Friend-or-Foe Q-learning in General-Sum Games

Article · January 2003

Source: CiteSeer

CITATIONS
463

READS
10,122

1 author:

Michael L. Littman
Brown University
301 PUBLICATIONS **37,807** CITATIONS

SEE PROFILE

广义和游戏中的敌友 Q 学习

文章·2003 年 1 月
Source: CiteSeer

1 author:

Michael L. Littman
Brown University
301 PUBLICATIONS **37,807** CITATIONS

Friend-or-Foe Q-learning in General-Sum Games

Michael L. Littman MLITTMAN@RESEARCH.ATT.COM
AT&T Labs Research, 180 Park Avenue, Florham Park, NJ 07932-0971

Abstract

This paper describes an approach to reinforcement learning in multiagent general-sum games in which a learner is told to treat each other agent as either a "friend" or "foe". This Q-learning-style algorithm provides strong convergence guarantees compared to an existing Nash-equilibrium-based learning rule.

1. Introduction

Reinforcement learning (Sutton & Barto, 1998; Kaelbling et al., 1996) is a learning paradigm for agents in unfamiliar or complex environments. One particular class of scenarios in which reinforcement learning holds a great deal of promise is adapting in the presence of other agents. The Markov game framework (van der Wal, 1981; Shapley, 1953; Owen, 1982) is a formalization of multiagent interaction that is a good match for existing reinforcement-learning theory.

This paper extends the state of the art of reinforcement learning in general-sum Markov games by presenting the friend-or-foe Q-learning (FFQ) algorithm. FFQ provides strong convergence guarantees and learns a policy that is optimal in several important scenarios.

The ICML conference has been home to a series of papers developing the theory of reinforcement learning in games. Littman (1994) introduced a Q-learning algorithm called minimax-Q for zero-sum two-player games. Littman and Szepesvári (1996) showed that minimax-Q convergences to the game-theoretic optimal value. Hu and Wellman (1998) described an extension to minimax-Q, called Nash-Q here, that attacks general-sum games by using a Nash equilibrium computation in the learning rule. Bowling (2000) clarified the convergence conditions of the algorithm. Hu and Wellman (2000) studied the convergence behavior of Nash-Q in several small but non-trivial environments.

Briefly, although the Nash-Q algorithm is highly general, the assumptions that are known to be sufficient to guarantee its convergence are quite restrictive. These assumptions put limits on the types of games that can be guaranteed to be solved (those with coordination or adversarial equilibria) as well as on the intermediate results of learning. Aside from zero-sum or fully cooperative games, for which convergent learning algorithms are already known, no general-sum game has been shown to satisfy the intermediate-result restrictions of the Nash-Q theorem.

This paper presents a new algorithm, friend-or-foe Q-learning (FFQ), that always converges. In addition, in games with coordination or adversarial equilibria, FFQ converges to precisely the values Nash-Q ought to. To do this, FFQ requires that other players are identified as being either "friend" or "foe" and fully cooperative or zero-sum learning is used as appropriate. Although the theoretical properties of FFQ are an improvement over those of Nash-Q, a complete treatment of general-sum games is still lacking.

1.1 Definitions

A *one-stage general-sum n-player game* is defined by a set of n players, their action-choice sets $A_1, \ldots, A_n$, and their payoff functions $R_1, \ldots, R_n$. Each payoff function R_i maps an action choice for each of the players to a scalar reward value.

A *one-stage policy* for player i, π_i, is a probability distribution over its actions A_i. The *expected payoff* to player i when players adopt one-stage policies $\pi_1, \ldots, \pi_n$ is abbreviated $R_i(\pi_1, \ldots, \pi_n)$, which is just the expected value of the values of R_i weighted by the probabilities under the given one-stage policies.

Every one-stage general-sum n-player game has a *Nash equilibrium*. This is a set of one-stage policies $\pi_1, \ldots, \pi_n$ such that no player can improve its expected payoff by unilaterally changing its one-stage policy:

$$R_i(\pi_1, \ldots, \pi_n) \geq R_i(\pi_1, \ldots, \pi_{i-1}, \pi'_i, \pi_{i+1}, \ldots, \pi_n), \quad (1)$$

for all one-stage policies π'_i and $1 \leq i \leq n$. A game can have more than one Nash equilibrium, and the expected payoff to player i can vary depending on the equilibrium considered.

$$\geq\geq^{\pi}$$

$$(1) \qquad \epsilon(\pi_{\epsilon,\ldots,\epsilon,1+i\pi,\epsilon,i_1\pi,\epsilon,1-i\pi,\epsilon,\ldots,1\pi)i}\beta \leq (\pi_{\epsilon,\ldots,1\pi)i}\beta$$

$$\pi\pi_{\pi\pi}\bar{\pi}$$

$$\pi\pi\pi\bar{\mathbb{Q}}_{\pi\ \pi}$$

$$\pi\bar{\pi}$$

$$\mathbf{\beta}$$

Of central importance in this paper are two special kinds of Nash equilibria. An *adversarial equilibrium* satisfies Equation 1 and also has the property that no player i is hurt by any change of the other players:

$$R_i(\pi_1, \ldots, \pi_n) \leq R_i(\pi'_1, \ldots, \pi'_{i-1}, \pi_i, \pi'_{i+1}, \ldots, \pi'_n), \tag{2}$$

for all combinations of one-stage policies $\pi'_1, \ldots \pi'_n$. Not every game has an adversarial equilibrium. However, in a *two-player zero-sum game* $R_1 = -R_2$ and all equilibria are adversarial equilibria (Equation 1 implies Equation 2 in this case).

In a *coordination equilibrium*, all players achieve their highest possible value:

$$R_i(\pi_1, \ldots, \pi_n) = \max_{a_1 \in A_1, \ldots, a_n \in A_n} R_i(a_1, \ldots, a_n) \tag{3}$$

Once again, such an equilibrium need not always exist. However, in *fully cooperative games* $R_1 = R_2 = \cdots = R_n$ and there is at least one coordination equilibrium.

As a concrete example of equilibria, consider the two-player one-stage game defined by the following pair of matrices, with row player 1 and column player 2:

$$\text{row:} R_1 = \begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix}, \text{column:} R_2 = \begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix}. \tag{4}$$

Consider the pair of one-stage policies $\pi_1 = (0, 1)$, $\pi_2 = (1, 0)$. This is a Nash equilibrium since row's payoff of 0 would be worse (-1) if row changed to action 1, and column's payoff of 0 would be worse (-1) if column changed to action 2. In addition, it is an adversarial equilibrium, since row's payoff would improve (to 1) if column changed actions, and column's payoff would improve (to 1) if row changed actions.

On the other hand, the pair of one-stage policies $\rho_1 = (1, 0)$, $\rho_2 = (0, 1)$ is a coordination equilibrium, since both players receive their maximum payoff of 2.

Equation 4 shows that a game can have coordination and adversarial equilibria with different values. In a game with both types of equilibria, either the coordination equilibria have a higher value than the adversarial equilibria or both players have constant-valued payoff functions: $R_1 = c_1$, $R_2 = c_2$.

1.2 Markov Games

A Markov game is a generalization of the one-stage game to multiple stages. A game consists of a finite set of states S. Each state $s \in S$ has its own payoff functions as in the one-stage game scenario. In addition, there is a transition function that takes a state and an action choice for each player and returns a probability distribution over next states. In this setting, a *policy* maps states to probability distributions over actions.

The *value* for a player in a game, given discount factor $0 \leq \gamma < 1$, is the discounted sum of payoffs. In particular, let $\pi_1, \ldots, \pi_n$ be a set of policies for the n players. The *Q function* for player i is defined to be

$$Q_i(s, a_1, \ldots, a_n) = R_i(s, a_1, \ldots, a_n) + \gamma \sum_{s' \in S} T(s, a_1, \ldots, a_n, s') Q_i(s', \pi_1, \ldots, \pi_n), \tag{5}$$

where $Q_i(s', \pi_1, \ldots, \pi_n)$ is a weighted sum of the values of $Q_i(s', a'_1, \ldots, a'_n)$ according to the πs. The value of Equation 5 represents the value to player i when the state is s and the players choose actions a_1 through a_n then continue using their policies.

Filar and Vrieze (1997) showed how the Q function links Markov games and one-stage games. Treating the Q functions at each state as payoff functions for individual one-stage games, the policies at the individual states are in equilibrium if and only if the overall multistage policies are in equilibrium. Therefore, an equilibrium for a Markov game can be found by finding a Q function with the appropriate properties.

2. Nash-Q

This section describes a method for learning a Q function for a player from experience interacting with other players in the game. Experience takes the form of a tuple $\langle s, a_1, \ldots, a_n, s', r_1, \ldots, r_n \rangle$, where the game starts in state s, players choose the given actions, and a transition occurs to state s' with the given payoffs received. The Nash-Q learning rule (Hu & Wellman, 1998) works by maintaining a set of approximate Q functions and updating them by

$$Q_i[s, a_1, \ldots, a_n] := (1 - \alpha_t) Q_i[s, a_1, \ldots, a_n] + \alpha_t \left(r_i + \gamma \mathsf{Nash}_i(s, Q_1, \ldots, Q_n) \right) \tag{6}$$

each time a new experience occurs. Here, $\mathsf{Nash}_i(s, Q_1, \ldots, Q_n) = Q_i(s, \pi_1, \ldots, \pi_n)$ where the πs are a Nash equilibrium for the one-stage game defined by the Q functions $Q_1, \ldots, Q_n$ at state s. The values α_t are a sequence of *learning rates*, and this paper assumes they satisfy the standard stochastic approximation conditions for convergence (Jaakkola et al., 1994) (square summable, but not summable).

The principle motivation behind this choice of learning rule is that in single-player games (Markov decision processes), the Nash function is a simple maximization. Thus, in this case, the update rule reduces to

ff

$$R_i(\pi_1,\ldots,\pi_n) \geq R_i(\pi_1',\ldots,\pi_{i-1}',\pi_i,\pi_{i+1}',\ldots,\pi_n'), \tag{2}$$

ππ = −

$$R_i(\pi_1,\ldots,\pi_n) = \max_{a_1\in A_1,\ldots,a_n\in A_n} R_i(a_1,\ldots,a_n) \tag{3}$$

== · · · =

$$\text{row}: R_1 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \text{column}: R_2 = \begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}. \tag{4}$$

π= ()π= () ff− ff− fffifi

ρ=()ρ= () fi

fi==

≥ γ <fiππ ff

$$Q_i(s,a_1,\ldots,a_n) = R_i(s,a_1,\ldots,a_n) + \gamma \sum_{s'\in S} T(s,a_1,\ldots,a_n,s')Q_i(s',\pi_1,\ldots,\pi_n), \tag{5}$$

πππ

fi

fi

$$Q_i[s,a_1,\ldots,a_n] := (1-\alpha_t)Q_i[s,a_1,\ldots,a_n] + \alpha_t\left(r_i + \gamma \mathrm{Nash}_i(s,Q_1,\ldots,Q_n)\right) \tag{6}$$

=πππ α

Q-learning, which is known to converge to optimal values (Watkins & Dayan, 1992). Similarly, restricted to zero-sum games, the Nash function is a minimax function. In this case, the update rule reduces to minimax-Q, which is also known to converge to optimal values (a Nash equilibrium; Littman & Szepesvári, 1996).

The major difference between the general Nash function and both maximization and minimax is that the latter two have unique values whereas Nash does not. Consider, for example, the example in Equation 4, which can have value 0 or 2 depending on the Nash equilibrium considered. As a result, the Nash-Q learning rule in Equation 6, where Nash returns the value of an arbitrary equilibrium, cannot converge since it may use a different value at each update.

Hu and Wellman (1998) recognized that adversarial and coordination equilibria are unique.

Proposition 1 *If a one-stage game has a coordination equilibrium, all of its coordination equilibria have the same value.*

Proof: This is fairly direct, as, in each equilibrium, players get their unique maximum value. ■

Proposition 2 *If a one-stage game has an adversarial equilibrium, all of its adversarial equilibria have the same value.*

Proof: Let $\pi_1, \ldots, \pi_n$ and $\rho_1, \ldots, \rho_n$ be two adversarial equilibria for a one-stage game with payoffs $R_1, \ldots, R_n$. Comparing the expected payoff for player i under the π equilibrium to that of the ρ equilibrium,

$$\begin{aligned} & R_i(\pi_1, \ldots, \pi_n) \\ & \quad \geq \quad R_i(\pi_1, \ldots, \pi_{i-1}, \pi'_i, \pi_{i+1}, \ldots \pi_n) \\ & \quad = \quad R_i(\rho'_1, \ldots, \rho'_{i-1}, \rho_i, \rho'_{i+1}, \ldots, \rho'_n) \\ & \quad \geq \quad R_i(\rho_1, \ldots, \rho_n). \end{aligned}$$

These three statements follow from Equation 1 of the π equilibrium, setting the arbitrary one-stage policies to particular values, and Equation 2 of the ρ equilibrium, respectively. Repeating this argument reversing the roles of π and ρ shows that the expected payoff to player i under both equilibria must be the same. ■

Because of the uniqueness of the value of adversarial and coordination equilibria, their existence could possibly play a role in ensuring convergence of a general-sum reinforcement-learning algorithm. Let us define the following two conditions:

Condition A: There exists an adversarial equilibrium for the entire game.

Condition B: There exists a coordination equilibrium for the entire game.

In fact, it is not known whether Nash-Q converges under these conditions. Hu and Wellman (1998) (clarified by Bowling, 2000) used two stronger conditions to prove convergence:

Condition A+: There exists an adversarial equilibrium for the entire game and for every game defined by the Q functions encountered during learning.

Condition B+: There exists a coordination equilibrium for the entire game and for every game defined by the Q functions encountered during learning.

As stated, even Conditions A+ and B+ are not sufficient to guarantee convergence. It is necessary for the same type of equilibrium to be returned by the Nash subroutine and used in the value updates throughout the learning process.

There are many ways to provide this strong consistency guarantee. One is to assume that the algorithm never encounters a one-stage game with multiple equilibria. This results in extremely restrictive conditions, but no change in the algorithm. It is possible that this is the interpretation intended by Hu and Wellman (1998) as formalized in the following theorem.

Theorem 3 *Under Conditions A+ or B+, Nash-Q converges to a Nash equilibrium as long as all equilibria encountered during learning are unique.*

Another approach to guaranteeing update consistency is to ensure that the learning algorithm knows precisely which type of equilibrium to use in value updates. This approach requires weaker assumptions than Theorem 3, but demands that the learning algorithm be informed in advance as to which equilibrium type to use in all value updates. This requirement is formalized in the following theorem.

Theorem 4 *Under Conditions A+ or B+, Nash-Q converges to a Nash equilibrium as long as the corresponding equilibrium type is always used in Equation 6.*

3. Friend-or-Foe Q-learning

Friend-or-Foe Q-learning (FFQ) is motivated by the idea that the conditions of Theorem 3 are too strict because of the requirements it places on the intermediate values of learning. The conditions of Theorem 4 are more realistic—as long as we let the algorithm know

$$
\begin{aligned}
& \mathrm{A}_i(x_1, \ldots, x_n) \\
\leq\ & \mathrm{A}_i(x_1, \ldots, x_{i-1}, x_i \vee x_{i+1}, x_{i+1}, \ldots, x_n) \\
=\ & \mathrm{A}_i(y_1, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_n) \\
\leq\ & \mathrm{A}_i(y_1, \ldots, y_n).
\end{aligned}
$$

$$
\begin{aligned}
& \mathrm{A}_i(x_1, \ldots, x_n) \\
\leq\ & \mathrm{A}_i(x_1, \ldots, x_{i-1}, x_i \vee x_{i+1}, x_{i+1}, \ldots, x_n) \\
=\ & \mathrm{A}_i(y_1, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_n) \\
\leq\ & \mathrm{A}_i(y_1, \ldots, y_n).
\end{aligned}
$$

what kind of opponent[1] to expect: "friend" (coordination equilibrium) or "foe" (adversarial equilibrium). However, given this additional information about its opponents, a more direct learning algorithm suffices, as described next.

For simplicity, this section describes the two-player version of FFQ from the perspective of player 1. In FFQ, the learner maintains a Q function only for itself. The update performed is Equation 6 with

$$\mathsf{Nash}_1(s, Q_1, Q_2) = \max_{a_1 \in A_1, a_2 \in A_2} Q_1[s, a_1, a_2] \qquad (7)$$

if the opponent is considered a friend and

$$\mathsf{Nash}_1(s, Q_1, Q_2) = \max_{\pi \in \Pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi(a_1) Q_1[s, a_1, a_2] \qquad (8)$$

if the opponent is considered a foe. Equation 7 is just ordinary Q-learning in the combined action space of the two players. Equation 8 is minimax-Q and can be implemented via a straightforward linear program.

For completeness, the following is FFQ's replacement for the Nash function in n-player games. Let X_1 through X_k be the actions available to the k friends of player i and Y_1 through Y_l be the actions available to its l foes. Then, the value for a state is calculated as

$$\mathsf{Nash}_i(s, Q_1, \ldots, Q_n) =$$

$$\max_{\pi \in \Pi(X_1 \times \cdots \times X_k)} \min_{y_1, \ldots, y_l \in Y_1 \times \cdots \times Y_l} \sum_{x_1, \ldots, x_k \in X_1 \times \cdots \times X_k}$$

$$\pi(x_1) \cdots \pi(x_k) Q_i[s, x_1, \ldots, x_k, y_1, \ldots, y_l].$$

The idea is simply that i's friends are assumed to work together to maximize i's value, while i's foes are working together to minimize i's value. Thus, n-player FFQ treats any game as a two-player zero-sum game with an extended action set.

Theorem 5 *Friend-or-foe Q-learning converges.*

The theorem follows from the convergence of minimax-Q. Of course, the standard assumptions on learning rates are required. In brief, Littman and Szepesvári (1996) showed that the following non-expansion condition is sufficient to guarantee convergence: for all Q_1, Q_1', and s,

$$|\mathsf{Nash}_1(s, Q_1, \ldots) - \mathsf{Nash}_1(s, Q_1', \ldots)| \leq$$

$$\max_{a_1, a_2} |Q_1(s, a_1, a_2) - Q_1'(s, a_1, a_2)|.$$

This condition holds when Nash is max or minimax; it does not hold for the general Nash function.

In general, the values learned by FFQ will not correspond to those of any Nash equilibrium policy. However, there are special cases in which it will. Let Friend-Q denote FFQ assuming all opponents are friends and Foe-Q denote FFQ assuming all its opponents are foes. We have the following theorem, parallel to Theorem 4.

Theorem 6 *Foe-Q learns values for a Nash equilibrium policy if the game has an adversarial equilibrium (Condition A) and Friend-Q learns values for a Nash equilibrium policy if the game has a coordination equilibrium (Condition B). This is true regardless of opponent behavior.*

Proof: Because of the connection between equilibria in Markov games and one-stage games mentioned in Section 1.2, it is sufficient to show that the value of a coordination equilibrium in a one-stage game is the maximum payoff (true by definition) and that value of an adversarial equilibrium in a one-stage game is the minimax value (shown next).

Let $R_1, \ldots, R_n$ be the payoffs in a one-stage game. Let $\pi_1, \ldots, \pi_n$ be one-stage policies that achieve the minimax value for player 1 (assuming all other players act as a team). Thus, $R_1(\pi_1, \ldots, \pi_n) \geq R_1(\pi_1', \pi_2, \ldots, \pi_n)$ and $R_1(\pi_1, \ldots, \pi_n) \leq R_1(\pi_1, \pi_2', \ldots, \pi_n')$ for arbitrary one-stage policy π_1' and set of policies $\pi_2', \ldots, \pi_n'$. Let $\rho_1, \ldots, \rho_n$ be a set of one-stage policies in adversarial equilibrium. Compare the expected payoff to player 1 under π and ρ:

$$\begin{aligned} R_1(\pi_1, \ldots, \pi_n) &\geq R_1(\pi_1', \pi_2, \ldots, \pi_n) \\ &= R_1(\rho_1, \rho_2', \ldots, \rho_n') \\ &\geq R_1(\rho_1, \ldots, \rho_n). \end{aligned}$$

The first inequality follows from the fact that π_1 is minimax, the equality from taking $\pi_1' = \rho_1$ and $\rho_2', \ldots, \rho_n' = \pi_2, \ldots, \pi_n$, and the second inequality from Equation 2.

Similarly, we have

$$\begin{aligned} R_1(\rho_1, \ldots, \rho_n) &\geq R_1(\rho_1', \rho_2, \ldots, \rho_n) \\ &= R_1(\pi_1, \pi_2', \ldots, \pi_n') \\ &\geq R_1(\pi_1, \ldots, \pi_n). \end{aligned}$$

The first inequality follows from Equation 1, the equality from taking $\rho_1' = \pi_1$ and $\pi_2', \ldots, \pi_n' = \rho_2, \ldots, \rho_n$, and the second inequality because π_1 is minimax.

Thus, the value of an adversarial equilibrium for a player can be found by a minimax calculation using only its own payoffs. ■

[1] For lack of a better word, this paper uses "opponent" to refer to the *other* player, independent of whether it acts in an oppositional manner.

$$\begin{aligned}\beta\,\beta &= \pi\pi = \pi\pi\\ &\le \beta(\pi,\dots,\pi)\cdot\\ &= \beta(\pi,\pi,\dots,\pi)\\ &\le \beta(\pi,\pi,\dots,\pi)\end{aligned}\qquad\begin{aligned}\beta &= \pi\pi = \pi\pi\\ &\le \beta(q,\dots,q)\cdot\\ &= \beta(q,q,\dots,q)\\ &\le \beta(q,\dots,q)\end{aligned}$$

$$\begin{aligned}\beta(\beta(\pi,\pi,\pi,\pi)) &= \beta(\pi,\pi,\pi,\pi)\\ &\le \beta(\pi,\pi,\pi,\pi)\end{aligned}\qquad\begin{aligned}\beta(\beta(q,q,q,q)) &= \beta(q,q,q,q)\\ &\le \beta(q,q,q,q)\end{aligned}$$

$$\begin{aligned}\beta(\beta(\beta(\pi,\pi,\pi,\pi))) &= \beta(\pi,\pi,\pi,\pi)\\ &\le \beta(\pi,\pi,\pi,\pi)\end{aligned}\qquad\begin{aligned}\beta(\beta(\beta(q,q,q,q))) &= \beta(q,q,q,q)\\ &\le \beta(q,q,q,q)\end{aligned}$$

$$\beta(\beta(\beta(\beta(\pi,\pi,\pi,\pi)))) = \beta(\pi,\pi,\pi,\pi) \le \beta(\pi,\pi,\pi,\pi)$$

$$\beta(\beta(\beta(\beta(q,q,q,q)))) = \beta(q,q,q,q) \le \beta(q,q,q,q)$$

$$\beta(\beta(\beta(\beta(\beta(\pi,\pi,\pi,\pi))))) = \beta(\pi,\pi,\pi,\pi) \le \beta(\pi,\pi,\pi,\pi)$$

$$\beta(\beta(\beta(\beta(\beta(q,q,q,q))))) = \beta(q,q,q,q) \le \beta(q,q,q,q)$$

$$\beta(\beta(\beta(\beta(\beta(\beta(\pi,\pi,\pi,\pi)))))) = \beta(\pi,\pi,\pi,\pi) \le \beta(\pi,\pi,\pi,\pi)$$

$$\beta(\beta(\beta(\beta(\beta(\beta(q,q,q,q)))))) = \beta(q,q,q,q) \le \beta(q,q,q,q)$$

$$\beta(\beta(\beta(\beta(\beta(\beta(\beta(\pi,\pi,\pi,\pi))))))) = \beta(\pi,\pi,\pi,\pi) \le \beta(\pi,\pi,\pi,\pi)$$

$$\beta(\beta(\beta(\beta(\beta(\beta(\beta(q,q,q,q)))))))$$

$$\begin{aligned}\mathsf{Adsl}_1(s,\mathcal{Q}_1,\mathcal{Q}_5) &= \sum_{\substack{\iota_1\in\iota_0\\ \pi\in\Pi(\iota_1)}} \sum_{\substack{\varepsilon_1\in\varepsilon_0\\ \Pi\Pi\Pi\\ \chi\beta\Pi}} \pi(\iota_0)\mathcal{Q}_1[s,\iota_0,\varepsilon_0]\\ &= \sum_{\substack{\varepsilon_1\in\varepsilon_0,\iota_1\in\iota_0\\ \chi\beta\Pi\Pi}} \mathcal{Q}_1[s,\iota_0,\varepsilon_0] \qquad (7)\end{aligned}$$

$$\mathsf{Adsl}_1(s,\mathcal{Q}_1,\mathcal{Q}_5) = (\mathcal{Q}_1,\mathsf{Adsl}_1(s,\mathcal{Q}_1,\dots)$$

$$\begin{aligned}\mathsf{Adsl}_1(s,\mathcal{Q}_1,\mathcal{Q}_5) &= \sum_{\substack{\iota_1\in\iota_0\\ \pi\in\Pi(\iota_1)}} \sum_{\substack{\varepsilon_1\in\varepsilon_0\\ \Pi\Pi\Pi\\ \chi\beta\Pi}} \pi(\iota_0)\mathcal{Q}_1[s,\iota_0,\varepsilon_0]\\ &= \sum_{\substack{\varepsilon_1\in\varepsilon_0,\iota_1\in\iota_0\\ \chi\beta\Pi\Pi}} \mathcal{Q}_1[s,\iota_0,\varepsilon_0] \qquad (8)\end{aligned}$$

$$\mathsf{Adsl}_1(s,\mathcal{Q}_1,\mathcal{Q}_5) = (\mathcal{Q}_1,\mathsf{Adsl}_1(s,\mathcal{Q}_1,\dots)$$

$$\mathsf{Adsl}_5(s,\mathcal{Q}_1,\dots,\mathcal{Q}_5) = \sum_{\substack{\iota_1,\dots,\iota_5\in\iota_0\times\cdots\times\iota_0\\ \Pi\Pi\Pi\\ \chi\beta\Pi}} \mathcal{Q}_5[s,\iota_1,\dots,\iota_5]\cdot\pi(\iota_1)\cdots\pi(\iota_5)\mathcal{Q}_5[s,\iota_1,\dots,\iota_5,\mathsf{l}_1,\dots,\mathsf{l}_5].$$

$$|\mathsf{Adsl}_1(s,\mathcal{Q}_1,\dots,) - \mathsf{Adsl}_1(s,\mathcal{Q}_1,\dots)| \ge \sum_{\substack{\varepsilon_1,\iota_1\\ \chi\beta\Pi}} |\mathcal{Q}_1(s,\iota_1,\varepsilon) - \mathcal{Q}'_1(s,\iota_1,\varepsilon)|\cdot$$

Regardless of the game, there is an interpretation to the values learned by Foe-Q.

Theorem 7 *Foe-Q learns a Q function whose corresponding policy will achieve at least the learned values, regardless of the opponent's selected policy.*

This is a straightforward consequence of the use of minimax in the update rule of Foe-Q.

4. Examples

This section describes the behavior of FFQ and Nash-Q in two simple 2-player games. Since FFQ's convergence is guaranteed, idealized results are given; empirical results are cited for Nash-Q. The games are described by Hu and Wellman (2000) and are depicted in Figure 1. In both games, the players can move simultaneously in the four compass directions. Hitting a wall is a no-op. The first player to reach its goal receives 100 points. In the event of a tie, both players are rewarded (non-zero sum). If two players collide (try to enter the same grid position), each receives -1 and no motion occurs.

In grid game 1, both players can receive their maximum score. However, since their paths must cross, they need to coordinate to ensure that they don't collide. The game has several non-trivial coordination equilibria corresponding to the direct, collision-free paths for the two players. Nash-Q consistently learns correct values in this game (Hu & Wellman, 2000).

A Friend-Q learner finds the same values as Nash-Q. Both Nash-Q and Friend-Q have the difficulty that the existence of equal-valued non-compatible coordination equilibria means that it is possible that following a greedy policy with respect to the learned values may not achieve the learned value. For example, it is possible that player A chooses **E** from the start state and player B chooses **W**. Although both of these actions are part of coordination equilibria and are assigned maximal value, their combination results in disaster—neither player reaches the goal.

In grid game 1, a Foe-Q learner will assign a 0 value to the start state, as the opponent can always prevent the learner from reaching its goal. Although it is possible that the learner will choose actions that will cause it to reach the goal in a timely fashion (for example, if it chooses **N-N-W-W** and the opponent chooses **E-E-N-N**), it is just as likely that the learner will simply remain "cowering" in its initial position while the other player completes the game. However, the worst that a Foe-Q learner will receive is 0, compared to the substantial negative value possible of Friend-Q if non-compatible actions are selected.

Grid game 2 is different in that it possesses neither a coordination nor an adversarial equilibrium. A player that chooses to pass through the barrier (heavy line in figure) gets through with probability 1/2 and risks being beaten to the goal half the time by an opponent that uses the center passage. On the other hand, since both players cannot use the center passage simultaneously, it is not possible for both players to receive their maximum score. Bowling (2000) found that simple Q learners consistently identify a Nash equilibrium (one player uses the center passage, the other attempts to pass through the barrier). Hu and Wellman (2000) showed that Nash-Q learners will often identify equilibria, but the probability depended on how the Nash function was implemented.

Friend-Q learns to use the center passage in grid game 2. Whether this strategy is reasonable or not depends on the behavior of the opponent. If both players are Friend-Q learners, say, both will choose the center passage to their mutual disadvantage.

A Foe-Q learner will avoid the possibility of conflict in the center passage by attempting to cross the barrier. Half of the time, this will be successful regardless of the opponent's choice of action.

In both of these examples, Friend-Q and Foe-Q behave reasonably given their respective assumptions on the behavior of their opponents. Nevertheless, neither is ideal, as discussed in the next section.

5. Discussion

Simply put, the goal of learning is to maximize expected payoff. In a game, this goal is complicated by the fact that opponents cannot be directly controlled and may behave according to unknown policies. Depending on assumptions about opponents' behavior, there are many different ways to approach the learning problem.

To help explore the space of possible learning strategies, consider again grid game 2, described in the previous section. The most critical decision a player must make in the game is in the first step, when it decides whether to move to the center (**C**) or through the barrier (**B**). The game can be approximated by the simple one-stage game defined by

$$R_1 = \begin{array}{c} \\ \mathbf{C} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{C} & \mathbf{B} \end{array} \\ \begin{bmatrix} -100 & +100 \\ +50 & +66 \end{bmatrix} \end{array}$$

and $R_2 = R_1^T$. This game is known in the game the-

$$
\begin{array}{c} \\ \mathbf{C} \\ \mathbf{B} \end{array}
\begin{bmatrix}
\mathbf{C} & \mathbf{B} \\
-100 & +100 \\
+20 & +80
\end{bmatrix}
= \mathbf{A}_{12}
$$

=

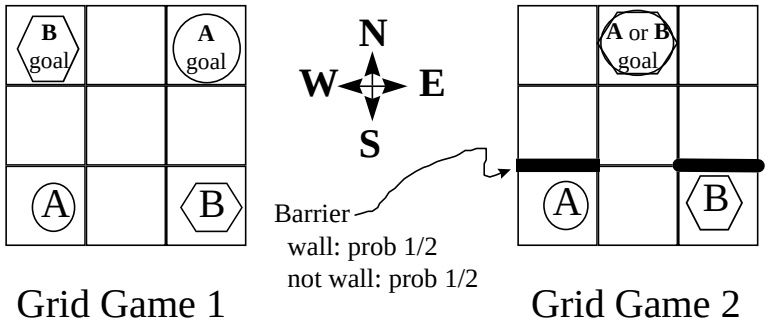


Figure 1. Two general-sum grid games.

ory literature as "chicken", because the player that is bold enough to use the center passage is the big winner, while the "chicken" that goes through the barrier scores less well. Of course, pairing off two center players is a disaster for all parties, as they collide entering the center passage.

Depending on the opponent faced, the learner can approach this game in several different ways:

worst case opponent: An adversary chooses the opponent used to evaluate the learner. Foe-Q finds the optimal policy for this scenario, which is **B**. It is extremely pessimistic, which can lead to overly conservative policies in some games.

best case opponent: The opponent will choose actions in a way that benefits the learner. This is appropriate only in purely cooperative settings, and is the perspective taken by Friend-Q, which selects **C** here.

unknown fixed opponent: The opponent faced during learning executes a fixed policy, which is used to evaluate the learning player. Q-learning finds the best response—the payoff maximizing policy with respect to the fixed opponent—in this scenario. Unfortunately, Q-learning need not learn an optimal policy if its opponents are not fixed—if they are also Q-learners, for example.

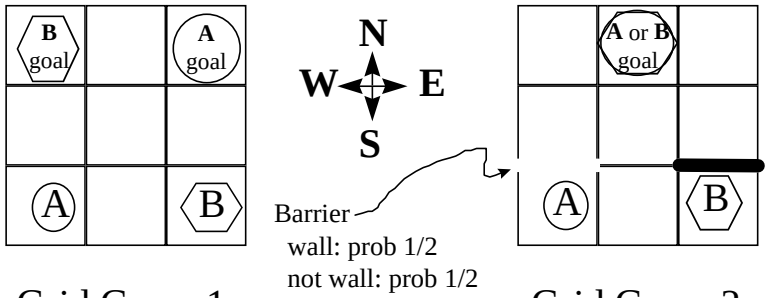
Nash opponent: In multiagent learning scenarios, it is reasonable to imagine that all players will continue adapting their behavior until they find a best response to the others. Under this assumption, the end result will be a Nash equilibrium. Therefore, finding a Nash equilibrium is a reasonable goal for a learner. However, it is not justified for a player to choose actions according to an *arbitrary* Nash equilibrium, since this does not maximize payoff. A more sensible approach, which has not been well explored, is choosing a Nash equilibrium policy that is somehow "close" to a best response to the observed behavior of the opponent.

best response opponent: If the learner assumes its opponents will adopt best response policies to its actions, it can choose a policy that maximizes its payoff under this assumption. Some early explorations of this approach in one-stage games indicate that all players can benefit in this scenario (Littman & Stone, 2001). By including simple history features, players can stabilize mutually beneficial strategies using the "threat" of aggressive punishment actions. For example, a player can propose to alternate between **C** and **B**, making it possible for both players to average +75. If the proposal is not accepted, the player can punish the opponent by executing **C**. A smart opponent will recognize that accepting this proposal is in its best interest. It is a multi-step Nash equilibrium, somewhat like tit-for-tat in the Prisoner's dilemma.

fixed-or-best-response opponent: The opponent types listed above can be roughly categorized into fixed strategies and best-response strategies. Another approach is to assume opponents belong to one of these two types and to attempt to identify which. Depending on the result, an appropriate learning strategy can be adopted. This seems like a very promising approach, and it has not received much attention.

Each of these assumptions on opponent type has benefits and drawbacks. If there is a single best assumption, it has yet to be recognized.

It is worth mentioning that grid game 2 admits an even more sophisticated strategy in which the **C** player stalls until the **B** player clears the barrier. This allows both players to receive a full payoff of +100 (minus the small expected penalty caused by the delay). It is possible that human beings would have a difficult time learning this policy, but it is worth studying the kinds of reasoning required for learning algorithms to converge to this kind of mutually beneficial behavior.



Grid Game 1

Grid Game 2

In a repeated game scenario, "sharing the wealth" like this can be in a player's self-interest if it prevents other players from becoming disgruntled and acting uncooperatively.

6. Conclusions

Friend-or-foe Q-learning (FFQ) provides an approach to reinforcement learning in general-sum games. Like Nash-Q, it should not be expected to find a Nash equilibrium unless either a coordination or an adversarial equilibrium exists.

Compared to Nash-Q, FFQ does not require learning estimates to the Q functions for opposing players, is easy to implement for multiplayer ($n \geq 2$) games, and provides guaranteed convergence. An extension of Theorem 4 shows that it can find equilibria in a large range of multiplayer games by mixing friends and foes. In contrast, Nash-Q does not necessarily require that the agent be told whether it is facing a "friend" or a "foe".

Foe-Q provides strong guarantees on the learned policy, specifically that the learner will act in a way that will achieve its learned value independent of its opponents' action choices. Furthermore, it chooses the policy that provides the largest such guarantee. Policies learned by Nash-Q need not have this property.

However, Friend-Q's guarantees are considerably weaker. Because of the possibility of incompatible coordination equilibria, the learner might not achieve its learned value, even if its opponent *is* a friend. Nash-Q also provides no answer to this problem.

In addition, neither Nash-Q nor FFQ address the problem of finding equilibria in cases where neither coordination nor adversarial equilibria exist. These are the most interesting games of all, since some degree of compromise is needed—the learner must be willing to accept an intermediate outcome between assuming its opponents will help it achieve its maximum value and assuming the opponents will force it to achieve its minimum value. This type of reasoning is much more subtle and remains to be addressed adequately in the field of reinforcement learning.

Acknowledgements

Michael Bowling, Junling Hu, Satinder Singh, Peter Stone, and Michael Wellman, and anonymous reviewers provided helpful comments during the development of this work.

References

Bowling, M. (2000). Convergence problems of general-sum multiagent reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 89–94).

Filar, J., & Vrieze, K. (1997). *Competitive Markov decision processes.* Springer-Verlag.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 242–250).

Hu, J., & Wellman, M. P. (2000). Experimental results on Q-learning for general-sum stochastic games. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 407–414).

Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, *6*, 1185–1201.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 157–163).

Littman, M. L., & Stone, P. (2001). Leading best-response strategies in repeated games. Research note.

Littman, M. L., & Szepesvári, C. (1996). A generalized reinforcement-learning model: Convergence and applications. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 310–318).

Owen, G. (1982). *Game theory.* Orlando, Florida: Academic Press. 2nd edition.

Shapley, L. (1953). Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, *39*, 1095–1100.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* The MIT Press.

van der Wal, J. (1981). *Stochastic dynamic programming.* No. 139 in Mathematical Centre tracts. Amsterdam: Mathematisch Centrum.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.

查看出版统计