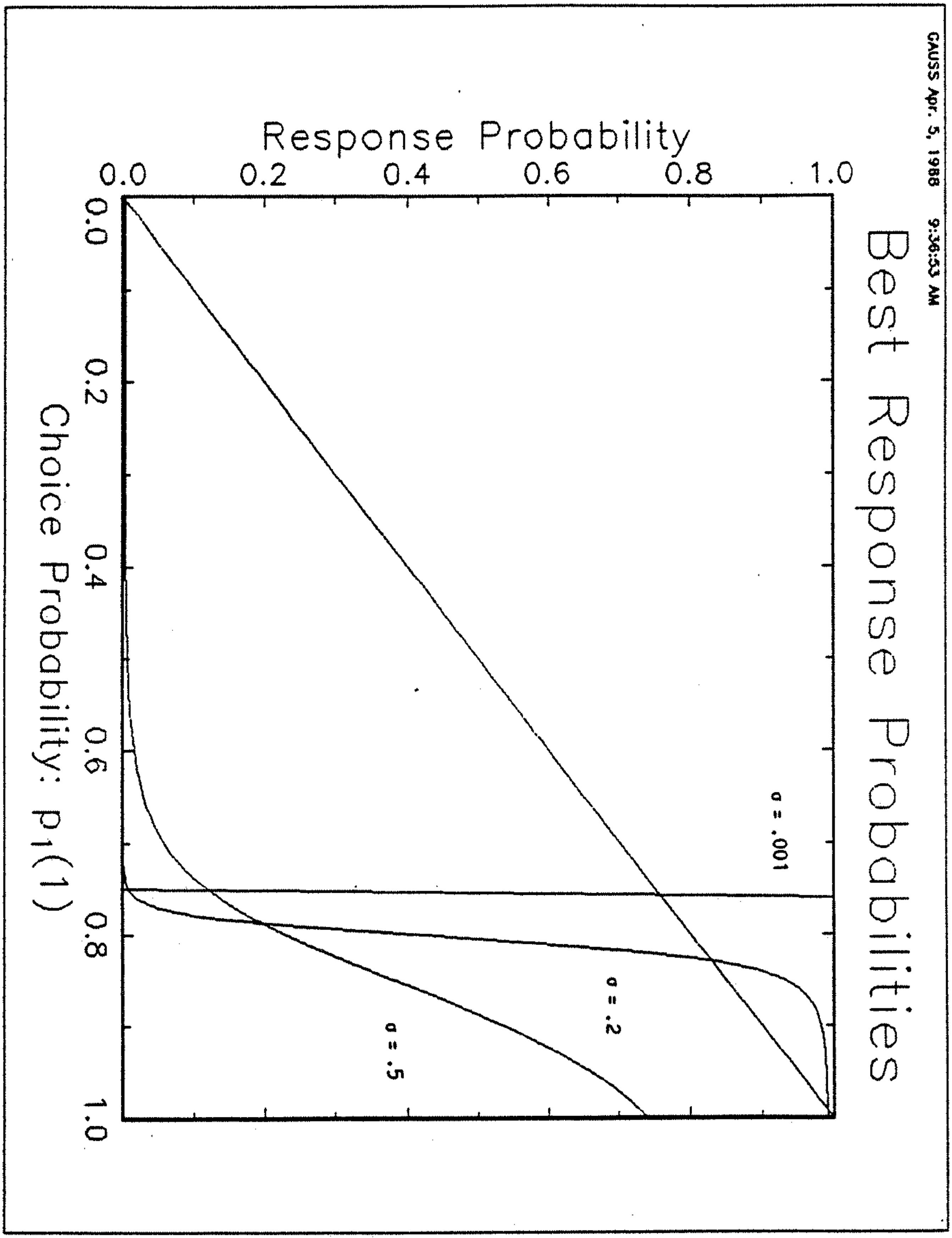
GAUSS Apr. 5, 1988 9:36:53 AM
Best Response Probabilities
Response Probability
0.0
0.2
0.4
0.6
0.8
1.0
Choice Probability: $p_1(1)$
0.0
0.2
0.4
0.6
0.8
1.0
σ = .001
σ = .2
σ = .5

FIGURE 2

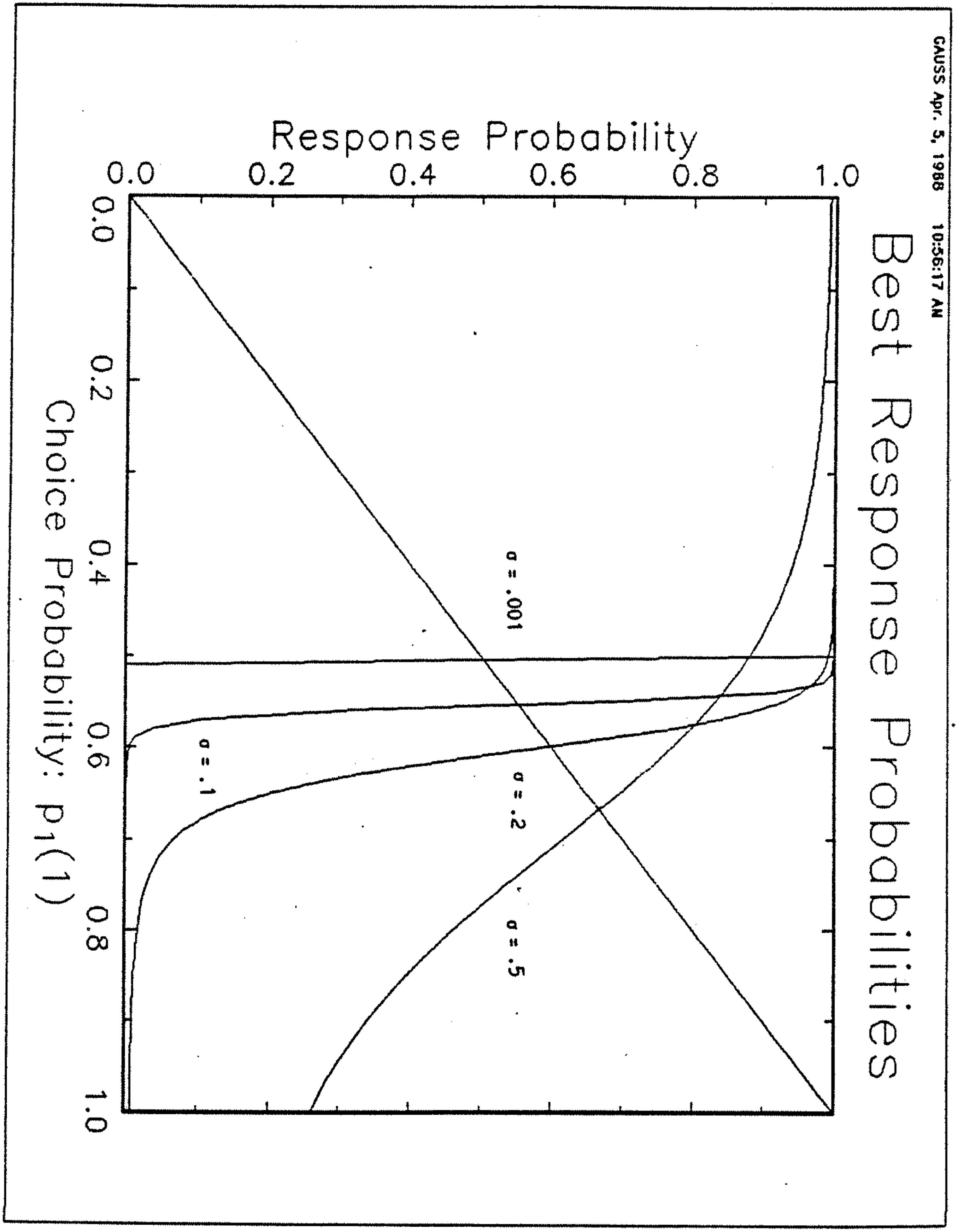
GAUSS Apr. 5, 1988 10:56:17 AN
Best Response Probabilities
Response Probability
0.0
0.2
0.4
0.6
0.8
1.0
Choice Probability: $p_1(1)$
0.0
0.2
0.4
0.6
0.8
1.0
σ = .001
σ = .1
σ = .2
σ = .5

FIGURE 3

Table 1: Convergence of Bayesian Learning Algorithm in Example 1

```
    enter 1 to enter new choice probabilities
    enter 2 to start from center of simplex
    enter 3 to start from existing values
    enter 4 to start from random initial probabilities
 enter (1,2,3): 4

 initial action probabilities
 p1
       0.047509       0.952491
 p2
       0.183894       0.816106

 iteration    100
 p1
       0.990569       0.009431
 p2
       0.99192        0.00808

 iteration    500
 p1
       0.998099       0.001901
 p2
       0.998371       0.001629
 iteration   1000
 p1
       0.999048       0.000952
 p2
       0.999185       0.000815
 iteration   2000
 p1
       0.999524       0.000476
 p2
       0.999592       0.000408
 iteration   5000
 p1
        0.99981        0.00019
 p2
       0.999837       0.000163
 iteration  10000
 p1
       0.999905   9.523959E-005
 p2
       0.999918   8.160245E-005
 p2
       0.999959   4.100830E-005
 iteration  20000
 p1
       0.999952   4.762218E-005
 p2
       0.999959   4.080327E-005
```

Table 2: Convergence of Bayesian Learning Algorithm in Example 2

```
   enter 1 to enter new choice probabilities
   enter 2 to start from center of simplex
   enter 3 to start from existing values
   enter 4 to start from random initial probabilities
enter (1,2,3): 4

initial action probabilities
p1
       0.24278         0.75722
p2
      0.223778        0.776222
iteration   100
p1
      0.002404        0.997596
p2
      0.002216        0.997784
iteration   500
p1
      0.000485        0.999515
p2
      0.000447        0.999553
iteration  1000
p1
      0.000243        0.999757
p2
      0.000224        0.999776
iteration  2000
p1
      0.000121        0.999879
p2
      0.000112        0.999888
iteration  5000
p1
 4.854633E-005        0.999951
p2
 4.474658E-005        0.999955
iteration 10000
p1
 2.427559E-005        0.999976
p2
 2.237553E-005        0.999978
iteration 20000
p1
 1.213840E-005        0.999988
p2
 1.118832E-005        0.999989
iteration 30000
p1
 8.092403E-006        0.999992
p2
 7.459006E-006        0.999993
```

Table 3: Convergence of Bayesian Learning Algorithm in Example 3

```
    enter 1 to enter new choice probabilities
    enter 2 to start from center of simplex
    enter 3 to start from existing values
    enter 4 to start from random initial probabilities
 enter (1,2,3): 4

 initial action probabilities
 p1
      0.276849        0.723151
 p2
      0.04664         0.95336
 iteration   100
 p1
      0.557197        0.442803
 p2
      0.78264         0.21736
 iteration   500
 p1
      0.525503        0.474497
 p2
      0.732628        0.267372
 iteration  1000
 p1
      0.490786        0.509214
 p2
       0.75629         0.24371
 iteration 10000
 p1
      0.506077        0.493923
 p2
      0.753429        0.246571
 iteration 20000
 p1
      0.505339        0.494661
 p2
      0.751915        0.248085
 iteration 50000
 p1
      0.503995        0.496005
 p2
      0.749086        0.250914
 iteration 75000
 p1
       0.49981         0.50019
 p2
      0.752084        0.247916
```

TABLE 10: COMPARISON OF TRAJECTORIES OF MODIFIED BAYESIAN LEARNING ALGORITHM AND NEWTON PATH FOLLOWING ALGORITHM

```
Payoff matrix for player 1      Payoff matrix for player 2
    0  0  3                         0  0 -1
    0  1  1                        -1  2  3
   -1  2  2                         1  1  2

Nash equilibria
p1=(1,0,0) p2=(1,0,0)
p1=(1/2,0,1/2) p2=(1/2,1/3,1/6) p1=(1/2,0,1/2) p2=(0,1/3,2/3)  p1=(1/2,0,1/2) p2=(1/3,1/3,1/3)
initial action probabilities                         initial sig=5
p1
    0.33333333      0.33333333      0.33333333
p2
    0.33333333      0.33333333      0.33333333

Iteration   100     sig= 0.49751860                fixed point at sig=    0.49751860     det=2.70
p1                                                 p1
    0.38943894      0.25082508      0.35973597        0.41895816      0.12110599      0.45993585
p2                                                 p2
    0.15181518      0.38943894      0.45874587        0.16805039      0.34880881      0.48314080
lp1                                                lp1(p2)
    0.35994101      0.12321779      0.51684120        0.41895816      0.12110599      0.45993585
lp2                                                lp2(p1)
    0.07886023      0.36331059      0.55782918        0.16805039      0.34880881      0.48314080
Iteration   200     sig= 0.35267281                fixed point at sig=    0.35267281     det=4.52
p1                                                 p1
    0.39966833      0.16583748      0.43449420        0.44144161      0.07740141      0.48115699
p2                                                 p2
    0.12603648      0.34991708      0.52404643        0.17780012      0.34346067      0.47873921
lp1                                                lp1(p2)
    0.44347253      0.05979309      0.49673438        0.44144161      0.07740141      0.48115699
lp2                                                lp2(p1)
    0.08105542      0.33458060      0.58436398        0.17780012      0.34346067      0.47873921
Iteration   300     sig= 0.28819521                fixed point at sig=    0.28819521     det=6.38
p1                                                 p1
    0.39977852      0.12403101      0.47619048        0.45357138      0.05659888      0.48982974
p2                                                 p2
    0.12403101      0.34994463      0.52602436        0.18902747      0.34072124      0.47025129
lp1                                                lp1(p2)
    0.43628010      0.03874827      0.52497163        0.45357138      0.05659888      0.48982974
lp2                                                lp2(p1)
    0.08280040      0.30243041      0.61476919        0.18902747      0.34072124      0.47025129
Iteration   400     sig= 0.24968808                fixed point at sig=    0.24968808     det=8.24
p1                                                 p1
    0.40482128      0.10058188      0.49459684        0.46139243      0.04434884      0.49425873
p2                                                 p2
    0.11055694      0.33250208      0.55694098        0.19900445      0.33906036      0.46193518
lp1                                                lp1(p2)
    0.49667751      0.02134208      0.48198041        0.46139243      0.04434884      0.49425873
lp2                                                lp2(p1)
    0.08696348      0.29207184      0.62096468        0.19900445      0.33906036      0.46193518
Iteration   500     sig= 0.22338353                fixed point at sig=    0.22338353     det=10.08
p1                                                 p1
    0.43379907      0.08050566      0.48569528        0.46693277      0.03626304      0.49680418
p2                                                 p2
    0.10645376      0.31803061      0.57551564        0.20765832      0.33795072      0.45439096
lp1                                                lp1(p2)
    0.54852338      0.01295916      0.43851746        0.46693277      0.03626304      0.49680418
lp2                                                lp2(p1)
    0.10691312      0.31587771      0.57720918        0.20765832      0.33795072      0.45439096
```

TABLE 10, PAGE 2

```
Iteration 1000      sig= 0.15803489
p1
    0.45987346      0.04329004      0.49683650
p2
    0.11821512      0.33999334      0.54179154
lp1
    0.46957452      0.00420203      0.52622344
lp2
    0.14114322      0.32130097      0.53755581
Iteration 2000      sig=0.11177546
p1
    0.47892720      0.02215559      0.49891721
p2
    0.14659337      0.33499917      0.51840746
lp1
    0.49060600      0.00091323      0.50848077
lp2
    0.18365986      0.33296195      0.48337820
Iteration 5000      sig=0.07070361
p1
    0.49356795      0.00886489      0.49756715
p2
    0.17143238      0.32740119      0.50116643
lp1
    0.56188960 4.03212685E-005       0.43807008
lp2
    0.23813203      0.34690248      0.41496548
Iteration 10000     sig= 0.04999750
p1
    0.49668366      0.00443289      0.49888344
p2
    0.20441289      0.32970036      0.46588674
lp1
    0.55379188 3.27158848E-006       0.44620485
lp2
    0.26332407      0.34357272      0.39310321
Iteration 20000     sig= 0.03535446
p1
    0.50039165      0.00226655      0.49734180
p2
    0.24240455      0.33569988      0.42189557
lp1
    0.44961211 2.58437883E-007       0.55038763
lp2
    0.29418016      0.35656987      0.34924997
Iteration 30000     sig= 0.02886703
p1
    0.50466096      0.00151106      0.49382798
p2
    0.25803584      0.33469995      0.40726420
lp1
    0.46426471 2.80980081E-008       0.53573527
lp2
    0.33154274      0.38791980      0.28053746
```

```
fixed point at sig=     0.15803489   det=18.83
p1
    0.48098115      0.01813324      0.50088561
p2
    0.23777024      0.33546942      0.42676034
lp1(p2)
    0.48098115      0.01813324      0.50088561
lp2(p1)
    0.23777024      0.33546942      0.42676034
fixed point at sig=     0.11117755   det=34.15
p1
    0.49052958      0.00795170      0.50151872
p2
    0.26962593      0.33415439      0.39621968
lp1(p2)
    0.49052958      0.00795170      0.50151872
lp2(p1)
    0.26962593      0.33415439      0.39621968
fixed point at sig=     0.07070361   det=64.61
p1
    0.49721623      0.00205294      0.50073083
p2
    0.30567833      0.33349934      0.36082233
lp1(p2)
    0.49721623      0.00205294      0.50073083
lp2(p1)
    0.30567833      0.33349934      0.36082233
fixed point at sig=     0.04999750   det=93.45
p1
    0.49935254      0.00044747      0.50019999
p2
    0.32453004      0.33336159      0.34210837
lp1(p2)
    0.49935254      0.00044747      0.50019999
lp2(p1)
    0.32453004      0.33336159      0.34210837
fixed point at sig=     0.03535446   det=142.87
p1
    0.49994368 3.78527852E-005       0.50001847
p2
    0.33226614      0.33333510      0.33439876
lp1(p2)
    0.49994368 3.78527852E-005       0.50001847
lp2(p1)
    0.33226614      0.33333510      0.33439876
fixed point at sig=     0.02886703   det=203.40
p1
    0.49999286 4.77656039E-006       0.50000236
p2
    0.33316809      0.33333352      0.33349839
lp1(p2)
    0.49999286 4.77656039E-006       0.50000236
lp2(p1)
    0.33316809      0.33333352      0.33349839
```

TABLE 10, PAGE 3

```
Iteration 40000     sig= 0.02499969               fixed point at sig=    0.02499969   det=268.39
p1                                                p1
    0.49869587      0.00113331      0.50017083        0.49999879 8.07577276E-007      0.50000040
p2                                                p2
    0.26147680      0.32995008      0.40857312        0.33330106      0.33333336      0.33336558
lp1                                               lp1(p2)
    0.60060944 2.06241677E-009       0.39939056       0.49999879 8.07577276E-007      0.50000040
lp2                                               lp2(p1)
    0.29251458      0.33512923      0.37235618        0.33330106      0.33333336      0.33336558
Iteration 50000     sig= 0.02233813               fixed point at sig=    0.02233813   det=335.24
p1                                                p1
    0.49297087      0.00090484      0.50612430        0.49999975 1.65221804E-007      0.50000008
p2                                                p2
    0.26908312      0.33413172      0.39678516        0.33332594      0.33333334      0.33334072
lp1                                               lp1(p2)
    0.47299662 5.53504607E-010       0.52700338       0.49999975 1.65221804E-007      0.50000008
lp2                                               lp2(p1)
    0.23550620      0.26593691      0.49855689        0.33332594      0.33333334      0.33334072
Iteration 60000     sig= 0.02041224               fixed point at sig=    0.02041224   det=401.09
p1                                                p1
    0.50284717      0.00075554      0.49639728        0.49999994 4.04413770E-008      0.50000002
p2                                                p2
    0.27608429      0.33383333      0.39008239        0.33333135      0.33333333      0.33333531
lp1                                               lp1(p2)
    0.48143309 1.53515306E-010       0.51856691       0.49999994 4.04413770E-008      0.50000002
lp2                                               lp2(p1)
    0.33743312      0.37706281      0.28550407        0.33333135      0.33333333      0.33333531
```

TABLE 11: EXAMPLE OF NON-CONVERGENT TRAJECTORY OF MODIFIED BAYESIAN LEARNING ALGORITHM WHEN INITIAL VALUE OF SIG IS TOO SMALL

```
Payoff matrix for player 1     Payoff matrix for player 2

    0  0  3                        0  0 -1
    0  1  1                       -1  0  3
   -1  2  4                        2  0  0

Nash equilibria
   p1=(1,0,0) p2=(1,0,0)       p1=(6/9,2/9,1/9) p2=(5/8,2/8,1/8)

initial action probabilities                  initial sig=.5
p1
   0.33333333     0.33333333     0.33333333
p2
   0.33333333     0.33333333     0.33333333
Iteration   100       sig= 0.04975186
p1
   0.70627063     0.22112211     0.07260726
p2
   0.60726073     0.15181518     0.24092409
lp1
   0.74429198     0.00091623     0.25479178
lp2
   0.12409348     0.57941331     0.29649322
Iteration   500     sig= 0.02233835
p1
   0.59747172     0.22621424     0.17631404
p2
   0.55156354     0.20026613     0.24817033
lp1
   0.01158076 1.97718317E-008      0.98841922
lp2
   0.86109823     0.00354985     0.13535192
Iteration  1000     sig= 0.01580349
p1
   0.74159174     0.11322011     0.14518815
p2
   0.77256077     0.10023310     0.12720613
lp1
   0.99994262 5.73789152E-005 6.15579437E-013
lp2
   0.99998661 1.33878293E-005 1.25227284E-016
Iteration 5000      sig= 0.00707036
p1
   0.94267813     0.02826101     0.02906085
p2
   0.95187629     0.02266213     0.02546157
lp1
   0.98197684     0.01802316 7.46249703E-055
lp2
   0.98557421     0.01442579 2.91725605E-055
Iteration  6000     sig= 0.00645443
p1
   0.94606454     0.02971727     0.02421819
p2
   0.95406321     0.02471810     0.02121869
lp1
   0.93967252     0.06032748 3.41254896E-060
lp2
   0.94787984     0.05212016 1.14866372E-059

Iteration 7500     sig= 0.00577312
p1
   0.85406390     0.12656090     0.01937519
p2
   0.85952984     0.12349465     0.01697551
lp1
1.87513324E-007      0.99999981 2.78235035E-052
lp2
2.53177796E-007      0.99999975 1.90048649E-036
Iteration  9000      sig= 0.00527017
p1
   0.71173573     0.27211791     0.01614635
p2
   0.71629078     0.23467763     0.04903159
lp1
5.27915147E-012      1.00000000 2.52983364E-028
lp2
4.45890463E-029 2.54297744E-009      1.00000000
Iteration 10000      sig= 0.00499975
p1
   0.67846549     0.30700263     0.01453188
p2
   0.64466887     0.21121221     0.14411892
lp1
   0.99999963 2.11839833E-007 1.60102860E-007
lp2
5.97175831E-046 8.34040775E-022      1.00000000
Iteration 20000      sig= 0.00353545
p1
   0.61788577     0.15350899     0.22860524
p2
   0.67273303     0.10561139     0.22165558
lp1
   1.00000000 3.27228288E-042 3.47453464E-030
lp2
   1.00000000 4.91376345E-038 2.32605059E-057
Iteration 30000      sig= 0.00288670
p1
   0.74525294     0.10234103     0.15240603
p2
   0.78181838     0.07040876     0.14777285
lp1
   1.00000000 1.34214615E-034 6.28203380E-075
lp2
   1.00000000 3.45081068E-031 4.07925165E-097
```

TABLE 12: EXAMPLE OF CONVERGENCE OF MODIFIED BAYESIAN LEARNING ALGORITHM BY SETTING INITIAL VALUE OF SIG SUFFICIENTLY HIGH

```
  Payoff matrix for player 1        Payoff matrix for player 2

      0  0  3                            0  0 -1
      0  1  1                           -1  0  3
     -1  2  4                            2  0  0

  Nash equilibria
     p1=(1,0,0) p2=(1,0,0)    p1=(6/9,2/9,1/9)  p2=(5/8,2/8,1/8)

  initial action probabilities                       initial sig=5
  p1
      0.33333333       0.33333333       0.33333333
  p2
      0.33333333       0.33333333       0.33333333

  Iteration  100        sig= 0.49751860
  p1
      0.38943894       0.16171617       0.44884488
  p2
      0.55775578       0.16171617       0.28052805
  lp1
      0.39381415       0.18225169       0.42393417
  lp2
      0.65900751       0.15398465       0.18700784
  Iteration 1000        sig= 0.15803489
  p1
      0.58474858       0.19613720       0.21911422
  p2
      0.71861472       0.12021312       0.16117216
  lp1
      0.70768331       0.19670176       0.09561493
  lp2
      0.69538053       0.15006110       0.15455838
  Iteration 5000        sig= 0.07070361
  p1
      0.64833700       0.20162634       0.15003666
  p2
      0.65413584       0.20902486       0.13683930
  lp1
      0.60835774       0.24306239       0.14857987
  lp2
      0.72300092       0.17960182       0.09739726
  Iteration 10000       sig= 0.04999750
  p1
      0.66546679       0.20101323       0.13351998
  p2
      0.65726761       0.20501283       0.13771956
  lp1
      0.73718825       0.18020700       0.08260475
  lp2
      0.74423052       0.19866463       0.05710485
  Iteration 20000       sig= 0.03535446
  p1
      0.65308401       0.21075613       0.13615986
  p2
      0.64993417       0.21910571       0.13096012
  lp1
      0.71436246       0.21279345       0.07284408
  lp2
      0.78573996       0.13771882       0.07654121
```

```
  Iteration 50000       sig= 0.02236046
  p1
      0.65819350       0.21318240       0.12862409
  p2
      0.63065405       0.23696193       0.13238402
  lp1
      0.61528355       0.17741983       0.20729662
  lp2
      0.83337815       0.11613422       0.05048763
  Iteration 60000       sig= 0.02041224
  p1
      0.66697777       0.21850191       0.11452031
  p2
      0.65204469       0.22593512       0.12202019
  lp1
      0.69752328       0.28730405       0.01517267
  lp2
      0.51671834       0.30808605       0.17519561

  Mixed point at sig=     0.02041224   det=138,090
  p1
      0.66400191       0.21724662       0.11875148
  p2
      0.63532478       0.23551494       0.12916028
  lp1(p2)
      0.66400191       0.21724662       0.11875148
  lp2(p1)
      0.63532478       0.23551494       0.12916028

  Mixed strategy of complete information game
  p1
      0.66666667       0.22222222       0.11111111
  p2
      0.62500000       0.25000000       0.12500000
```