In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98),* pages 242-250, Madison, WI, USA, July 1998

Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm

Junling Hu and Michael P. Wellman
Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109-2110, USA
{junling, wellman}@umich.edu
http://ai.eecs.umich.edu/people/{junling,wellman}

Abstract

In this paper, we adopt general-sum stochastic games as a framework for multiagent reinforcement learning. Our work extends previous work by Littman on zero-sum stochastic games to a broader framework. We design a multiagent Q-learning method under this framework, and prove that it converges to a Nash equilibrium under specified conditions. This algorithm is useful for finding the optimal strategy when there exists a unique Nash equilibrium in the game. When there exist multiple Nash equilibria in the game, this algorithm should be combined with other learning techniques to find optimal strategies.

1 Introduction

Reinforcement learning has gained attention and extensive study in recent years [5, 12]. As a learning method that does not need a model of its environment and can be used online, reinforcement learning is well-suited for multiagent systems, where agents know little about other agents, and the environment changes during learning. Applications of reinforcement learning in multiagent systems include soccer [1], pursuit games [14, 3] and coordination games [2]. In most of these systems, single-agent reinforcement learning methods are applied without much modification. Such approach treats other agents in the system as a part of the environment, ignoring the difference between responsive agents and passive environment. In this paper, we propose that a multiagent reinforcement learning method should explicitly take other agents into account. We also propose that a new framework is needed for multiagent reinforcement learning.

The framework we adopt is *stochastic games* (also called *Markov games*) [4, 15], which are the generalization of the Markov decision processes to the case of two or more controllers. Stochastic games are defined as non-cooperative games, where agents pursue their self-interests and choose their actions independently.

Littman [6] has introduced 2-player zero-sum stochastic games for multiagent reinforcement learning. In zero-sum games, one agent's gain is always the other agent's loss, thus agents have strictly opposite interests. In this paper, we adopt the framework of general-sum stochastic games, in which agents need no longer have opposite interests. General-sum games include zero-sum games as special cases. In general-sum games, the notions of "optimality" loses its meaning since each agent's payoff depends on other agents' choices. The solution concept *Nash equilibrium* [8] is adopted. In a Nash equilibrium, each agent's choice is the best response to the other agents' choices. Thus, no agent can gain by unilateral deviation.

we are interested in the Nash equilibrium solution because we want to design learning agent for noncooperative multiagent systems. In such systems, every agent pursues its own goal and there is no communication among agents. A Nash equilibrium is more plausible and self-enforcing than any other solution concept in such systems.

If the payoff structure and state transition probabilities are known to all the agents, we can solve for an Nash equilibrium strategy using a nonlinear programming method proposed by Filar and Vrieze [4]. In this paper, we are interested in situations where agents have incomplete information of other agents' payoff functions and the state transition probabilities. We show that an multiagent Q-learning algorithm can be designed, and it converges to the Nash equilibrium Q values under certain restrictions of the game. Our algorithm is designed for 2-player general-sum stochastic games, but can be extended to n-player general-sum games.

Our learning algorithm guarantees that an agent can learn a Nash equilibrium. But it does not say whether the other agent will learn the same Nash equilibrium. When there exist only one Nash equilibrium in the game, our learning algorithm works effectively. However, a game can have multiple Nash equilibria. In that case, our learning algorithm needs to be combined with empirical estimation of the action choices of the other agent.

2 Some preliminaries

We state some basic game theory concepts in this section. All concepts here refer to single-state (static) games. In later sections, we will see how the concepts here are connected to multi-state stochastic games.

For zero-sum games, the payoff matrices of two players can be described as $(M, -M)$, since one player's payoff is always the negative of the other. It is sufficient to simplify the game by either M or $-M$. Thus, 2-player zero-sum games are also called *matrix games.* For 2-player general-sum games, the agents' payoff matrices M^1 and M^2 are unrelated. The solutions of the game depend on both M^1 and M^2. Such games are called *bimatrix games.*

Definition 1 *A pair of matrices (M^1, M^2) constitutes a* bimatrix game, *where M^1 and M^2 are of the same size. The payoff $r^k(a^1, a^2)$ to player k can be found in the corresponding entry of the matrix M^k, $k = 1, 2$. The rows of M^k correspond to actions of player 1, $a^1 \in A^1$. The columns of M^k correspond to actions of player 2, $a^2 \in A^2$. A^1 and A^2 are the sets of discrete actions of players 1 and 2 respectively.*

Next, we state some solution concepts for bimatrix games. The main concept is Nash equilibrium [9]. In a Nash equilibrium, each agent's action is the best response to other agents' choices.

Definition 2 *A* pure strategy Nash equilibrium *for bimatrix game G is an action profile (a^1_*, a^2_*) such that*

$$\begin{aligned} r^1(a^1_*, a^2_*) &\geq r^1(a^1, a^2_*) \quad \textit{for all } a^1 \in A^1 \\ r^2(a^1_*, a^2_*) &\geq r^2(a^1_*, a^2) \quad \textit{for all } a^2 \in A^2 \end{aligned}$$

An example of a bimatrix game can be seen in Figure 1, in which the strategy pair (a^1_1, a^2_1) constitutes a pure strategy Nash equilibrium.

$$M^1 \qquad\qquad\qquad\qquad M^2$$

$$\begin{array}{cc} & \begin{array}{ccc} a^2_1 & a^2_2 & a^2_3 \end{array} \\ \begin{array}{c} a^1_1 \\ a^1_2 \end{array} & \begin{pmatrix} 1 & -2 & 4 \\ 0 & 1 & 1 \end{pmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{ccc} a^2_1 & a^2_2 & a^2_3 \end{array} \\ \begin{array}{c} a^1_1 \\ a^1_2 \end{array} & \begin{pmatrix} 2 & 1 & 0 \\ 0 & -3 & 2 \end{pmatrix} \end{array}$$

Figure 1: A bimatrix game example

Definition 3 *A* mixed strategy Nash equilibrium *for bimatrix game G is a pair of vectors (ρ^1_*, ρ^2_*), such that*

$$\begin{aligned} \rho^1_* M^1 \rho^2_* &\geq \rho^1 M^1 \rho^2_* \quad \textit{for all } \rho^1 \in \sigma(A^1) \\ \rho^1_* M^2 \rho^2_* &\geq \rho^1_* M^2 \rho^2 \quad \textit{for all } \rho^2 \in \sigma(A^2) \end{aligned}$$

where $\sigma(A^k)$ is the set of probability distributions over action space A^k, such that for any $\rho^k \in \sigma(A^k)$, $\sum_{a \in A^k} \rho^k(a) = 1$.[1]

$\rho^1 M^1 \rho^2 = \sum_{a^1} \sum_{a^2} \rho^1(a^1) r^1(a^1, a^2) \rho^2(a^2)$ is the expected payoff of agent 1 under the situation that player1 and player 2 adopt their mixed strategies ρ^1 and ρ^2 respectively.

The reason we are interested in mixed strategies is that an arbitrary bimatrix game may not have a pure strategy Nash equilibrium, but it always has a mixed strategy Nash equilibrium.

Theorem 1 *(Nash, 1951) There exists a mixed strategy Nash equilibrium for any finite bimatrix game.*

A mixed strategy Nash equilibrium for any bimatrix game can be found by Mangasarian-Stone algorithm [7], which is a quadratic programming algorithm.

3 Markov Decision Process and reinforcement learning

For comparison purpose, we state the framework of Markov decision process here. Later we can see how the stochastic game framework is related to Markov decision process.

Definition 4 *A Markov Decision Process is a tuple $< S, A, r, p >$, where S is the discrete state space, A is the discrete action space, $r : S \times A \to R$ is the reward function of the agent, and $p : S \times A \to \Delta$ is the transition function, where Δ is the set of probability distributions over state space S.*

[1] We abuse the notation a little here. $\rho^1_* M^1 \rho^2_*$ should be $(\rho^1_*)' M^1 \rho^2_*$, where ρ^1_* is transposed before being multiplied to the matrix M^1.

In a Markov decision process, the objective of the agent is to find a strategy (policy) π so as to maximize the expected sum of discounted rewards,

$$v(s,\pi)=\sum_{t=0}^{\infty}\beta^t E(r_t|\pi,s_0=s) \tag{1}$$

where s_0 is the initial state, r_t is the reward at time t, and $\beta\in[0,1)$ is the discount factor. We can rewrite Equation (1) as

$$v(s,\pi)=r(s,a_\pi)+\beta\sum_{s'}p(s'|s,a_\pi)v(s',\pi) \tag{2}$$

where a_π is action determined by policy π. It has been proved that there exists an optimal policy π^* such that for any $s\in S$, the following Bellman equation holds:

$$v(s,\pi^*)=\max_a\left\{r(s,a)+\beta\sum_{s'}p(s'|s,a)v(s',\pi^*)\right\}, \tag{3}$$

where $v(s,\pi^*)$ is called the optimal value for state s.

If the agent knows the reward function and the state transition function, it can solve for π^* by some iterative searching methods [10]. The learning problem arises when the agent does not know the reward function or the state transition probabilities. Now the agent needs to interact with the environment to find out its optimal policy. The agent can learn about the reward function and the state transition function, and then solve for its optimal policy using Equation (3). Such approach is called model-based reinforcement learning. The agent can also directly learn about its optimal policy without knowing the reward function or the state transition function. Such approach is called model-free reinforcement learning. One of the model-free reinforcement learning methods is Q-learning [16].

The basic idea of Q-learning is that we can define the right-hand side of Equation (3) as

$$Q^*(s,a)=r(s,a)+\beta\sum_{s'}p(s'|s,a)v(s',\pi^*) \tag{4}$$

By this definition, $Q^*(s,a)$ is the total discounted reward attained by taking action a in state s and then following the optimal policy thereafter. Then by Equation (3),

$$v(s,\pi^*)=\max_a Q^*(s,a). \tag{5}$$

If we know $Q^*(s,a)$, then the optimal policy π^* can be found, which is alway taking an action so as to maximize $Q^*(s,a)$ under any state s.

In Q-learning, the agent starts with arbitrary initial values of $Q(s,a)$ for all $s\in S, a\in A$. At each time t, the agent choose an action and observes its reward r_t. The agent then updates its Q-values based on the following Equation:

$$Q_{t+1}(s,a)=(1-\alpha_t)Q_t(s,a)+\alpha_t[r_t+\beta\max_b Q_t(s',b)]. \tag{6}$$

where $\alpha_t\in[0,1)$ is the learning rate. The learning rate α_t needs to decay over time in order for the learning algorithm to converge. Watkins and Dayan [16] proved that sequence (6) converges to the optimal $Q^*(s,a)$.

4 The stochastic game framework

Markov decision process (MDP) is a single agent decision problem. A natural extension of MDP to multiagent systems is stochastic games, which essentially are n-agent Markov decision processes. In this paper, we focus on 2-player stochastic games since they have been well studied.

4.1 Definition of stochastic games

Definition 5 *A 2-player stochastic game Γ is a 6-tuple $<S,A^1,A^2,r^1,r^2,p>$, where S is the discrete state space, A^k is the discrete action space of player k for $k=1,2$, $r^k: S\times A^1\times A^2\to R$ is the payoff function for player k, $p: S\times A^1\times A^2\to\Delta$ is the transition probability map, where Δ is the set of probability distributions over state space S.*

To have a closer look at a stochastic game, consider a process that is observable at discrete time points $t=0,1,2,\ldots$. At each time point t, the state of the process is denoted by s_t. Assume s_t takes on values from the set S. The process is controlled by 2 decision makers, referred to as player 1 and player 2, respectively. In state s, each player independently chooses actions $a^1\in A^1, a^2\in A^2$ and receives rewards $r^1(s,a^1,a^2)$ and $r^2(s,a^1,a^2)$, respectively. When $r^1(s,a^1,a^2)+r^2(s,a^1,a^2)=0$ for all s,a^1,a^2, the game is called *zero sum*. When the sum is not restricted to 0 or any constant, the game is called a *general-sum* game.

It is assumed that for every $s,s'\in S$, the transition from s to s' given that the players take actions $a^1\in A^1$ and $a^2\in A^2$, is independent of time. That is, there exist stationary transition probabilities $p(s'|s,a^1,a^2)$

for all $t = 0, 1, 2, \ldots$, satisfying the constraint

$$\sum_{s'=1}^{m} p(s'|s, a^1, a^2) = 1, \tag{7}$$

The objective of each player is to maximize a discounted sum of rewards. Let $\beta \in [0, 1)$ be the discount factor, let π^1 and π^2 be the strategies of players 1 and 2 respectively. For a given initial state s, the two players receive the following values from the game:

$$v^1(s, \pi^1, \pi^2) = \sum_{t=0}^{\infty} \beta^t E(r_t^1|\pi^1, \pi^2, s_0 = s) \tag{8}$$

$$v^2(s, \pi^1, \pi^2) = \sum_{t=0}^{\infty} \beta^t E(r_t^2|\pi^1, \pi^2, s_0 = s) \tag{9}$$

A strategy $\pi = (\pi_0, \ldots, \pi_t, \ldots)$ is defined over the whole course of the game. π_t is called the *decision rule* at time t. A strategy π is called a *stationary strategy* if $\pi_t = \bar{\pi}$ for all t, where the decision rule is fixed over time. π is called a *behavior strategy* if $\pi_t = f(h_t)$, where h_t is the history up to time t,

$$h_t = (s_0, a_0^1, a_0^2, s_1, a_1^1, a_1^2, \ldots, a_{t-1}^1, a_{t-1}^2, s_t). \tag{10}$$

A stationary strategy is a special case of behavior strategy when $h_t = \emptyset$.

A decision rule assigns mixed strategies to different states. A decision rule of a stationary strategy has the following form: $\bar{\pi} = (\bar{\pi}(s^1), \ldots, \bar{\pi}(s^m))$, where m is the maximal number of states. $\bar{\pi}(s)$ is a mixed strategy under state s.

A Nash equilibrium for stochastic games is defined as following, assuming that the players have complete information about the payoff functions of both players.

Definition 6 *In stochastic game Γ, a* Nash equilibrium point *is a pair of strategies (π_*^1, π_*^2) such that for all $s \in S$*

$$v^1(s, \pi_*^1, \pi_*^2) \geq v^1(s, \pi^1, \pi_*^2) \quad \forall \pi^1 \in \Pi^1$$

and

$$v^2(s, \pi_*^1, \pi_*^2) \geq v^2(s, \pi_*^1, \pi^2) \quad \forall \pi^2 \in \Pi^2$$

The definition of Nash equilibrium requires that each agent's strategy is a best response to the other's strategy. Such definition of Nash equilibrium is similar as in other games. The strategies that constitute a Nash equilibrium can be behavior strategies, Markov strategies, or stationary strategies. In this paper, we are interested in stationary strategies, which are the most simple strategies. The following theorem shows that there always exist a Nash equilibrium in stationary strategies for any stochastic game.

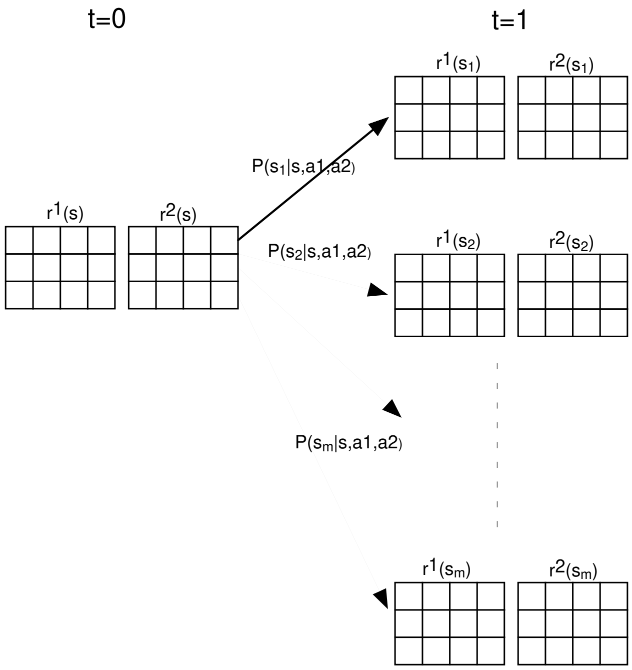

Figure 2: Stochastic games and bimatrix games

Theorem 2 *(Filar and Vrieze [4], Theorem 4.6.4) Every general-sum discounted stochastic game possesses at least one equilibrium point in stationary strategies.*

4.2 Stochastic games and bimatrix games

We can view each stage of a stochastic game as a bimatrix game, as in Figure 2.

At each time period of a stochastic game, under state s, agent 1 and 2 choose their actions independently and receive their payoffs according to the bimatrix game $(r^1(s), r^2(s))$. Repeated games can be seen as a degenerate case of stochastic games when there is only one state. For example, let $\bar{s}$ be the index of the only state, a repeated game will always have the bimatrix game $(r^1(\bar{s}), r^2(\bar{s}))$ at each time period.

5 Multiagent reinforcement learning

We want to extend traditional reinforcement learning method based on Markov decision process to stochastic games. We assume that our games have incomplete but perfect information, meaning agents do not know other agents' payoff functions but they can observe other agents' immediate payoffs and actions taken previously.

5.1 Issues in designing a multiagent Q-learning algorithm

The target of our Q-learning is the optimal Q-values, which we define as the following:

$$Q^1_*(s, a^1, a^2) = r^1(s, a^1, a^2) + \beta \sum_{s'=1}^{N} p(s'|s, a^1, a^2) v^1(s', \pi^1, \pi^2) \quad (11)$$

$$Q^2_*(s, a^1, a^2) = r^2(s, a^1, a^2) + \beta \sum_{s'=1}^{N} p(s'|s, a^1, a^2) v^2(s', \pi^1, \pi^2) \quad (12)$$

The optimal Q-value of state s and action pair (a^1, a^2) is the total discounted reward received by an agent when both agents execute actions (a^1, a^2) in state s and follow their Nash equilibrium strategies (π^1, π^2) thereafter.

To learn about these Q-values, an agent needs to maintain m Q-tables for its own Q-values, where m is the total number of states. For each agent k, $k = 1, 2$, a Q-table $Q^k(s)$ has its rows corresponding to $a^1 \in A^1$, columns corresponding to $a^2 \in A^2$, and each entry as $Q^k(s, a^1, a^2)$, $k = 1, 2$. The total number of entries agent k needs to learn is $m \times |A^1| \times |A^2|$, where $|A^1|$ and $|A^2|$ are the sizes of action spaces A^1 and A^2. Assuming $|A^1| = |A^2| = |A|$, then space requirement is $m \times |A|^2$. For n agents, the space requirement is $m \times |A|^n$, which is exponential in the number of agents. Thus for large number of agents, we need to find some compact representation of action space.

As in single-agent Q-learning, the learning agent in multiagent systems updates its Q tables for a given state after it observes the state, actions taken by both agents, and the rewards received by agents. The difference is in the updating rule. In single-agent Q-learning, the Q-values are updated as following,

$$Q_{t+1}(s, a) = (1 - \alpha_t) Q_t(s, a) + \alpha_t [r_t + \beta \max_b Q_t(s', b)].$$

In multiagent Q-learning, we cannot just maximize our own Q-values since the Q-values depend on the action of the other agent.

If it is a zero-sum game, we can minimize over the other agent's actions, and then choose our own maximal after that. This is the minimax-Q learning algorithm in Littman [6]. For general-sum games, we cannot use mini-max algorithm because the two agent's payoffs are not the opposite of each other. We propose that an agent adopt a Nash strategy to update its Q-values, and this is the best an agent can do in a general-sum game.

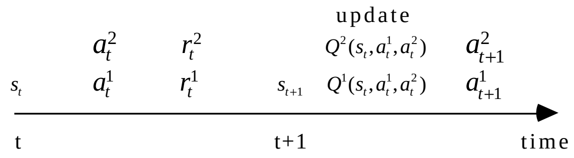

Figure 3: Time line of actions

5.2 A multiagent Q-learning algorithm

Our Q-learning agent, say agent 1, updates its Q-values according to the following rule:

$$Q^1_{t+1}(s, a^1, a^2) = (1 - \alpha_t) Q^1_t(s, a^1, a^2) + \alpha_t [r^1_t + \beta \pi^1(s') Q^1_t(s') \pi^2(s')] \quad (13)$$

where $(\pi^1(s'), \pi^2(s'))$ is a mixed strategy Nash equilibrium for the bimatrix game $(Q^1_t(s'), Q^2_t(s'))$. In order to find out $\pi^2(s')$, agent 1 needs to learn about $Q^2_t(s')$ in the game. The learning is as following:

$$Q^2_{t+1}(s, a^1, a^2) = (1 - \alpha_t) Q^2_t(s, a^1, a^2) + \alpha_t [r^2_t + \beta \pi^1(s') Q^2_t(s') \pi^2(s')] \quad (14)$$

Therefore, a learning agent maintains two Q-tables for each state, one for its own Q-values and one for the other agent's. This is possible since we assume an agent can observe the other agent's immediate rewards and previous actions during learning.

The detail of our Q-learning algorithm is stated in Table 1.

When the game is zero-sum, $Q^1(s, a^1, a^2) = -Q^2(s, a^1, a^2) = Q(s, a^1, a^2)$. Thus agent 1 needs to learn only one Q-table for every state. Our Q-learning algorithm becomes,

$$Q_{t+1}(s, a^1, a^2) = (1 - \alpha_t) Q_t(s, a^1, a^2) + \alpha_t [r_t + \beta \max_{\pi^1(s') \in \sigma(A^1)} \min_{\pi^2(s') \in \sigma(A^2)} \pi^1(s') Q_t(s') \pi^2(s')]$$

This is different from Littman's minimax-Q learning algorithm where Q-value is updated as

$$Q_{t+1}(s, a^1, a^2) = (1 - \alpha_t) Q_t(s, a^1, a^2) + \alpha_t [r_t + \beta \max_{\pi^1(s') \in \sigma(A^1)} \min_{a^2 \in A^2} \pi^1(s') Q_t(s', a^2)]$$

Table 1: Multiagent Q-learning algorithm for Agent 1

Initialize:
  Let $t = 0$,
  For all s in S, a^1 in A^1, and a^2 in A^2,
    let $Q_t^1(s, a^1, a^2) = 1, Q_t^2(s, a^1, a^2) = 1$
  initialize s_0
Loop
  Choose action a_t^1 based on $\pi^1(s_t)$, which is a mixed strategy Nash equilibrium solution of the bimatrix game $(Q^1(s_t), Q^2(s_t))$.
  Observe r_t^1, r_t^2, a_t^2, and s_{t+1}
  Update Q^1, and Q^2 such that
  $Q_{t+1}^1(s, a^1, a^2) = (1 - \alpha_t)Q_t^1(s, a^1, a^2) + \alpha_t[r_t^1 + \beta\pi^1(s_{t+1})Q_t^1(s_{t+1})\pi^2(s_{t+1})]$
  $Q_{t+1}^2(s, a^1, a^2) = (1 - \alpha_t)Q_t^2(s, a^1, a^2) + \alpha_t[r_t^1 + \beta\pi^1(s_{t+1})Q_t^2(s_{t+1})\pi^2(s_{t+1})]$
  where $(\pi^1(s_{t+1}), \pi^2(s_{t+1}))$ are mixed strategy Nash solutions of the bimatrix game $(Q^1(s_{t+1}), Q^2(s_{t+1}))$
  Let $t := t + 1$

In Littman's Q-learning algorithm, it is assumed that the other agent will always choose a pure Nash equilibrium strategy instead of a mixed strategy.

Another thing to note is that in our Q-learning algorithm, how an agent chooses its action at each time t is not important for the convergence of the learning. But the action choices are important for short-term performance. In this paper, we have not studied the issue of action choice, but will explore it in our future work.

5.3 Convergence of our algorithm

In this section, we prove the convergence of our Q-learning algorithm under certain assumptions. The first two assumptions are standard ones in Q-learning:

Assumption 1 *Every state and action have been visited infinitely often.*

Assumption 2 *the learning rate α_t satisfies the following conditions:*

1. $0 \le \alpha_t < 1, \sum_{t=0}^{\infty} \alpha_t = \infty, and \sum_{t=0}^{\infty} \alpha_t^2 < \infty,$
2. $\alpha_t(s, a^1, a^2) = 0$ *if* $(s, a^1, a^2) \neq (s_t, a_t^1, a_t^2)$.

We make further assumptions regarding the structure of the game:

Assumption 3 *A Nash equilibrium $(\pi^1(s), \pi^2(s))$ for any bimatrix game $(Q^1(s), Q^2(s))$ satisfies one of the following properties:*

1. *The Nash equilibrium is global optimal.*
   $\pi^1(s)Q^k(s)\pi^2(s) \ge \hat{\pi}^1(s)Q^k(s)\hat{\pi}^2(s) \quad \forall\hat{\pi}^1(s) \in \sigma(A^1), \hat{\pi}^2(s) \in \sigma(A^2)$, *and* $k = 1, 2$.
2. *If the Nash equilibrium is not a global optimal, then an agent receives a higher payoff when the other agent deviates from the Nash equilibrium strategy.*
   $\pi^1(s)Q^1(s)\pi^2(s) \le \pi^1(s)Q^1(s)\hat{\pi}^2(s) \quad \forall\hat{\pi}^2(s) \in \sigma(A^2)$, *and*
   $\pi^1(s)Q^2(s)\pi^2(s) \le \hat{\pi}^1(s)Q^2(s)\pi^2(s) \quad \forall\hat{\pi}^1(s) \in \sigma(A^1)$.

Our convergence proof is based on the following two Lemmas proved by Szepesvári and Littman [13].

Lemma 1 *(Conditional Average Lemma) Under Assumptions 1-2, the process $Q_{t+1} = (1 - \alpha_t)Q_t + \alpha_t w_t$ converges to $E(w_t|h_t, \alpha_t)$, where h_t is the history at time t.*

Lemma 2 *Under Assumptions 1-2, If the process defined by $U_{t+1}(x) = (1 - \alpha_t(x))U_t(x) + \alpha_t(x)[P_t v^*](x)$ converges to v^* and P_t satisfies $\| P_t V - P_t v^* \| \le \gamma \| V - v^* \| + \lambda_t$ for all V, where $0 < \gamma < 1$ and $\lambda_t \ge 0$ converges to 0, then the iteration defined by*

$$V_{t+1}(x) = (1 - \alpha_t(x))V_t(x) + \alpha_t(x)[P_t V_t](x)$$

converges to v^*.

In order to prove that the convergence point of our Q-learning algorithm is actually the Nash equilibrium point, we need the following theorem proved by Filar and Vrieze [4].

Theorem 3 *(Filar and Vrieze [4]) The following assertions are equivalent:*

1. *For each $s \in S$, the pair $(\pi^1(s), \pi^2(s))$ constitutes an equilibrium point in the static bimatrix game $(Q^1(s), Q^2(s))$ with equilibrium payoffs $(v^1(s, \pi^1, \pi^2), v^2(s, \pi^1, \pi^2))$, and for k=1,2 the entry (a^1, a^2) in $Q^k(s)$ equals*

$$Q^k(s, a^1, a^2) = r^k(s, a^1, a^2) + \beta \sum_{s'=1}^{N} p(s'|s, a^1, a^2)v^k(s', \pi^1, \pi^2).$$

2. (π^1, π^2) *is an equilibrium point in the discounted stochastic game* Γ *with equilibrium payoff* $\left(\boldsymbol{v}^1(\pi^1,\pi^2), \boldsymbol{v}^2(\pi^1,\pi^2)\right)$*, where* $\boldsymbol{v}^k(\pi^1,\pi^2) = (v^k(s^1,\pi^1,\pi^2), \cdots, v^k(s^m,\pi^1,\pi^2))$*,* $k = 1, 2$.

The above theorem showed that the Nash solution of the bimatrix game $(Q^1(s), Q^2(s))$ defined in Theorem 3 will also be part of the Nash solution for the whole game. If the sequence in our Q-learning algorithm converges to the Q-values defined in Theorem 3, then a pair of stationary Nash equilibrium strategies $(\bar{\pi}^1, \bar{\pi}^2)$ can be derived, where $\bar{\pi}^k = (\bar{\pi}^k(s^1), \cdots, \bar{\pi}^k(s^m))$ for $k = 1, 2$. For each state s, $\bar{\pi}^k(s)$ is part of a Nash equilibrium solution of the bimatrix game $(Q^1(s), Q^2(s))$.

Lemma 3 *Let* $P_t^k Q^k(s) = r_t^k + \beta \pi^1(s) Q^k(s) \pi^2(s)$*,* $k = 1, 2$*, where* $(\pi^1(s), \pi^2(s))$ *is a pair of mixed Nash equilibrium strategies for the bimatrix game* $(Q^1(s), Q^2(s))$*. Then* $P_t = (P_t^1, P_t^2)$ *is a contraction mapping.*

Proof. Case 1: $P_t^k Q^k(s) \geq P_t^k \hat{Q}^k(s) \quad \forall k = 1, 2.$

We have

$$
\begin{aligned}
0 &\leq P_t^k Q^1(s) - P_t^k \hat{Q}^1(s) \\
&= \beta \left(\pi^1(s) Q^1(s) \pi^2(s) - \hat{\pi}^1(s) \hat{Q}^1(s) \hat{\pi}^2(s)\right) \\
&\leq \beta \left(\pi^1(s) Q^1(s) \pi^2(s) - \pi^1(s) \hat{Q}^1(s) \hat{\pi}^2(s)\right) \quad (15) \\
&\leq \beta \left(\pi^1(s) Q^1(s) \hat{\pi}^2(s) - \pi^1(s) \hat{Q}^1(s) \hat{\pi}^2(s)\right) \quad (16) \\
&= \beta \sum_{a^1} \sum_{a^2} \pi^1(s,a^1) \hat{\pi}^2(s,a^2) (Q^1(s,a^1,a^2) - \\
&\quad \hat{Q}^1(s,a^1,a^2)) \quad (17) \\
&\leq \beta \sum_{a^1} \sum_{a^2} \pi^1(s,a^1) \hat{\pi}^2(s,a^2) \parallel Q^1(s) - \hat{Q}^1(s) \parallel \\
&= \beta \parallel Q^1(s) - \hat{Q}^1(s) \parallel,
\end{aligned}
$$

where $\parallel Q^k(s) - \hat{Q}^k(s) \parallel = \max_{a^1,a^2} |Q^k(s,a^1,a^2) - \hat{Q}^k(s,a^1,a^2)|$. Inequality (15) derives from definition of Nash equilibrium. Inequality (16) is from property 2 of Assumption 2. For cases satisfying property 1 of Assumption 2, the proof is simpler, and we omit it here.

$k = 2$, similar proof as above. Under property 1 of Assumption 2, we have

$$
\begin{aligned}
0 &\leq P_t^k Q^2(s) - P_t^k \hat{Q}^2(s) \\
&\leq \beta \sum_{a^1} \sum_{a^2} \pi^1(s,a^1) \pi^2(s,a^2) \parallel Q^2(s) - \hat{Q}^2(s) \parallel \\
&= \beta \parallel Q^2(s) - \hat{Q}^2(s) \parallel .
\end{aligned}
$$

Under property 2 of Assumption 2, we have

$$
\begin{aligned}
0 &\leq P_t^k Q^2(s) - P_t^k \hat{Q}^2(s) \\
&\leq \beta \sum_{a^1} \sum_{a^2} \hat{\pi}^1(s,a^1) \pi^2(s,a^2) \parallel Q^2(s) - \hat{Q}^2(s) \parallel \\
&= \beta \parallel Q^2(s) - \hat{Q}^2(s) \parallel .
\end{aligned}
$$

Case 2: $P_t^k Q^k(s) \leq P_t^k \hat{Q}^k(s)$. Similar proof as in Case 1. For $k = 1$, under property 2 of Assumption 2, we have

$$
\begin{aligned}
0 &\leq P_t^k \hat{Q}^1(s) - P_t^k Q^1(s) \\
&\leq \beta \sum_{a^1} \sum_{a^2} \hat{\pi}^1(s,a^1) \pi^2(s,a^2) \parallel \hat{Q}^1(s) - Q^1(s) \parallel \\
&= \beta \parallel \hat{Q}^1(s) - Q^1(s) \parallel .
\end{aligned}
$$

Therefore we have $|P_t^k Q^k(s) - P_t^k \hat{Q}^k(s)| \leq \beta \parallel Q^k(s) - \hat{Q}^k(s) \parallel$. Since this holds for every state s, we have $\parallel P_t^k Q^k - P_t^k \hat{Q}^k \parallel \leq \beta \parallel Q^k - \hat{Q}^k \parallel$. □

Now we proceed to prove our main theorem, which states that the multiagent Q-learning methods converges to the "optimal" (Nash equilibrium) Q values.

Theorem 4 *In stochastic game* Γ*, under Assumptions 1-3, the coupled sequences* $\{Q_t^1, Q_t^2\}$*, updated by*

$$
\begin{aligned}
&Q_{t+1}^k(s,a^1,a^2) = \\
&\quad (1-\alpha_t) Q_t^k(s,a^1,a^2) + \alpha_t [r_t^k + \beta \pi^1(s') Q_t^k(s') \pi^2(s')] \quad (18)
\end{aligned}
$$

where $k = 1, 2$*, converge to the Nash equilibrium Q values* (Q_*^1, Q_*^2)*, with* Q_*^k *defined as*

$$
\begin{aligned}
&Q_*^k(s,a^1,a^2) = \\
&\quad r^k(s,a^1,a^2) + \beta \textstyle\sum_{s'=1}^N p(s'|s,a^1,a^2) v^k(s', \pi_*^1, \pi_*^2), \quad (19)
\end{aligned}
$$

where $(\pi^1(s'), \pi^2(s'))$ *is a pair of mixed Nash equilibrium strategies for the bimatrix game* $(Q_t^1(s'), Q_t^2(s'))$*, function* v^k *is defined as in (8) and (9), and* (π_*^1, π_*^2) *is a Nash equilibrium solution for stochastic game* Γ.

Proof. By Lemma 3, $\parallel P_t^k Q^k - P_t^k Q_*^k \parallel \leq \beta \parallel Q^k - Q_*^k \parallel$.

From Lemma 1, the sequence

$$
\begin{aligned}
&Q_{t+1}^k(s,a^1,a^2) = \\
&\quad (1-\alpha_t) Q_t^k(s,a^1,a^2) + \alpha_t [r_t^k + \beta \pi^1(s') Q^k(s') \pi^2(s')]
\end{aligned}
$$

converges to

$$
\begin{aligned}
E(r_t^k + \beta \pi^1 Q^k(s') \pi^2) &= \sum_{s'} P(s'|s,a^1,a^2) \\
&\quad \left(r^k(s,a^1,a^2) + \beta \pi^1(s') Q^k(s') \pi^2(s')\right).
\end{aligned}
$$

Define T^k as

$$(T^kQ^k)(s,a^1,a^2) = \sum_{s'} P(s'|s,a^1,a^2)\Big(r^k(s,a^1,a^2) + \beta\pi^1(s')Q^k(s')\pi^2(s')\Big)$$

From above, the sequence $\{Q_t^k\}$ converges to T^kQ^k. It is easy to show that T^k is a contraction mapping. To see this is true, rewrite T^k as $T^kQ^k(s) = \sum_{s'} P(s'|s,a^1,a^2)P_tQ^k(s)$. Since P_t is a contraction mapping of Q^k and $P(s'|s,a^1,a^2) \geq 0$, T^k is also a contraction mapping of Q^k. We proceed to show that Q_*^k defined in (19) is the fixed point of T^k. From the definition of T^k, we have

$$\begin{aligned} & (T^kQ_*^k)(s,a^1,a^2) \\ = & \sum_{s'} P(s'|s,a^1,a^2)\Big(r^k(s,a^1,a^2) + \beta\pi_*^1(s')Q_*^k(s')\pi_*^2(s')\Big) \\ = & r^k(s,a^1,a^2) + \sum_{s'} P(s'|s,a^1,a^2)\beta\pi_*^1(s')Q_*^k(s')\pi_*^2(s') \end{aligned}$$

By Theorem 3, $\pi_*^1(s')Q_*^k(s')\pi_*^2(s') = v^k(s',\pi_*^1,\pi_*^2)$, thus $Q_*^k = T^kQ_*^k$. Therefore the sequence

$$Q_{t+1}^k(s,a^1,a^2) = (1-\alpha_t)Q_t^k(s,a^1,a^2) + \alpha_t[r_t^1 + \beta\pi_*^1Q_*^k(s')\pi_*^2] \quad (20)$$

converges to $T^kQ_*^k = Q_*^k$. By Lemma 2, the sequence (18) converges to Q_*^k. □

5.4 Discussions

First we want to point out the convergence result does not depend on the sequence of actions taken by either agent. The convergence result only requires that every action has been tried and every state has been visited. It does not require that agent 1 and agent 2 agree on the Nash equilibrium of each bimatrix Q-game during the learning. In fact, agent 1 can learn its optimal Q-value without any behavior assumption of agent 2, as long as agent 1 can observe agent 2's immediate rewards.

Second, the convergence depends on certain restrictions on the bimatrix games during learning. This is required because Nash equilibrium operator is usually not a contraction operator. However, we can probably relax the restriction by proving that a Nash equilibrium operator is a non-expansion operator. Then by the theorem in Szepesvári and Littman [13], the convergence is guaranteed.

6 Future work

There are several issues we have not addressed in this paper. The first is the equilibrium selection problem. When there exist multiple Nash equilibria, learning one Nash equilibrium strategy does not guarantee the other agent will choose the same Nash equilibrium. Our future work is to combine empirical estimation of the other agent's strategy with reinforcement learning of the Nash equilibrium strategy.

Another issue is related to the action choice during the learning. Even though the multiagent reinforcement learning method converges, it requires infinite trials. During the learning, an agent can choose a myopic action or other kinds of actions. If the agent chooses the action to maximize its current Q-value, its approach is called greedy approach. The drawback of this greedy approach is that the agent may be trapped in a local optimal. To avoid this problem, the agent should explore other possible actions. However, there is cost associated with exploration. By conducting exploration, an agent gives up a better current reward. In our future work, we intend to design an algorithm that can handle exploration and exploitation tradeoff in stochastic games.

References

[1] Tucker Balch. Learning roles: Behavioral diversity in robot teams. In Sen [11].

[2] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In Sen [11]. To appear in AAAI-98.

[3] Edwin De Jong. Non-random exploration bonuses for online reinforcement learning. In Sen [11].

[4] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Process*. Springer-Verlag, 1997.

[5] Leslie Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996.

[6] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163. New Brunswick, 1994.

[7] O. L. Mangasarian and H. Stone. Two-person nonzero-sum games and quadratic programming. *Journal of Mathematical Analysis and Applications*, 9:348–355, 1964.

[8] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[9] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[10] Martin L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming.* New York : John Wiley & Sons, 1994.

[11] Sandip Sen, editor. *Collected papers from the AAAI-97 workshop on multiagent learning.* AAAI Press, 1997.

[12] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press/Bradford Books, March 1998.

[13] Csaba Szepesvári and Michael L. Littman. A unified analysis of value-function-based reinforcement-learning algorithms. submitted for review, December 1997.

[14] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337, Amherst, MA, June 1993. Morgan Kaufmann.

[15] Frank Thusijsman. *Optimality and Equilibria in Stochastic Games.* Amsterdam, the Netherlands : Centrum voor Wiskunde en Informatica, 1992.

[16] Christopher J.C.H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 3:279–292, 1992.