Convergence Analysis of Certain Reinforcement Learning Algorithms in Self-Play

Srinivas Turaga
Yonatan Loewenstein

I study the emergent behavior in the self-play of simple value function based reinforcement learning algorithms in an iterative two player general-sum game. The players are assumed to have limited knowledge of the game. Therefore learning algorithms only use information about their own choice and recieved reward histories to compute a value function based on which actions are chosen stochastically. Two related algorithms are studied, with fixed points that correspond to matching and Nash equilibria respectively. Linear stability analysis is used to probe the dynamics about the fixed points. Linear analysis shows that the matching equilibrium is stable, while the algorithm with the Nash equilibrium has marginal stability. Numerical simulations suggest that the Nash equilibrium is unstable for this class of algorithms and that algorithms with equilibria lying near the matching equilibrium are progressively more stable.

Introduction

Many interactions between people can be phrased in terms of the principles of game theory. We may think of people as players choosing from a space of actions, each of which produces a different outcome or payoff. What makes the problem interesting is that the payoff that a player receives generally depends strongly on the actions of the other players in the game. Classical game theory provides a mechanism to study optimal behavior in games involving rational players. Nash equilibria, which are the mathematically optimal outcome in games, have been well studied and characterized. Recently, however, studies suggest that people and animals behave differently from that predicted by classical game theory. The new field of behavioral game theory has arisen to explain the behavior of humans in game and decision theoretic scenarios.

As agents capable of learning and generalization, it is natural to believe that even if people start playing games with suboptimal strategies, they will eventually begin to get better and with practice reach optimality. With this in mind, we propose to study the emergent behavior of simple learning agents that play each other repeatedly in simple one-stage general-sum games. From this exercise, we hope to learn general principles underlying the behavior of learning agents such as people and animals in game theoretic situations.

In this work, we study the behavior of two simple reinforcement learning agents. It has been suggested that learning based in the theory of reinforcement learning [Sutton & Barto 1998] is used by animals at both the neural and behavioral level [Schultz 2002; Barraclough, Conroy, Lee 2004]. In this framework, we make the simplifying assumptions that people are only given access to their choice and reward histories, but have no knowledge of the other players actions or outcomes. We further assume that playeres use stochastic strategies where actions are chosen from a probability distribution which is learned and fine-tuned over time. Given this, we may think of the learning problem in terms of estimating the correct probability distribution over actions when given one's own choice and action histories.

Learning Algorithms and Notation

We consider two related reinforcement learning algorithms for study. These algorithms both use value functions to estimate the rewards to be gained by choosing a given action. In our system, each player has a value function over the discrete action space of two choices. The value of each action is updated independently.

$$p_j^i = \Pr(a^i(t) = j) = \frac{q_j^i[t]}{\sum_{k=1,2} q_k^i[t]} \tag{1}$$

$$r_j^i(t) = \delta_{1,,a^{k\neq i}(t)}\, R_{j,1}^i + (1 - \delta_{1,,a^{k\neq i}(t)})\, R_{j,2}^i$$

Algorithm 1

$$\Delta \mathrm{q}_j^i(t) = -\,\delta_{j,,a^i(t)}\, (\phi\, q_j^i[t] - r_j^i(t)) \tag{2}$$

Algorithm 2

$$\Delta \mathrm{q}_j^i(t) = -\ (\phi\, q_j^i[t] - \delta_{j,,a^i(t)}\, r_j^i(t)) \tag{3}$$

Here, $a^i(t)$ is the action chosen by the ith player at time t with a probability of p^i_j, and r^i_j is the reward obtained on the same play as a consequence of the choices of the two players for that play according to the fixed payoff matrices R^i for each player. q^i_j is the value function estimated by player i for action j over time using reward and choice histories. $\delta_{j,,a^i(t)}$ is the usual Kronecker delta function, which in this case is 1 whenever the jth action is taken at time t, and is 0 otherwise.

The essential difference between the two algorithms lies in the decay term. Algorithm 1 only allows a decay of a value function every time the action corresponding to that function is taken. In Algorithm 2, on the other hand the value function steadily decays, regardless of action choice. The latter might be interpreted as a memory leak, while the former simply attempts to correct the estimated value of an action at every opportunity given.

Methods

Analysis of stochastic algorithms is complicated. By construction, there is no way to predict the exact trajectory of the algorithms from any given starting point. However, we can use statistical methods to study the properties of the distribution of trajectories in phase space. A first order approximation might be made by simply considering the mean of the distribution. Here, we study the behavior of our algorithms in the mean by considering the average affect that our updates might have on the state of the system. This is known in physics as the mean-field approximation. Applying this leads to the following sets of equations, where the δ's are replaced by the probability of a particular action being taken.

$$\langle r^i_j \rangle = p^{k\neq i}_1 \, R^i_{j,1} + (1 - p^{k\neq i}_1)\, R^i_{j,2} \tag{4}$$

Algorithm 1

$$\left\langle \Delta \mathrm{q}^i_j(t) \right\rangle = -p^i_j \;\; (\phi\, q^i_j[t] - \langle r^i_j \rangle) \tag{5}$$

Algorithm 2

$$\left\langle \Delta \mathrm{q}^i_j(t) \right\rangle = - \;\; (\phi\, q^i_j[t] - p^i_j \,\langle r^i_j \rangle) \tag{6}$$

A further simplification is to consider the continuous time version of the system. This allows us to use techniques from continuum dynamical systems theory to study the behavior of the two players. This version of the system results from considering the limit where the updates made are infinitesimally small and the algorithm is allowed to run for a very long time. The dynamical system now looks like:

Algorithm 1

$$\left\langle \dot{q}^i_j(t) \right\rangle = -p^i_j \;\; (\phi\, q^i_j[t] - \langle r^i_j \rangle) \tag{7}$$

Algorithm 2

$$\left\langle \dot{q}^i_j(t) \right\rangle = - \;\; (\phi\, q^i_j[t] - p^i_j \,\langle r^i_j \rangle) \tag{8}$$

This is a 4-dimensional nonlinear system, since there are two players, each with two actions. A self-normalizing scaling given by equation (1) translates the value function to the more interesting action choice probabilities. Interestingly, we are really only concerned with a 2-dimensional projection of value function equations given by p^i_1, since p^i_2 is constrained to be (1-p^i_1).

Given this system of equations, we may compute its fixed points and their stability in order to achieve qualitative understanding of the dynamics. The stability of the fixed points may be probed in several ways including linear analysis and Lyapunov analysis [Strogatz 1994]. Here, we apply linear stability analysis to understand the nature of the fixed points of these algorithms. Further, we perform numerical simulations of the original stochastic algorithms and the approximate dynamics derived above to provide experimental insight into the behavior of the two algorithms.

Results

The system of equations resulting from Algorithm 1 is richer than that resulting from Algorithm 2. This is due to the fact that equation (8) is 2nd order in q_j^i, while equation (7) is 3rd order in q_j^i. We find that for general sum games with arbitrary payoff matrices, algorithm 1 has 10 fixed points, while algorithm 2 has 5 fixed points, demonstrating the difference in complexity. Most of these equilibria correspond to trivial solutions where one or more of the value functions is zero. In most noncooperative games, these are undesirable equilibria with sub-optimal payoff. The non-trivial fixed points of the dynamics for algorithm 1 correspond to the 2 symmetric matching equilibrium, where

$$\frac{p_1^i}{p_2^i} = \frac{\langle r_1^i \rangle}{\langle r_2^i \rangle} \tag{9}$$

In the case of symmetric payoff matrices for the two players, these two fixed points will collapse to one with equal values for both value functions (as in the case of matching pennies and other constant sum games). The single non-trivial fixed point of algorithm 2 corresponds to the Nash equilibrium of

$$p_1^1 = \frac{R_{2,2}^2 - R_{1,2}^2}{(R_{1,1}^2 + R_{2,2}^2) - (R_{1,2}^2 + R_{2,1}^2)} \tag{10}$$

for player 1 and vice versa for player 2.

We will first focus on the stability analysis about the more interesting non-trivial fixed points and then discuss the stability about the trivial fixed points. Unfortunately, linear stability analysis about the Nash equilibrium with algorithm 2 shows marginal stability. This is a border line case where the real part of the eigenvalues of the Jacobian matrix, evaluated at the fixed point, are all zero. In this case, linear analysis is inconclusive and nonlinear methods are needed. Numerical simulations for a constant sum game (figure (2)) of equations (8) (red curve) and (3) (blue curve) show that system tends to diverge from fixed point even for trajectories starting close to the fixed points. This suggests that the nonlinear terms in algorithm 2 lead to instability of the Nash fixed point.

Algorithm 2 proved to be significantly more difficult to analyze analytically for the general case of arbitrary payoff matrices. We simplified the problem by choosing a set of payoff matrices corresponding to a constant sum game and computing the stability of the matching equilibrium for that game. In this situation, we found that the single resulting matching equilibrium was stable. Additionally, we found that all the other equilibria were unstable in atleast one direction. This suggests that the matching equilibrium might be globally stable, although we haven't rigorously proved it as such. Numerical simulations (figure (1)) of equations (7) (red curve) and (2) (blue curve) also demonstrate robust convergence to the matching equilibrium, even for trajectories starting far away from this fixed point.

In the phase space trajectories shown in both sets of numerical simulations below, the trajectory starts location indicated by the green star and ends at the location shown by the blue star. Since the simulations were for constant sum games, the single matching equilibrium is the same as the Nash equilibrium and lies at p=0.5 for both players. We find that the deterministic mean-field and the stochastic algorithm generally converge to the same point at the matching equilibrium for algorithm 1, while the convergence of algorithm 2 is different for the two cases and diverges from the Nash equilibrium even after reaching it.

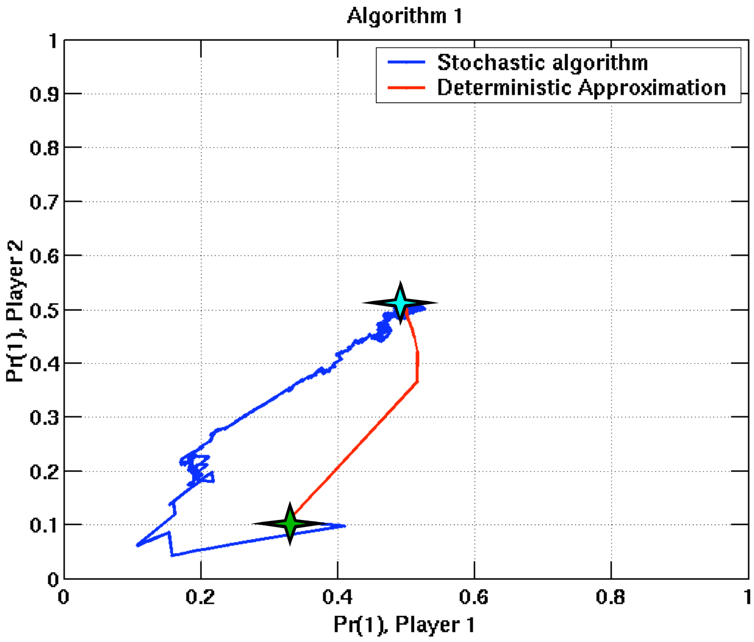

Figure 1

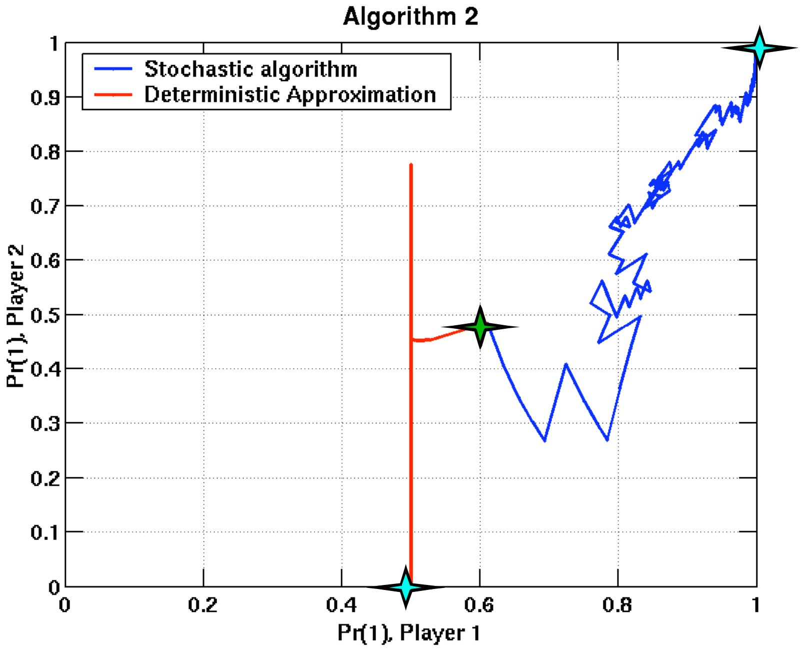

Figure 2

Discussion

Using mean-field and continuous time approximations, we have shown that the matching equilibrium of algorithm 1 in self-play is locally stable and have suggested its global stability for constant sum games. Similarly, we have found suggest the instability of the Nash equilibrium of algorithm 2. However, we must bear the following in mind while generalizing the results above to the original stochastic algorithm. The mean-field approximation may not be appropriate if the bulk of the actual distribution of trajectories is away from the mean. In other words, if the mean is different from the mode of the distribution, than the analysis is erroneous. Additionally, the continuous time approximation may also not necessarily predict the convergence behavior of the actual system. However, results such as those by Singh *et al* [Singh, Kearns, Mansour 2000] suggest that the continuous time approximation may hold for systems such as this. Additionally, our numerical simulations corroborate our analytic findings, suggesting the validity of the overall approach.

A potentially controversial assumption made in our study that player use stochastic strategies. It is still an open question as to whether humans and animals are capable of truly random behavior. Any amount of determinism employed by the player has the potential of nullifying all the conclusions in this work. The framework used by us is inappropriate for the analysis of deterministic strategies which may be more complex involving many internal player states.

But caveats aside, an interesting result is that for this class of algorithms, the Nash equilibrium is unstable, while the matching equilibrium is stable. This result is congruous with many earlier studies that have pitted people and animals against computers and have found probability matching or tit-for-tat behavior.

In the algorithms analyzed above, we have only considered the two extreme cases where the value function decays only when the corresponding action is taken and where the value function decays on every play. We may, instead, consider a continuum of algorithms in which the value function has a separate decay rate (ϕ_2) ranging from 0 to ϕ_1 .

$$\Delta \mathrm{q}_j^i(t) = -\,\delta_{j,,a^i(t)}\,(\phi_1\, q_j^i[t] - r_j^i(t)) - (1 - \delta_{j,,a^i(t)})\,\phi_2\, q_j^i[t] \tag{11}$$

This more general equation has stable matching behavior for $\phi_2 = 0$ and unstable Nash behavior for $\phi_2 = \phi_1$. Preliminary simulations with this algorithm suggest that for intermediate settings, the fixed point is progressively more stable with a location between the Nash and the matching equilibria.

To conclude, we present the analysis of a general reinforcement learning based model for learning agent in general-sum games. We find evidence in support of previous experiments where matching behavior was seen.

References

AG Barto, RS Sutton (1998) *Reinforcement Learning: An Introduction*, MIT Press.

DJ Barraclough, ML Conroy, D Lee (2004) "Prefrontal cortex and decision making in a mixed strategy game," *Nat. Neurosci.* **7**:404-401

W Schultz (2002) "Getting formal with dopamine and reward," *Neuron*, **36**:241-263

S Singh, M Kearns, Y Mansour (2000) "Nash Convergence of Gradient Dynamics in General-Sum Games," *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, pages 541-548.

SH Strogatz (1994) *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Perseus Book Publishing.