Convergence Problems of General-Sum Multiagent Reinforcement Learning

Michael Bowling
Carnegie Mellon University
Computer Science Department

ICML 2000
Overview

- Stochastic Game Framework
- Q-Learning for General-Sum Games [Hu & Wellman, 1998]
- Counterexample and Flaw
- Discussion
Stochastic Game Framework

- MDPs
  - Single Agent
  - Multiple State
- Matrix Games
  - Multiple Agent
  - Single State
- Stochastic Games
  - Multiple Agent
  - Multiple State
A Markov decision process (MDP) is a tuple, \((S, A, T, R)\), where,

- \(S\) is the set of states,
- \(A\) is the set of actions,
- \(T\) is a transition function \(S \times A \times S \rightarrow [0, 1]\),
- \(R\) is a reward function \(S \times A \rightarrow \mathbb{R}\).
A *matrix game* is a tuple \((n, A_1, \ldots, A_n, R_1, \ldots, R_n)\), where,

- \(n\) is the number of players,
- \(A_i\) is the set of actions available to player \(i\) — \(A\) is the joint action space \(A_1 \times \cdots \times A_n\),
- \(R_i\) is player \(i\)’s payoff function \(A \to \mathbb{R}\).

\[
R_1 = \begin{pmatrix}
a_1 & \cdots & R_1(a) & \cdots \\
\vdots & & \vdots & \vdots \\
\end{pmatrix}
\]

\[
R_2 = \begin{pmatrix}
a_1 & \cdots & R_2(a) & \cdots \\
\vdots & & \vdots & \vdots \\
\end{pmatrix}
\]
Matrix Game – Examples

Matching Pennies

\[ R_{\text{row}} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad R_{\text{col}} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \]

This is a zero-sum matrix game.

Coordination Game

\[ R_{\text{row}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad R_{\text{col}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \]

This is a general-sum matrix game.
Matrix Games – Solving

- No optimal opponent independent strategies.

- Mixed (i.e. stochastic) strategies does not help.

- Opponent dependent strategies,

  **Definition 1** *For a game, define the best-response function for player* \( i \), \( BR_i(\sigma_{-i}) \), *to be the set of all, possibly mixed, strategies that are optimal given the other player(s) play the possibly mixed joint strategy* \( \sigma_{-i} \).
Matrix Games – Solving

- Best-response equilibrium [Nash, 1950],

**Definition 2** A Nash equilibrium is a collection of strategies (possibly mixed) for all players, $\sigma_i$, with,

$$\sigma_i \in BR_i(\sigma_{-i}).$$

- Example Games:
  - *Matching Pennies*: Both players playing each action with equal probability.
  - *Coordination Game*: Both players play action 1 or both players play action 2.
Stochastic Game Framework

- Stochastic Games
  - Multiple State
  - Multiple Agent
- MDPs
  - Single Agent
  - Multiple State
- Matrix Games
  - Multiple Agent
  - Single State

Stochastic Games
- Multiple Agent
- Multiple State
A stochastic game is a tuple \((n, S, A_1 \ldots, T, R_1 \ldots)\), where,

- \(n\) is the number of agents,
- \(S\) is the set of states,
- \(A_i\) is the set of actions available to agent \(i\),
  - \(A\) is the joint action space \(A_1 \times \ldots \times A_n\),
- \(T\) is the transition function \(S \times A \times S \rightarrow [0, 1]\),
- \(R_i\) is the reward function for the \(i^{th}\) agent \(S \times A \rightarrow \mathbb{R}\).
Q-Learning for Zero-Sum Games: Minimax-Q

[Littman, 1994]

- Explicitly learn equilibrium policy.
- Maintain $Q$ value for state/joint-action pairs.
- Update rule:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma V(s')),$$

where,

$$V(s') = \text{Value} \left[ Q(s', \bar{a}) \right]_{\bar{a} \in A}.$$ 

Converges to the game’s equilibrium, with usual assumptions.
Q-Learning for General-Sum Games

[Hu & Wellman, 1998]

- Explicitly learn equilibrium policy.
- Maintain $n$ $Q$ values for state/joint-action pairs.
- Update rule:

$$Q^i(s, a) \leftarrow (1 - \alpha)Q^i(s, a) + \alpha(r^i + \gamma V^i(s')),$$

where,

$$V^i(s') = \text{Value}^i\left[Q(s')\right]_{\bar{a} \in A, \ i=1...n}$$

Does this converge to an equilibrium?
Assumption 1 A Nash equilibrium \((\pi^1(s), \pi^2(s))\) for all matrix games \((Q^1_t(s), Q^2_t(s))\) as well as \((Q^1_*(s), Q^2_*(s))\) satisfy one of the following properties:

1.) The equilibrium is a global optimal.

\[\forall \rho^k \quad \pi^1(s)Q^k(s)\pi^2(s) \geq \rho^1(s)Q^k(s)\rho^2(s)\]

2.) The equilibrium receives a higher payoff if the other agent deviates from the equilibrium strategy.

\[\forall \rho^k \quad \pi^1(s)Q^1(s)\pi^2(s) \leq \pi^1(s)Q^1(s)\rho^2(s)\]
\[\pi^1(s)Q^2(s)\pi^2(s) \leq \rho^1(s)Q^2(s)\pi^2(s)\]
Q-Learning for General-Sum Games

- Proof depends on the update rule being a contraction mapping:
  \[ \forall Q^k \quad \| P_t^k Q^k - P_t^k Q^*_k \| \leq \gamma \| Q^k - Q^*_k \|, \]
  where,
  \[ P_t^k Q^k(s) = r_t^k + \gamma \text{Value}_k \left( Q(s') \right). \]
- I.e., the update function always moves \( Q^k \) closer to \( Q^*_k \), the \( Q \) values of the equilibrium.

Unfortunately, this is not true with their stated assumption.
Counterexample

\[ Q^*(s_0) = (\gamma(1 - \epsilon), \gamma(1 - \epsilon)) \]
\[ Q^*(s_1) = \begin{pmatrix} 1, 1 & 1 - 2\epsilon, 1 + \epsilon \\ 1 + \epsilon, 1 - 2\epsilon & 1 - \epsilon, 1 - \epsilon \end{pmatrix} \]
\[ Q^*(s_2) = (0, 0) \]

\[ Q^* \text{ Satisfies Property 2 of the Assumption.} \]
Counterexample

\[ Q(s_0) = (\gamma, \gamma) \]
\[ Q(s_1) = \begin{pmatrix} 1 + \epsilon, 1 + \epsilon & 1 - \epsilon, 1 \\ 1 + \epsilon, 1 - \epsilon & 1 - 2\epsilon, 1 - 2\epsilon \end{pmatrix} \]
\[ Q(s_2) = (0, 0). \]

\[ ||Q - Q^*|| = \epsilon \]

\[ Q \text{ Satisfies Property 1 of the Assumption.} \]
Counterexample

\[ Q(s_0) = (\gamma, \gamma) \]
\[ Q(s_1) = \begin{pmatrix} 1 + \epsilon, 1 + \epsilon & 1 - \epsilon, 1 \\ 1, 1 - \epsilon & 1 - 2\epsilon, 1 - 2\epsilon \end{pmatrix} \]
\[ Q(s_2) = (0, 0). \]

\[ PQ(s_0) = (\gamma(1 + \epsilon), \gamma(1 + \epsilon)) \]
\[ PQ(s_1) = \begin{pmatrix} 1, 1 & 1 - 2\epsilon, 1 + \epsilon \\ 1 + \epsilon, 1 - 2\epsilon & 1 - \epsilon, 1 - \epsilon \end{pmatrix} \]
\[ PQ(s_2) = (0, 0). \]

\[ ||PQ - PQ^*|| = 2\gamma\epsilon > \epsilon \]
Proof Flaw

• The proof of the Lemma handles the following cases:
  – When \( Q^*(s) \) meets Property 1 of the Assumption.
  – When \( Q(s) \) meets Property 2 of the Assumption.

\[
\begin{array}{c|cc}
Q(s) \text{ meets} & Q^*(s) \text{ meets} \\
\hline
\text{Property 1} & X & \\
\text{Property 2} & X & X \\
\end{array}
\]

• Fails to handle case where \( Q^*(s) \) meets Property 2, and \( Q(s) \) meets Property 1.
  – This is the case of the counterexample.
Strengthening the Assumption

Easy Answer: Rule out the unhandled case.

**Assumption 2** The Nash equilibrium of all matrix games, $Q_t(s)$, as well as $Q_*(s)$ must satisfy property 1 in Assumption 1

OR

the Nash equilibrium of all matrix games, $Q_t(s)$, as well as $Q_*(s)$ must satisfy property 2 of Assumption 1.
Discussion: Applicability of the Theorem

- $Q_t$ satisfies assumption $\not\Rightarrow Q_{t+1}$ satisfies assumption.
  - Problem with their original assumption.
  - Magnified by the further restrictions of new assumption.

- All $Q_t$ values must satisfy same property as the unknown $Q_\ast$.

These limitations prevent a real guarantee of convergence.
Why is convergence in general-sum games difficult?

- Short answer: Small changes in $Q$ values can cause a large change in the state’s equilibrium value.

- But some general-sum games are “easy”:
  - Fully collaborative ($R_i = R_j$ $\forall i, j$) [Claus & Boutilier, 1998]
  - Iterated dominance solvable [Fudenberg & Levine, 1999]

- Other general-sum games are also “easy”.
  - Even games with multiple equilibria.
  - See paper.
Conclusion

There is still much work to be done on learning equilibria in general-sum games.

Thanks to Manuela Veloso, Nicolas Meuleau, and Leslie Kaelbling for helpful discussions and ideas.