CS 748, Spring 2021: Week 1, Lecture 1

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Spring 2021

Multiagent Reinforcement Learning

This lecture is based on:

Markov games as a framework for multiagent reinforcement learning, Michael L. Littman. In Proceedings of the Eleventh International Conference on Machine Learning (ICML 1994), pp. 157–163, Morgan Kaufmann, 1994.

Rock-Paper-Scissors

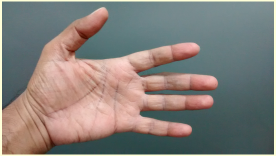
- 2 players: A and O.
- Each has action set {Rock, Paper, Scissors}.

Rock-Paper-Scissors

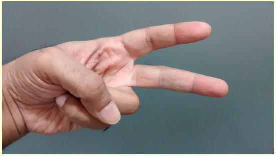
- 2 players: A and O.
- Each has action set {Rock, Paper, Scissors}.

Rock

Paper



Scissors

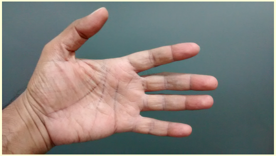


Rock-Paper-Scissors

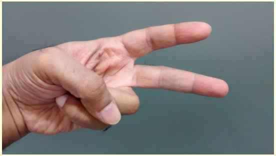
- 2 players: A and O.
- Each has action set {Rock, Paper, Scissors}.

Rock

Paper



Scissors



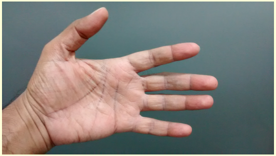
- A and O choose their action simultaneously.

Rock-Paper-Scissors

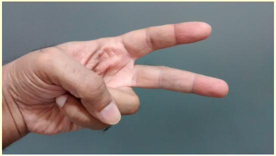
- 2 players: A and O.
- Each has action set {Rock, Paper, Scissors}.

Rock

Paper

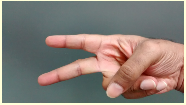


Scissors

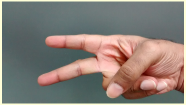


- A and O choose their action simultaneously.
- No winner if both players play same action.
- Rock beats Scissors.
- Scissors beats Paper.
- Paper beats Rock.

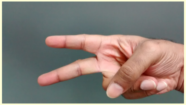
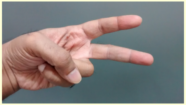
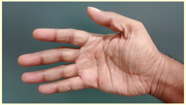
Illustration

Game	A's action	O's action	Winner
1			A

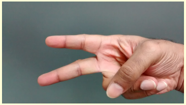
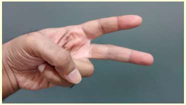
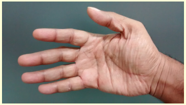
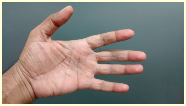
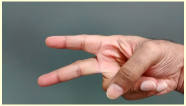
Illustration

Game	A's action	O's action	Winner
1			A
2			–

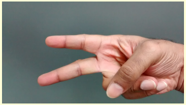
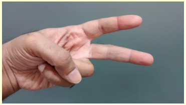
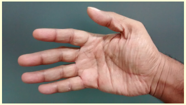
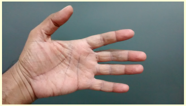
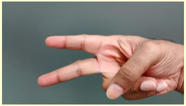
Illustration

Game	A's action	O's action	Winner
1			A
2			–
3			A

Illustration

Game	A's action	O's action	Winner
1			A
2			–
3			A
4			O

Illustration

Game	A's action	O's action	Winner
1			A
2			–
3			A
4			O

How to play this game?!

Multiagent Reinforcement Learning

- Matrix game
- Markov game (Stochastic game)
- Minimax-Q learning algorithm
- Discussion

Matrix Games: Examples

- Row denotes A's action.
- Column denotes O's action.
- Entry denotes A's reward, O's reward.

Rock-Paper-Scissors

	R	P	S
R	0,0	-1,1	1,-1
P	1,1	0,0	-1,1
S	-1,1	1,-1	0,0

Matrix Games: Examples

- Row denotes A's action.
- Column denotes O's action.
- Entry denotes A's reward, O's reward.

Rock-Paper-Scissors

	R	P	S
R	0,0	-1,1	1,-1
P	1,1	0,0	-1,1
S	-1,1	1,-1	0,0

Prisoner's Dilemma

	Silent	Betray
Silent	-1,-1	-3,0
Betray	0,-3	-2,-2

Matrix Games: Examples

- Row denotes A's action.
- Column denotes O's action.
- Entry denotes A's reward, O's reward.

Rock-Paper-Scissors

	R	P	S
R	0,0	-1,1	1,-1
P	1,1	0,0	-1,1
S	-1,1	1,-1	0,0

Prisoner's Dilemma

	Silent	Betray
Silent	-1,-1	-3,0
Betray	0,-3	-2,-2

Chicken

	Swerve	Straight
Swerve	0,0	-1,1
Straight	1,-1	$-10^{10}, -10^{10}$

Matrix Games: Examples

- Row denotes A's action.
- Column denotes O's action.
- Entry denotes A's reward, O's reward.

Rock-Paper-Scissors

	R	P	S
R	0,0	-1,1	1,-1
P	1,1	0,0	-1,1
S	-1,1	1,-1	0,0

Prisoner's Dilemma

	Silent	Betray
Silent	-1,-1	-3,0
Betray	0,-3	-2,-2

Chicken

	Swerve	Straight
Swerve	0,0	-1,1
Straight	1,-1	$-10^{10}, -10^{10}$

- We restrict our attention to zero-sum games such as Rock-Paper-Scissors.

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Assume A is going to execute a piece of code to play the game (might depend on history, might be stochastic).

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Assume A is going to execute a piece of code to play the game (might depend on history, might be stochastic).
- Assume O knows what code A is going to execute, and will play a best response.

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Assume A is going to execute a piece of code to play the game (might depend on history, might be stochastic).
- Assume O knows what code A is going to execute, and will play a best response.
- Minimax optimality: what must A do to give itself as large a reward guarantee as possible?

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Assume A is going to execute a piece of code to play the game (might depend on history, might be stochastic).
- Assume O knows what code A is going to execute, and will play a best response.
- Minimax optimality: what must A do to give itself as large a reward guarantee as possible?
- In general, A must use randomness in action-selection.

Two Player Zero-Sum Matrix Game

- A : set of A's actions.
- O : set of O's actions.
- $\rho : A \times O \to [-R_{\max}, R_{\max}]$.
- Is there such a thing as an "optimal" policy (for A)?

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Assume A is going to execute a piece of code to play the game (might depend on history, might be stochastic).
- Assume O knows what code A is going to execute, and will play a best response.
- Minimax optimality: what must A do to give itself as large a reward guarantee as possible?
- In general, A must use randomness in action-selection.
- **Notation**: Policy = "strategy"; deterministic policy = "pure strategy"; randomised policy = "mixed strategy".

Solving for a Minimax-optimal Strategy

- Let A play strategy $(\pi_\mathsf{R}, \pi_\mathsf{P}, \pi_\mathsf{S})$.
- If A's strategy is fixed, O can play a deterministic best response

$$\operatorname*{argmin}_{o \in O} \left(\pi_\mathsf{R}\rho(\mathsf{R}, o) + \pi_\mathsf{P}\rho(\mathsf{P}, o) + \pi_\mathsf{S}\rho(\mathsf{S}, o)\right).$$

- Consequently A's expected reward is

$$V = \min_{o \in O} \left(\pi_\mathsf{R}\rho(\mathsf{R}, o) + \pi_\mathsf{P}\rho(\mathsf{P}, o) + \pi_\mathsf{S}\rho(\mathsf{S}, o)\right).$$

- Can be solved through an LP with variables $\pi_\mathsf{R}, \pi_\mathsf{P}, \pi_\mathsf{S}, V$.

Maximise V subject to

$$\begin{aligned} \pi_\mathsf{R}\rho(\mathsf{R}, o) + \pi_\mathsf{P}\rho(\mathsf{P}, o) + \pi_\mathsf{S}\rho(\mathsf{S}, o) &\geq V \text{ for } o \in O, \\ \pi_\mathsf{R}, \pi_\mathsf{P}, \pi_\mathsf{S} &\geq 0, \\ \pi_\mathsf{R} + \pi_\mathsf{P} + \pi_\mathsf{S} &= 1. \end{aligned}$$

- Solution: $V^\star = 0; \pi^\star_\mathsf{R} = \pi^\star_\mathsf{P} = \pi^\star_\mathsf{S} = \frac{1}{3}$.

Nash Equilibrium

- Suppose A, 0 play strategies π^A, π^O, respectively.
- π^A, π^O constitute a Nash equilibrium if each is a best response to the other:

$$\pi^O \in \operatorname*{argmin}_{\pi} V(\pi^A, \pi),$$
$$\pi^A \in \operatorname*{argmax}_{\pi} V(\pi, \pi^0).$$

- Satisfied in Rock-Paper-Scissors by

$$\pi^A = \pi^O = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

Nash Equilibrium

- Suppose A, 0 play strategies π^A, π^O, respectively.
- π^A, π^O constitute a Nash equilibrium if each is a best response to the other:

$$\pi^O \in \underset{\pi}{\operatorname{argmin}}\, V(\pi^A, \pi),$$
$$\pi^A \in \underset{\pi}{\operatorname{argmax}}\, V(\pi, \pi^0).$$

- Satisfied in Rock-Paper-Scissors by

$$\pi^A = \pi^O = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

- Finite Matrix games are guaranteed to have at least one Nash Equilibrium.

Nash Equilibrium

- Suppose A, 0 play strategies π^A, π^O, respectively.
- π^A, π^O constitute a Nash equilibrium if each is a best response to the other:

$$\pi^O \in \underset{\pi}{\operatorname{argmin}}\, V(\pi^A, \pi),$$
$$\pi^A \in \underset{\pi}{\operatorname{argmax}}\, V(\pi, \pi^0).$$

- Satisfied in Rock-Paper-Scissors by

$$\pi^A = \pi^O = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

- Finite Matrix games are guaranteed to have at least one Nash Equilibrium.
- In two player zero sum games, π^A and π^O constitute a Nash equilibrium if and only if they are both minimax-optimal.

Multiagent Reinforcement Learning

- Matrix game
- Markov game (Stochastic game)
- Minimax-Q learning algorithm
- Discussion

Two Player Zero-sum Markov Game

- S: set of states.
- A: set of A's actions.
- O: set of O's actions.
- $R : S \times A \times O \to [-R_{\text{max}}, R_{\text{max}}]$.
- $T : S \times A \times O \times S \to [0, 1]$, such that $\sum_{s' \in S} T(s, a, o, s') = 1$ for $s, s' \in S$, $a \in A$, $o \in O$.
- $\gamma \in [0, 1)$: discount factor.

Two Player Zero-sum Markov Game

- S: set of states.
- A: set of A's actions.
- O: set of O's actions.
- $R : S \times A \times O \rightarrow [-R_{\max}, R_{\max}]$.
- $T : S \times A \times O \times S \rightarrow [0, 1]$, such that $\sum_{s' \in S} T(s, a, o, s') = 1$ for $s, s' \in S$, $a \in A$, $o \in O$.
- $\gamma \in [0, 1)$: discount factor.

- Guaranteed unique minimax value function $V^\star : S \rightarrow \mathbb{R}$.

Two Player Zero-sum Markov Game

- S: set of states.
- A: set of A's actions.
- O: set of O's actions.
- $R : S \times A \times O \to [-R_{\max}, R_{\max}]$.
- $T : S \times A \times O \times S \to [0, 1]$, such that $\sum_{s' \in S} T(s, a, o, s') = 1$ for $s, s' \in S$, $a \in A$, $o \in O$.
- $\gamma \in [0, 1)$: discount factor.

- Guaranteed unique minimax value function $V^\star : S \to \mathbb{R}$.
- Note that actions are taken simultaneously.

Two Player Zero-sum Markov Game

- S: set of states.
- A: set of A's actions.
- O: set of O's actions.
- $R : S \times A \times O \to [-R_{\max}, R_{\max}]$.
- $T : S \times A \times O \times S \to [0, 1]$, such that $\sum_{s' \in S} T(s, a, o, s') = 1$ for $s, s' \in S$, $a \in A$, $o \in O$.
- $\gamma \in [0, 1)$: discount factor.

- Guaranteed unique minimax value function $V^\star : S \to \mathbb{R}$.
- Note that actions are taken simultaneously.
- Examples of Markov games: boxing, soccer, carrying furniture up the stairs.

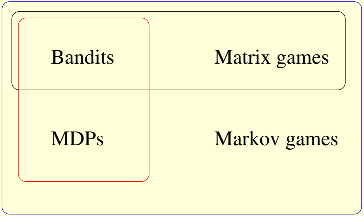
Solving for a Minimax-optimal Strategy

- We solve for A's minimax-optimal strategy using an iterative approach (a form of value iteration).

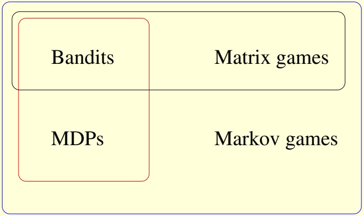
Initialise $V^0 : S \to \mathbb{R}$ arbitrarily. $i \leftarrow 0$.
Do:
  For $s \in S, a \in A, o \in O$:
    $Q(s, a, o) \leftarrow R(s, a, o) + \gamma \sum_{s' \in S} T(s, a, o, s') V^i(s')$.
  For $s \in S$:
    $V^{i+1}(s) \leftarrow \max_{\pi \in \text{PD}(A)} \min_{o \in O} \sum_{a \in A} \pi(a) Q(s, a, o)$.
  $i \leftarrow i + 1$.
While $\|V^i - V^{i-1}\|_\infty > \epsilon$.
Return V^i.

- Converges to $V^\star$; can derive $Q^\star$ and $\pi^\star$ from $V^\star$.
- Similar approach can yield O's minimax-optimal strategy.

Generality of Markov Games

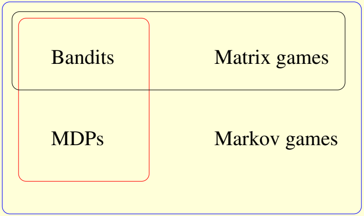


Generality of Markov Games



- Can A learn to play "well" against O in a Markov game (S, A, O, R, T, γ)?

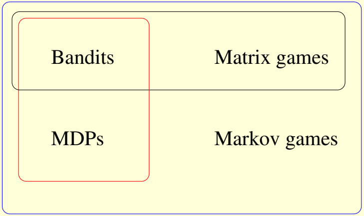
Generality of Markov Games



- Can A learn to play "well" against O in a Markov game (S, A, O, R, T, γ)?
- R and T unknown; players go along a trajectory

$$s^0, a^0, o^0, r^0, s^1, a^1, o^1, r^1, s^2, a^2, o^2, r^2, \ldots.$$

Generality of Markov Games



- Can A learn to play "well" against O in a Markov game (S, A, O, R, T, γ)?
- R and T unknown; players go along a trajectory

$$s^0, a^0, o^0, r^0, s^1, a^1, o^1, r^1, s^2, a^2, o^2, r^2, \ldots.$$

- Can A act according to a minimax-optimal policy in the limit of experience (assuming O performs infinite exploration)?

Multiagent Reinforcement Learning

- Matrix game
- Markov game (Stochastic game)
- Minimax-Q learning algorithm
- Discussion

Minimax-Q Learning Algorithm

- Assume A implements this algorithm.
- Initialise Q^0, V^0, π^0 arbitrarily.
- Take $\alpha_t = \epsilon_t = \frac{1}{t+1}$ for $t \geq 0$.
- Pick actions uniformly at random with probability ϵ_t; follow π^t with remaining probability.

Minimax-Q Learning Algorithm

- Assume A implements this algorithm.
- Initialise Q^0, V^0, π^0 arbitrarily.
- Take $\alpha_t = \epsilon_t = \frac{1}{t+1}$ for $t \geq 0$.
- Pick actions uniformly at random with probability ϵ_t; follow π^t with remaining probability.
- Upon encountering transition $(s^t, a^t, o^t, r^t, s^{t+1})$ for $t \geq 0$:

1. $Q^{t+1}(s^t, a^t, o^t, s^{t+1}) \leftarrow \begin{cases} Q^t(s^t, a^t, o^t, s^{t+1})(1 - \alpha_t) \\ +\alpha_t(r^t + \gamma V^t(s^{t+1})). \end{cases}$

Minimax-Q Learning Algorithm

- Assume A implements this algorithm.
- Initialise Q^0, V^0, π^0 arbitrarily.
- Take $\alpha_t = \epsilon_t = \frac{1}{t+1}$ for $t \geq 0$.
- Pick actions uniformly at random with probability ϵ_t; follow π^t with remaining probability.
- Upon encountering transition $(s^t, a^t, o^t, r^t, s^{t+1})$ for $t \geq 0$:

1. $Q^{t+1}(s^t, a^t, o^t, s^{t+1}) \leftarrow \begin{cases} Q^t(s^t, a^t, o^t, s^{t+1})(1 - \alpha_t) \\ +\alpha_t(r^t + \gamma V^t(s^{t+1})). \end{cases}$
2. Using LP for each $s \in S$, set
   $\pi^{t+1}(s) \leftarrow \operatorname{argmax}_{\pi \in \mathrm{PD}(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) Q^{t+1}(s, a, o)$.

Minimax-Q Learning Algorithm

- Assume A implements this algorithm.
- Initialise Q^0, V^0, π^0 arbitrarily.
- Take $\alpha_t = \epsilon_t = \frac{1}{t+1}$ for $t \geq 0$.
- Pick actions uniformly at random with probability ϵ_t; follow π^t with remaining probability.
- Upon encountering transition $(s^t, a^t, o^t, r^t, s^{t+1})$ for $t \geq 0$:

1. $Q^{t+1}(s^t, a^t, o^t, s^{t+1}) \leftarrow \begin{cases} Q^t(s^t, a^t, o^t, s^{t+1})(1 - \alpha_t) \\ +\alpha_t(r^t + \gamma V^t(s^{t+1})). \end{cases}$
2. Using LP for each $s \in S$, set
   $\pi^{t+1}(s) \leftarrow \text{argmax}_{\pi \in \text{PD}(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) Q^{t+1}(s, a, o)$.
3. For $s \in S$, set
   $V^{t+1}(s) \leftarrow \min_{o \in O} \sum_{a \in A} \pi^{t+1}(s, a) Q^{t+1}(s, a, o)$.

Minimax-Q Learning Algorithm

- Assume A implements this algorithm.
- Initialise Q^0, V^0, π^0 arbitrarily.
- Take $\alpha_t = \epsilon_t = \frac{1}{t+1}$ for $t \geq 0$.
- Pick actions uniformly at random with probability ϵ_t; follow π^t with remaining probability.
- Upon encountering transition $(s^t, a^t, o^t, r^t, s^{t+1})$ for $t \geq 0$:

1. $Q^{t+1}(s^t, a^t, o^t, s^{t+1}) \leftarrow \begin{cases} Q^t(s^t, a^t, o^t, s^{t+1})(1 - \alpha_t) \\ +\alpha_t(r^t + \gamma V^t(s^{t+1})). \end{cases}$
2. Using LP for each $s \in S$, set
   $\pi^{t+1}(s) \leftarrow \text{argmax}_{\pi \in \text{PD}(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) Q^{t+1}(s, a, o)$.
3. For $s \in S$, set
   $V^{t+1}(s) \leftarrow \min_{o \in O} \sum_{a \in A} \pi^{t+1}(s, a) Q^{t+1}(s, a, o)$.

- Induces actions according to minimax-optimal policy in the limit if O visits each state-action pair infinitely often.

Multiagent Reinforcement Learning

- Matrix game
- Markov game (Stochastic game)
- Minimax-Q learning algorithm
- Discussion

Summary and Outlook

- Game theory predates theory of MDPs.
- Question of learning in games more recent.
- Technical issues: nonstationarity, size of joint-action space.
- Desiderata: convergence in self-play, convergence to best response against opponent playing fixed strategy, etc.
- Mechanism design considers how games must be set up so desired group behaviour emerges.
- Social choice theory specifically looks at protocols for surveys/elections.
- Multiagency also a key aspect in the co-evolution of populations.
- Many recent applications: ad auctions, on-line markets, games (Poker, Go, Robot soccer), surveillance.
- Wide-ranging questions: do a course on game theory/multiagent systems!