

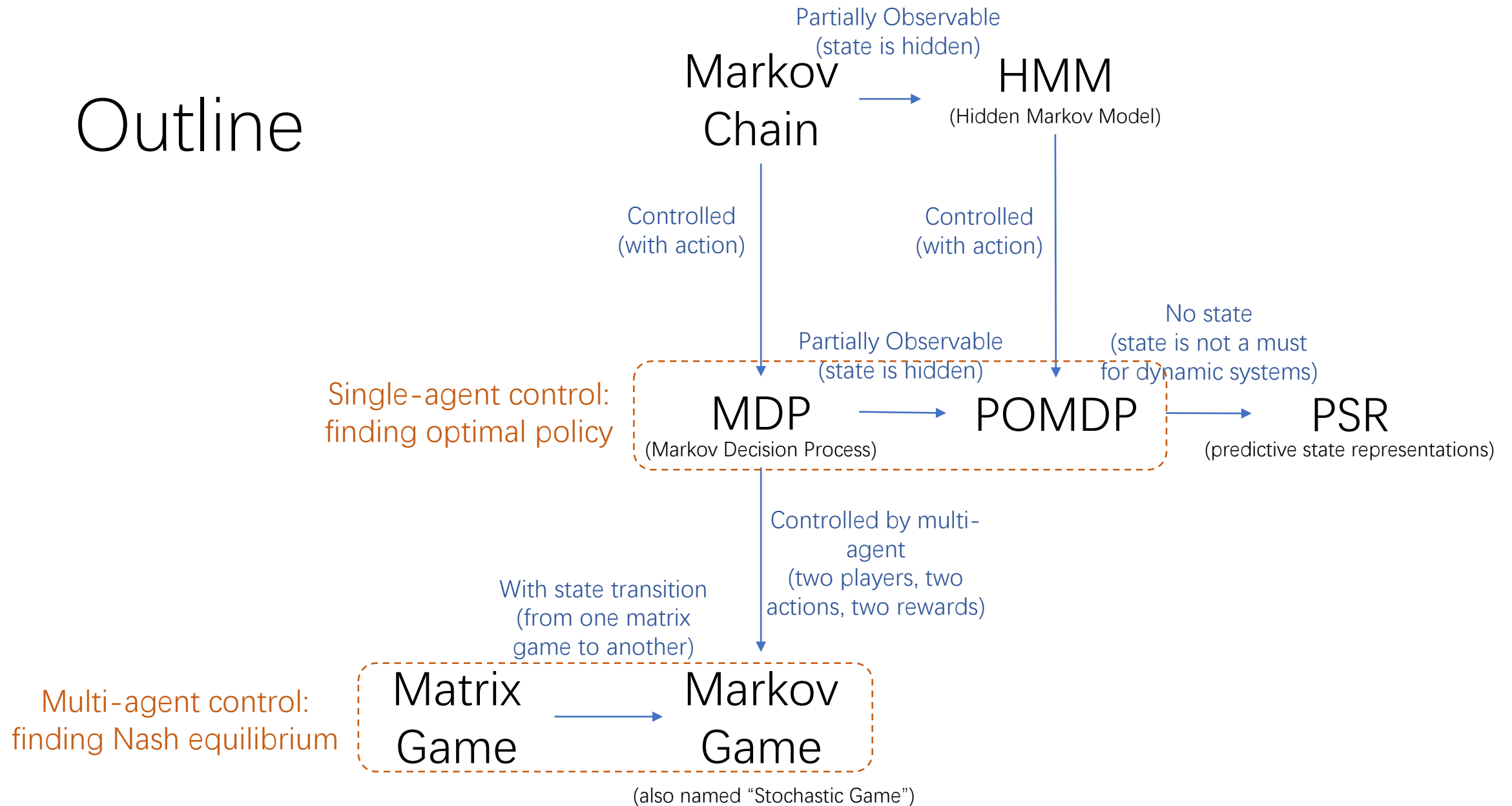
Matrix Game, Markov Game, POMDP, PSR

Xihan Li

Oct 22, 2021

Contents based on <https://sites.google.com/view/cjin/ele524>

Outline



Matrix Game

- A set of players
 - e.g., you (row player, player 1) and your opponent (column player, player 2)
- Each player chooses an action
 - e.g., $\mathcal{A} = \mathcal{B} = \{rock, paper, scissor\}$, you choose $a \in \mathcal{A}$, your opponent choose $b \in \mathcal{B}$
- Each player receives a reward
 - e.g., when you choose $a = rock$ and your opponent choose $b = paper$, you receive reward -1 (lose) and your opponent receive 1 (win)
 - More generally, when you choose a and your opponent choose b , you receive $R_1(a, b)$ and your opponent receive $R_2(a, b)$
- Zero-sum game: $R_1(a, b) + R_2(a, b) + \dots = 0$
 - So we can use a single function $R(a, b)$ to denote the reward in 2-player setting

Matrix Game: policy (strategy)

Your opponent's action b

	$R_1(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	-1	1
	Paper	1	0	-1
	Scissor	-1	1	0

Your reward

Your opponent's action b

	$R_2(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	1	-1
	Paper	-1	0	1
	Scissor	1	-1	0

Your opponent's reward

Here we have action $a \in \mathcal{A}$, what about the policy $\pi(\cdot) \in \Delta_{\mathcal{A}}$?

Different from MDP, we don't have state here.

- MDP: deterministic policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$ or stochastic policy $\pi: \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$
- Matrix Game: **pure strategy** $\mu \in \mathcal{A}, \nu \in \mathcal{B}$ or **mixed strategy** $\mu \in \Delta_{\mathcal{A}}, \nu \in \Delta_{\mathcal{B}}$

$\Delta_{\mathcal{A}}$: Distribution (simplex) over action set \mathcal{A} . E.g., $(0.3, 0.3, 0.4) \in \Delta_{\mathcal{A}}$, which means you have probability 0.3 to play rock or paper, probability 0.4 to play scissor.

E.g.

- Pure strategy: $\mu = \text{rock}$ (you always play rock), $\nu = \text{paper}$ (your opponent always play paper), $R(\mu, \nu) = -1$ (you always lose)
- Mixed strategy: $\mu = (\frac{1}{2}, \frac{1}{2}, 0)$ (you play $\frac{1}{2}$ rock, $\frac{1}{2}$ paper), $\nu = (0, \frac{1}{2}, \frac{1}{2})$ (your opponent play $\frac{1}{2}$ paper, $\frac{1}{2}$ scissor)

Matrix Game: reward

Your opponent's action b

	$R_1(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	-1	1
	Paper	1	0	-1
	Scissor	-1	1	0

Your reward

Your opponent's action b

	$R_2(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	1	-1
	Paper	-1	0	1
	Scissor	1	-1	0

Your opponent's reward

What about the reward $r \in \mathbb{R}$?

Different from MDP, we have separate rewards for each player.

Beside your action, your reward is also determined by what your opponent plays.

- MDP: $r(s, a): \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- Matrix Game: $R_1(a, b), R_2(a, b): \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ (matrices $R_i \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$)

Expected reward for player i : $f_i(\mu, \nu) = \mathbb{E}_{a \sim \mu, b \sim \nu}[R_i(\mu, \nu)] = \sum_{a,b} \mu(a) R_i(\mu, \nu) \nu(b) = \mu^\top R_i \nu$

E.g. Mixed strategy: $\mu = (\frac{1}{2}, \frac{1}{2}, 0)$ (you play $\frac{1}{2}$ rock, $\frac{1}{2}$ paper), $\nu = (0, \frac{1}{2}, \frac{1}{2})$ (your opponent play $\frac{1}{2}$ paper, $\frac{1}{2}$ scissor)

Your expected reward: $\mathbb{E}_{a \sim \mu, b \sim \nu}[R(\mu, \nu)] = \frac{1}{4} \times (-1) + \frac{1}{4} \times 1 + \frac{1}{4} \times 0 + \frac{1}{4} \times (-1) = -\frac{1}{4}$

$$[0.5 \quad 0.5 \quad 0] \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} = -0.25$$

Matrix Game: best response

Your opponent's action b

	$R_1(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	-1	1
	Paper	1	0	-1
	Scissor	-1	1	0

Your reward

Your opponent's action b

	$R_2(a, b)$	Rock	Paper	Scissor
Your action a	Rock	0	1	-1
	Paper	-1	0	1
	Scissor	1	-1	0

Your opponent's reward

What about the reward and the optimal policy π^* ?

In matrix game, our optimal policy (strategy) is relevant to the policy (strategy) of the opponent.

- MDP: optimal policy $\pi^* = \arg \max_{\pi} \mathbb{E}_{a \sim \pi} [\sum r_t]$ (the policy that maximize the cumulated reward)
- Matrix Game: **best response for you** $\mu^*(v) = \arg \max_{\mu} \mathbb{E}_{a \sim \mu, b \sim v} [R_1(a, b)]$ (the strategy that maximize the reward given v), **best response for your opponent** $v^*(\mu) = \arg \max_v \mathbb{E}_{a \sim \mu, b \sim v} [R_2(a, b)]$

E.g.

- When your opponent play $v = (1, 0, 0)$ (always play rock), your best response is $\mu = (0, 1, 0)$ (always play paper) so that $\mathbb{E}_{a \sim \mu, b \sim v} [R(a, b)] = 1$ is maximized.

Matrix Game: Nash equilibrium

		Your opponent's action b		
Your action a	$R_1(a, b)$	Rock	Paper	Scissor
	Rock	0	-1	1
	Paper	1	0	-1
	Scissor	-1	1	0
		Your reward		

		Your opponent's action b		
Your action a	$R_2(a, b)$	Rock	Paper	Scissor
	Rock	0	1	-1
	Paper	-1	0	1
	Scissor	1	-1	0
		Your opponent's reward		

Is there some “optimal policy (strategy)” that does not depend on the opponent's policy (strategy)?

A **Nash Equilibrium** is a strategy (μ, ν) such that neither player will gain anything by deviating from his own strategy while the opposing player continues to play its current strategy.

E.g., $\mu = \nu = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is a Nash equilibrium (you cannot increase your expectation of reward if your opponent plays rock, paper and scissor equality, vice versa.)

Theorem: *Every game with a finite number of players and action profiles has at least one Nash equilibrium.*

Complete Proof: <https://www.cs.ubc.ca/~jiang/papers/NashReport.pdf>

Zero-sum, 2-player Nash equilibrium proof

- Let

$$f(\mu, \nu) = \mathbb{E}_{a \sim \mu, b \sim \nu}[R(\mu, \nu)] = \sum_{a,b} \mu(a) R(\mu, \nu) \nu(b) = \mu^\top R \nu$$

- be your expected reward ($-f(\mu, \nu)$ for your opponent)
- In zero-sum, 2-player setting, a strategy pair (μ^*, ν^*) is a Nash equilibrium if
 - Your expected reward $f(\mu^*, \nu^*) \geq \max_{\mu} f(\mu, \nu^*)$
 - Your opponent's expected reward $-f(\mu^*, \nu^*) \geq \max_{\nu} -f(\mu^*, \nu) \Rightarrow f(\mu^*, \nu^*) \leq \min_{\nu} f(\mu^*, \nu)$

- That is,

$$\max_{\mu} f(\mu, \nu^*) \leq f(\mu^*, \nu^*) \leq \min_{\nu} f(\mu^*, \nu) \quad \forall \mu, \nu \in \Delta_{\mathcal{A}}$$

$$\max_x -f(x) = -\min_x f(x)$$

Which means

$$\min_{\nu} \max_{\mu} f(\mu, \nu) \leq f(\mu^*, \nu^*) \leq \max_{\mu} \min_{\nu} f(\mu, \nu)$$

$$\min_{\nu} [\max_{\mu} f(\mu, \nu)] \leq \max_{\mu} f(\mu, \nu^*)$$

Zero-sum, 2-player Nash equilibrium proof

Lemma:

$$\min_b \max_a f(a, b) \geq \max_a \min_b f(a, b)$$

Proof:

$$f(a, b) \geq \min_b f(a, b) \quad \forall a, b$$

$$\max_a f(a, b) \geq \max_a \min_b f(a, b) \quad \forall b$$

$$\min_b \max_a f(a, b) \geq \max_a \min_b f(a, b)$$



Zero-sum, 2-player Nash equilibrium proof

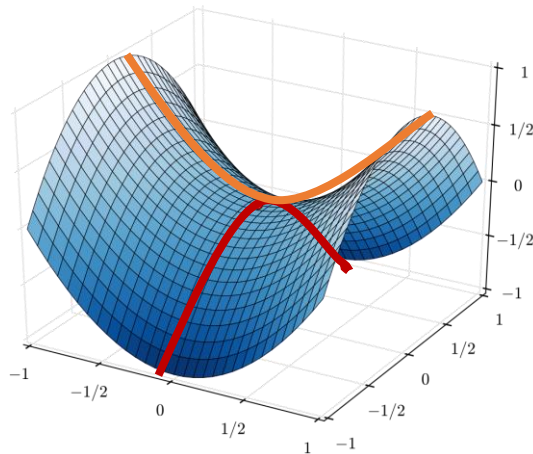
So, the existence of a Nash Equilibrium implies that

$$\min_v \max_\mu f(\mu, v) = f(\mu^*, v^*) = \max_\mu \min_v f(\mu, v)$$

Von Neumann's minimax theorem: Let \mathcal{U}, \mathcal{V} be convex, compact sets, $f: \mathcal{U} \times \mathcal{V} \rightarrow \mathbb{R}$ is a convex-concave continuous function (meaning that $f(\mu, \cdot)$ is convex $\forall \mu$ and $f(\cdot, v)$ is concave $\forall v$). Then

$$\min_v \max_\mu f(\mu, v) = \max_\mu \min_v f(\mu, v)$$

In our case, $f(\mu, v) = \mu^\top R v$ is bi-linear (convex-concave), $\mu \in \Delta_{\mathcal{A}}, v \in \Delta_{\mathcal{B}}$ are simplex (convex)



Both the left and right side are constrained optimization problems

$$\begin{aligned} \min_v \max_\mu \mu^\top R v \quad s.t. \quad \mu \in \Delta_{\mathcal{A}}, v \in \Delta_{\mathcal{B}} \\ \max_\mu \min_v \mu^\top R v \quad s.t. \quad \mu \in \Delta_{\mathcal{A}}, v \in \Delta_{\mathcal{B}} \end{aligned}$$

Which can be transformed as Linear Programming models. By the duality of Linear Programming, the equality also holds.

Finding Nash equilibrium

- (projected) gradient descent ascent

$$x_{t+1} = x_t + \eta \partial_x f(x_t, y_t)$$

$$y_{t+1} = y_t - \eta \partial_y f(x_t, y_t)$$

Markov Games

Matrix Game with state and transitions.

- Each state $s \in \mathcal{S}$ is a Matrix game.
- $P(s'|s, a, b)$ is the transition probability (with multiple actions)
- $r_i(s, a, b)$ is the reward of player $i = 1, 2$ for Matrix Game s when player 1 plays a and player 2 plays b .

Policy:

- MDP: deterministic policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$ or stochastic policy $\pi: \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$
- Matrix Game: pure strategy $\mu \in \mathcal{A}, \nu \in \mathcal{B}$ or mixed strategy $\mu \in \Delta_{\mathcal{A}}, \nu \in \Delta_{\mathcal{B}}$
- Markov Game: deterministic policy $\mu: \mathcal{S} \rightarrow \mathcal{A}, \nu: \mathcal{S} \rightarrow \mathcal{B}$ or stochastic policy $\mu: \mathcal{S} \rightarrow \Delta_{\mathcal{A}}, \nu: \mathcal{S} \rightarrow \Delta_{\mathcal{B}}$
 - very similar to MDP, but have multiple policy for multiple players

Markov Games

Lecture 3: MDP

Value function and expected reward for player i given a state s

$$V_i^{\mu, \nu}(s) = \mathbb{E}_{\mu, \nu}[G_i^t | s_t = s], G_i^t = \sum_{k=t}^T r_i(s, a, b)$$

$$V_h^\pi(s) = \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) | s_h = s \right]$$

State-action value function for player i

$$Q_i^{\mu, \nu}(s, a, b) = \mathbb{E}_{\mu, \nu}[G_i^t | s_t = s, a_t = a, b_t = b]$$

$$Q_h^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) | s_h = s, a_h = a \right]$$

Bellman Equation for player i

$$Q_i^{\mu, \nu}(s, a, b) = r_i(s, a, b) + \mathbb{E}_{s' \sim P(\cdot | s, a, b)} V_i^{\mu, \nu}(s')$$

$$\begin{aligned} V_i^{\mu, \nu}(s) &= \sum_{a \in \mathcal{A}, b \in \mathcal{B}} \mu(s, a) Q_i^{\mu, \nu}(s, a, b) \nu(s, b) \\ &= \mu(s)^\top Q_i^{\mu, \nu}(s) \nu(s) \end{aligned}$$

$$\begin{cases} V_h^\pi(s) &= \sum_{a \in \mathcal{A}} Q_h^\pi(s, a) \pi_h(a | s) \\ Q_h^\pi(s, a) &= r_h(s, a) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot | s, a)} V_{h+1}^\pi(s') \end{cases}$$

Similarly, when we are in a zero-sum, 2-player setting, we can write $r(s, a, b)$ directly without specifying the player.

Markov Games: best response & Nash Equilibrium

- If the policy of your opponent ν is given, the Markov Game becomes an MDP with optimal policy $\mu^*(\nu) = \operatorname{argmax}_{\mu} V_1^{\mu,\nu}$, which is called the best response. Similarly, $\nu^*(\mu) = \operatorname{argmax}_{\nu} V_2^{\mu,\nu}$.
- For 2-player zero-sum game, $V_2^{\mu,\nu} = -V_1^{\mu,\nu}$ (so we just write $V_1^{\mu,\nu} = V^{\mu,\nu}$)
- If (μ^*, ν^*) is a Nash equilibrium, then

$$V^{\mu^*,\nu^*} \geq \max_{\mu} V^{\mu,\nu^*}, -V^{\mu^*,\nu^*} \geq \max_{\nu} -V^{\mu^*,\nu} \Rightarrow V^{\mu^*,\nu^*} \leq \min_{\nu} V^{\mu^*,\nu}$$

$$\max_{\mu} V^{\mu,\nu^*} \leq V^{\mu^*,\nu^*} \leq \min_{\nu} V^{\mu^*,\nu}$$

Like the case in Matrix Game, we have

$$\min_{\nu} \max_{\mu} V^{\mu,\nu} = V^{\mu^*,\nu^*} = \max_{\mu} \min_{\nu} V^{\mu,\nu}$$

Replace $f(\mu, \nu)$ in Matrix Game to cumulated reward $V^{\mu,\nu}$

Markov Game: finding Nash equilibrium

Lecture 3: MDP

$$\text{Bellman Equation: } V(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q(s, a)$$

For all state $s \in \mathcal{S}$:

$$\begin{aligned} V(s) &= \max_{\pi \in \Delta_{\mathcal{A}}} \sum_{a \in \mathcal{A}} \pi(a|s) Q(s, a), \text{ in which } Q(s, a) = r(s, a) + \mathbb{E}_{s' \sim P(s'|s, a)} V(s') \\ &= \max_{a \in \mathcal{A}} Q(s, a), \text{ since } \pi(s) \in \Delta_{\mathcal{A}}, \text{ vector } \pi \text{ will be a one-hot vector when (greedily) maximized} \end{aligned}$$

For a fixed opponent policy ν , the Markov Game becomes an MDP, and we can find the best response via value iteration above

$$\text{Bellman Equation: } V^{\mu, \nu}(s) = \sum_{a \in \mathcal{A}, b \in \mathcal{B}} \mu(s, a) Q^{\mu, \nu}(s, a, b) \nu(s, b) = \sum_{a \in \mathcal{A}} \mu(s, a) [\sum_{b \in \mathcal{B}} Q^{\mu, \nu}(s, a, b) \nu(s, b)]$$

For all state $s \in \mathcal{S}$:

$$\begin{aligned} V^{\mu, \nu}(s) &= \max_{\mu \in \Delta_{\mathcal{A}}} \sum_{a \in \mathcal{A}, b \in \mathcal{B}} \mu(s, a) Q^{\mu, \nu}(s, a, b) \nu(s, b), \text{ in which } Q^{\mu, \nu}(s, a, b) = r(s, a, b) + \mathbb{E}_{s' \sim P(\cdot|s, a, b)} V^{\mu, \nu}(s') \\ &= \max_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} Q^{\mu, \nu}(s, a, b) \nu(s, b) \end{aligned}$$

To find the Nash equilibrium, we use

For all state $s \in \mathcal{S}$:

$$V^{\mu, \nu}(s) = \min_{\nu \in \Delta_{\mathcal{B}}} \max_{\mu \in \Delta_{\mathcal{A}}} \sum_{a \in \mathcal{A}, b \in \mathcal{B}} \mu(s, a) Q^{\mu, \nu}(s, a, b) \nu(s, b)$$

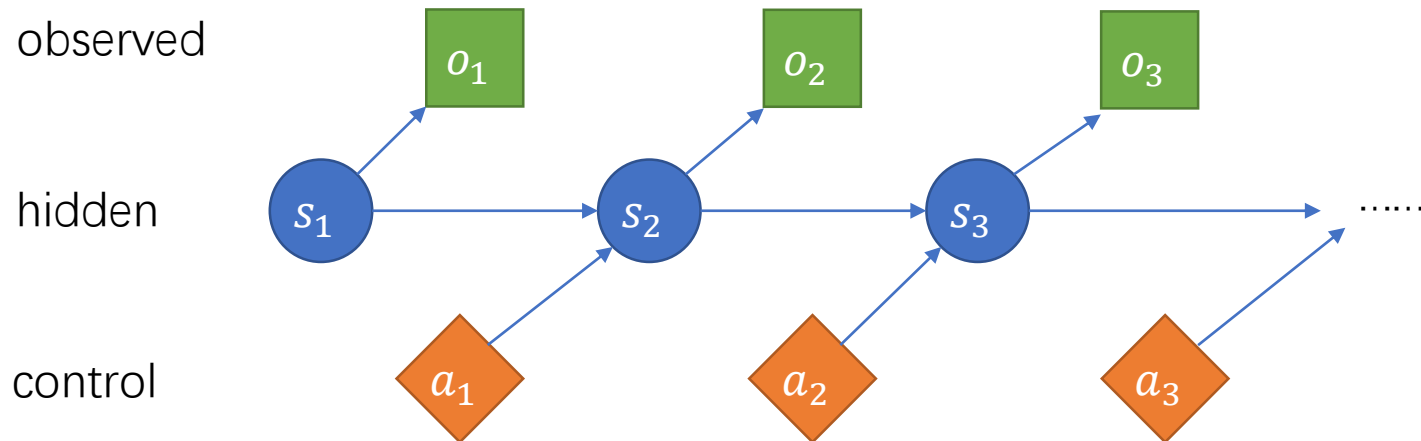
For each state $s \in \mathcal{S}$, finding a Nash Equilibrium for the Matrix Game with reward matrix $Q^{\mu, \nu}(s) \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{B}|}$

Partially Observable MDP

- The state of many applications are not fully observable
 - Poker (you don't know your opponent's hand)
 - StarCraft (fog)
- Need more general models to describe the problem

POMDP: adds observation \mathcal{O} to the MDP formalization

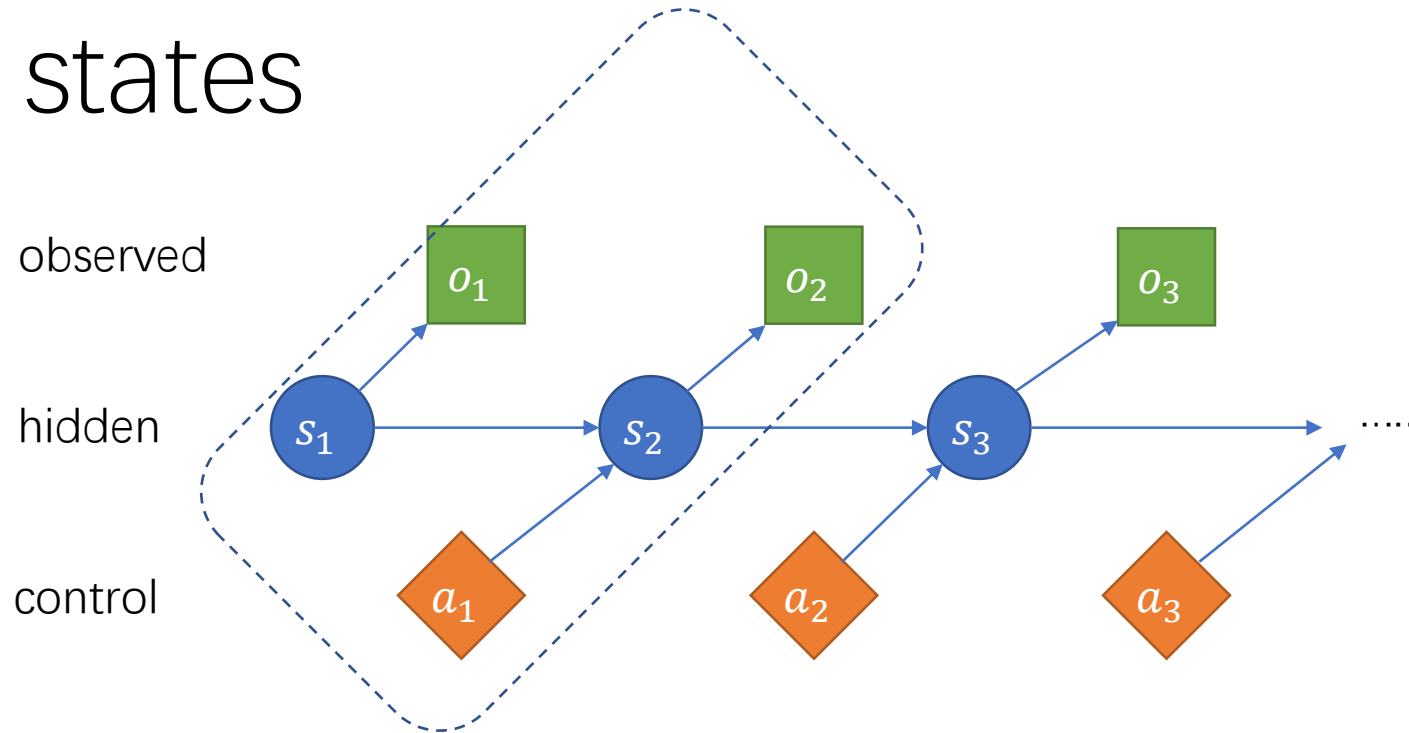
- $\mathbb{O}(o|s)$: observation probability (under a state s , the possibility to observe o)
- $r(o)$: reward is a function of observation.



“History” and policy

- Instead of state s , decisions is based on the entire history
$$\tau_t = (o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o_t)$$
- Policy is a mapping from history to (distribution of) action, $\pi(\tau) \in \Delta_{\mathcal{A}}$
- Bellman Equation
 - $V^\pi(\tau_t) = \sum_{a \in \mathcal{A}} \pi(a|\tau_t) Q^\pi(\tau_t, a)$
 - $Q^\pi(\tau_t, a_t) = \mathbb{E}_{o_{t+1} \sim P(\cdot|\tau_t, a_t)} [r(o_{t+1}) + V^\pi(\{\tau_t, a_t, o_{t+1}\})]$
- Bellman Optimality Equation
 - $V^*(\tau_t) = \max_{a \in \mathcal{A}} Q^*(\tau_t, a)$
 - $Q^*(\tau_t, a_t) = \mathbb{E}_{o_{t+1} \sim P(\cdot|\tau_t, a_t)} [r(o_{t+1}) + V^*(\{\tau_t, a_t, o_{t+1}\})]$
 - Optimal Policy $\pi^*(\tau_t) = \arg \max_{a \in \mathcal{A}} Q^*(\tau_t, a)$
- Planning in POMDP in general cannot be done computational efficiently.

Belief states



- History gives a distribution over s_2
 - If two histories generate the same belief states, then there should not be difference in the future. (i.e., earlier view has redundancy)
- Belief state $b_t \in \Delta_{\mathcal{S}}$ is a simplex (distribution) over all state \mathcal{S}
- Policy $\pi: \Delta_{\mathcal{S}} \rightarrow \Delta_{\mathcal{A}}$ only needs to rely on sufficient statistics
- POMDP \Leftrightarrow belief-state MDP

Belief states

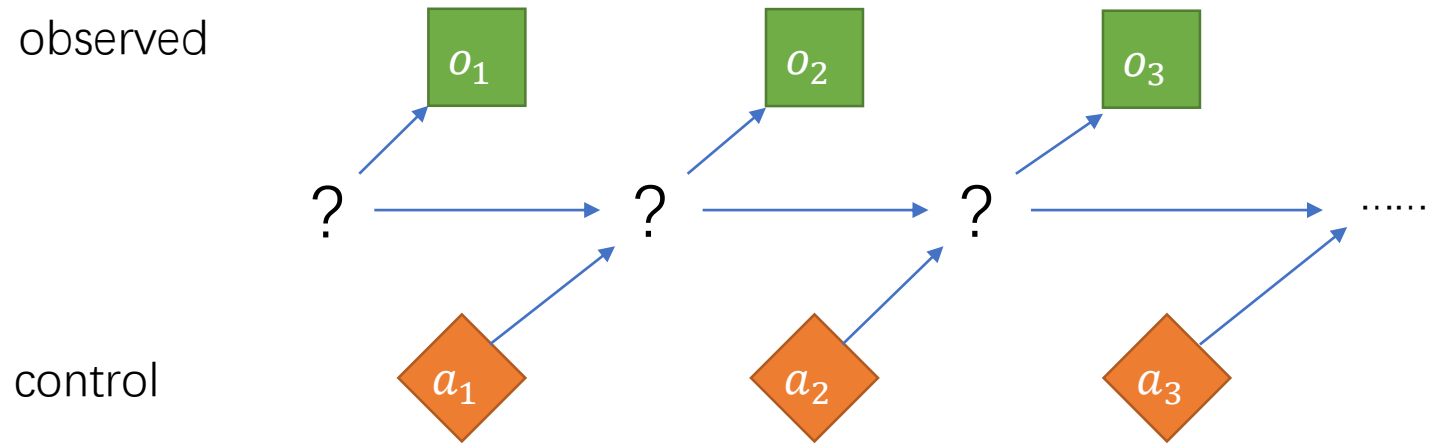
- Update on belief states: the probability of state $s_{t+1} \in \mathcal{S}$ in b_{t+1} is

$$b_{t+1}(s_{t+1}) = \frac{P(s_{t+1}, o_{t+1} | a_t, b_t)}{P(o_{t+1} | a_t, b_t)}$$

So b_{t+1} is a function of b_t, a_t, o_{t+1} ($b_{t+1} = f(b_t, a_t, o_{t+1})$)

- Bellman Equation
 - $V^\pi(b_t) = \sum_{a \in \mathcal{A}} \pi(a | b_t) Q^\pi(b_t, a)$
 - $Q^\pi(b_t, a_t) = \mathbb{E}_{o_{t+1} \sim P(\cdot | b_t, a_t)} [r(o_{t+1}) + V^\pi(f(b_t, a_t, o_{t+1}))]$
- In general, $V^*(b)$ is not a linear function in b
 - Still in general computationally intractable

Predictive State Representation



- State is not a must in dynamic systems
 - In practical applications, there may or may not exist interpretable hidden states. They may not be unique, nor “intrinsic”
- Define a test $t = (a^1, o^1, \dots, a^k, o^k)$ of length k
- System-dynamics vector:
$$p(t) = \Pr(o_1 = o^1, \dots, o_k = o^k | a_1 = a^1, \dots, a_k = a^k)$$
- Once we know system dynamics vector, we know everything about the dynamic system

System-dynamic Matrix

$$p(t) = \Pr(o_1 = o^1, \dots, o_k = o^k | a_1 = a^1, \dots, a_k = a^k)$$

- It will be easier to see the structure in matrix form
- Test $t = (a^1, o^1, \dots, a^k, o^k)$, history $h = (a_1, o_1, \dots, a_l, o_l)$

$$p(t|h) = \Pr(o_{l+1} = o^1, \dots, o_{l+k} = o^k | h, a_{l+1} = a^1, \dots, a_{l+k} = a^k)$$

Concatenate
 h and t

$$P(t|h) = \frac{p(ht)}{p(h)}$$

System-dynamic matrix can be computed by system-dynamic vector

For POMDP with $|S|$ hidden states, $\text{rank}(\text{SD matrix}) \leq |S|$

Proof. $p(t|h) = \sum_s p(t|s)p(s|h) = b[h]^\top u_t$ (s-dimensional inner product)

Fact: There exists dynamic system whose $\text{rank}(\text{SD matrix}) = 3$, but cannot be represented by any finite POMDP

Empty set \emptyset →

	$t_0, \dots,$	t_i, \dots
h_0	$P(t_i h_j)$	
.		
.		
.		
h_j		
.		
.		
.		

Core test Q and Predictive State Representation $\psi(h)$

- $Q = \{q_1, \dots, q_k\}$, full column rank
- $\psi(h) = [p(q_1|h), \dots, p(q_k|h)]$
- Then $p(t|h) = m_t^\top \psi(h)$
 - Predicting a new column t using core set.
 - Linear coefficient m_t should not depends on h
- $\psi(h)$ is called Predictive State Representation of h
 - A sufficient statistics, similar to the role of belief state