

Advanced Topics in Reinforcement Learning

Lecture 20: Multi-agent Learning II

Josiah Hanna

University of Wisconsin — Madison

Announcements

- Work on final projects.
- Read “Deep Reinforcement Learning from Human Preferences”
- Grades (mostly) updated.
 - Homework
 - Participation

Mid-Course Evaluation

- Thanks for filling it out!
- Discussion points:
 - Reading responses
 - Homework
 - Class pacing / Topics
 - Exam

Learning Outcomes

After today, you will be able to:

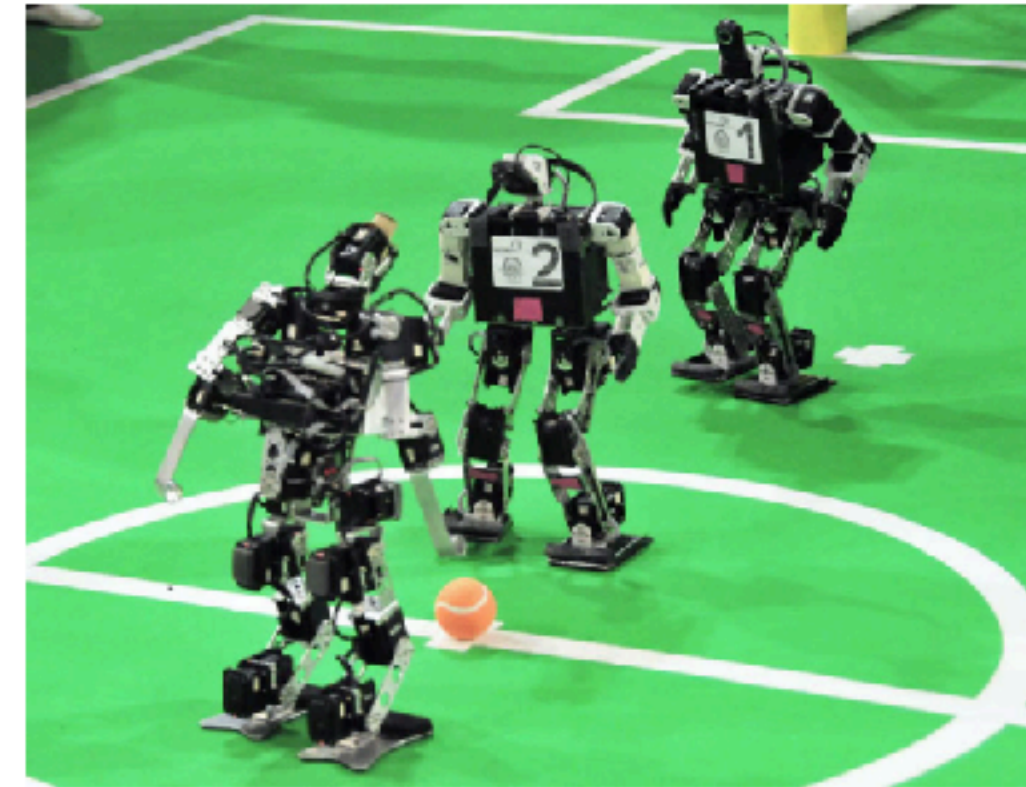
1. Compare and contrast single- and multi-agent RL in terms of challenges and solution concepts.
2. Compare and contrast the strengths of different MARL approaches.
3. Identify when different MARL approaches are suitable.

Multi-Agent Systems

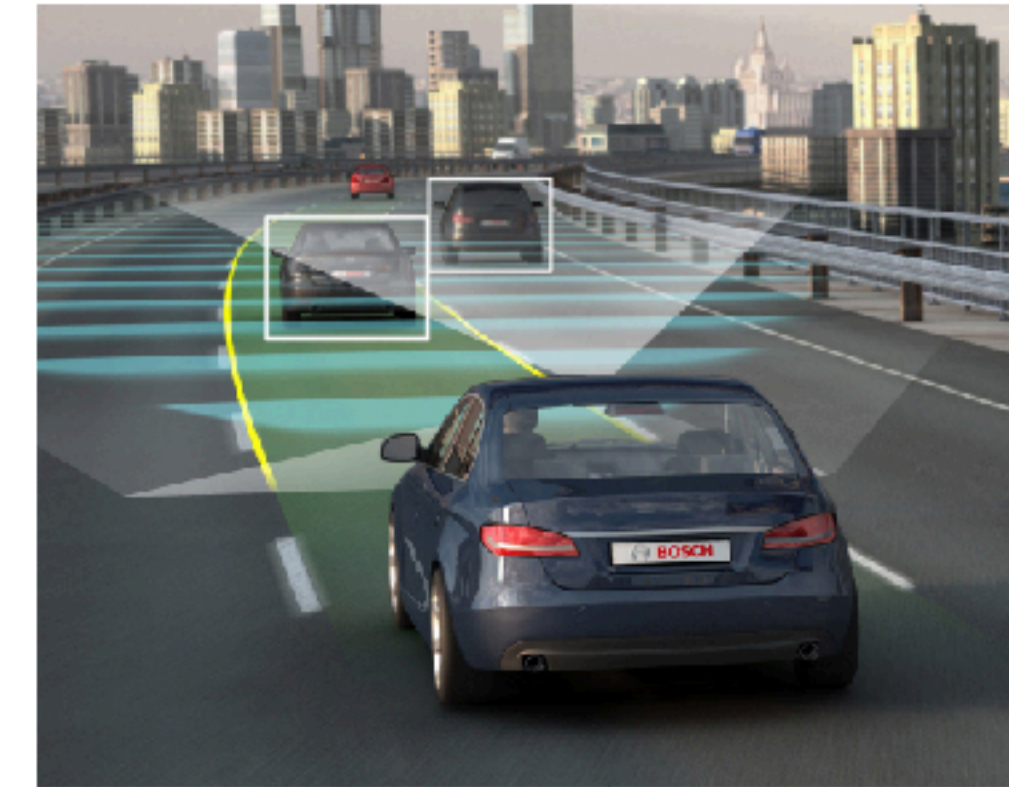
Games



Robot soccer



Autonomous cars



Negotiation/markets



Wireless networks



Smart grid



Challenges in Multi-Agent Learning

- Multi-agent credit assignment.
- Curse of multiple agents.
- Non-stationarity in learning.
- Equilibrium Selection.

Stochastic Games

- Set of states \mathcal{S} .
- For each agent i :
 - Action set \mathcal{A}_i .
 - Reward function, $r_i : \mathcal{S} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n \rightarrow \mathbf{R}$.
- Transition function, $p : \mathcal{S} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n \times \mathcal{S} \rightarrow [0,1]$.
- Discount factor γ .

Interaction in Stochastic Games

- Begin in state s_0 .
- At time t :
 - Each agent chooses action according to $\pi(A_t = a | S_t)$.
 - Each agent receives reward $r_i(S_t, A_t^1, \dots, A_t^n)$.
 - Transition to next state.
- How does this affect Markov property?

What do we want to converge to?

- Each agent wants to maximize reward but doing so depends on what other agents do.
 - Convergence defined in terms of policy profiles, $\pi = (\pi_1, \dots, \pi_n)$.
 - Other forms of convergence were discussed in the reading. Why might they be useful?
- If all use the same reward function, then the optimal policy profile is to just maximize the expected return in each state.
- If not, many different solution concepts exist. Some examples:
 - Minimax optimality
 - Nash equilibrium
 - Pareto Optimality

Minimax Optimality

- A policy is minimax optimal for an agent if it has the best worst-case value.
- Typically considered in two player zero-sum games.
 - Two agents and $r_1(s, a_1, a_2) = -r_2(s, a_1, a_2)$.
- Agent 1 selects policy π ; all other agents select the policy that makes π as bad as possible for Agent 1.

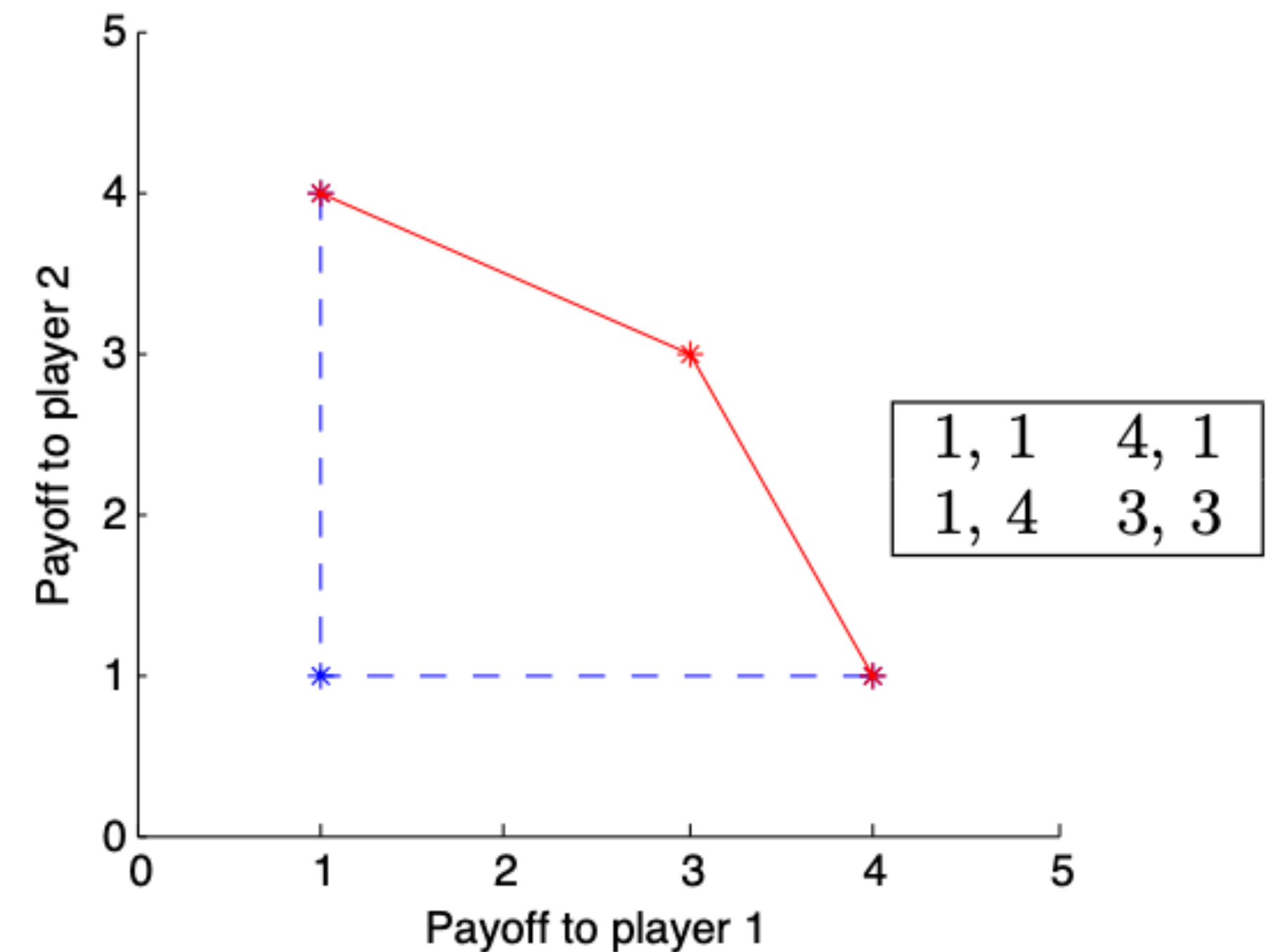
Nash Equilibrium

- A policy profile is a Nash equilibrium if no agent has an incentive to change their policy (mutual best-response).
- Formally, profile π is a Nash equilibrium if $\forall i, \pi' v_{\pi'}^i(s) \leq v_{\pi}^i(s)$ where π' is identical to π except for agent i 's policy.
- Assumes all agents are rational.

	C	D
C	-1,-1	-5,0
D	0,-5	-3,-3

Pareto Optimality

- Cannot improve one agent's value without decreasing another agent's value.
- Formally, a policy profile, π , is Pareto-optimal in state s if there is no other profile, π' such that $\forall i, v_{\pi'}^i(s) \geq v_{\pi}^i(s)$ and $\exists i, v_{\pi'}^i(s) > v_{\pi}^i(s)$.
- Not necessarily, fair in distribution.



Independent Learning

- Simplest MARL algorithm is for each agent to pretend other agents are part of environment and run single-agent RL.
- Lose theoretical guarantees of single-agent RL; still can work in practice.
 - Example: OpenAI's Dota team.
- Shortcomings:
 - Single-agent RL converges to deterministic policy but may need a stochastic policy for optimality in Markov / Stochastic games.
 - May never converge due to non-stationarity.
 - High variance action-value updates due to lack of multi-agent credit assignment.
 - Sub-optimal equilibrium selection.

Centralized Learning

- Treat **cooperative** multi-agent RL problem as one big single-agent problem.
- Learn a policy that takes as input the state of all agents and outputs an action for each agent.
- Example: Deepmind's Star Craft playing agent.
- Shortcomings:
 - Curse of multiple agents.
 - Agents must either share a reward or agent rewards must be turned into a single reward.
 - Observations of all agents are needed to compute an action for any single agent.
- Main benefit: (kind of) avoids multi-agent credit assignment and non-stationarity problems.

Centralized Training / Decentralized Execution

- Objective: take advantage of centralized training but enable each agent to operate independently of others.
- Counterfactual Multi-agent Policy Gradients (COMA) implements this idea with policy gradient learning.
- Each agent learns a policy $\pi_{\theta_i}(a | s)$ with gradient ascent on θ_i .

$$\bullet \quad \nabla_{\theta_i} J(\theta) = \left[Q(s, a_1, \dots, a_i, \dots, a_n) - \sum_a \pi_{\theta_i}(a | s) Q(s, a_1, \dots, a, \dots, a_n) \right] \nabla_{\theta_i} \log \pi_{\theta_i}(a_i | s)$$

Baseline is independent of agent i's action

Game-Theoretic Reinforcement Learning

- What if different agent's have different rewards?
- Why can we not simply learn $Q_i(s, a_1, \dots, a, \dots, a_n)$ for agent i and take actions with $\arg \max_a Q_i(s, a_1, \dots, a, \dots, a_n)$?
 - Non-stationary if others are learning.
 - We don't know what actions will be taken by other agents.
- Game-Theoretic RL uses various solution types from game theory to prescribe how other agents will act.

Game-Theoretic Reinforcement Learning

- Assume all agent's are rational w.r.t. their own current action-value functions.
- Agent i maintains an action-value function for all other agents.
- At each state, action-value functions induce a normal form game.

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

- Solution of normal form games is a policy profile,
 $\pi = (\pi_1, \dots, \pi_n)$.

Minimax-Q uses minimax solution (Littman, 1994)

Nash-Q uses Nash equilibrium (Hu and Wellman, 2003)

CE-Q uses correlated equilibrium (Greenwald and Hall, 2003)

- Use this profile to prescribe how other agents will act in
 $\arg \max_a Q_i(s, a_1, \dots, a, \dots, a_n)$.

Minimax Q-learning

- Standard Q-learning:

- $Q(s, a) \leftarrow Q(s, a) + \alpha(R + \gamma \max_{a'} Q(s', a') - Q(s, a))$

- Minimax Q-learning:

- $V(s) = \max_{\pi \in \Delta(\mathcal{A}_1)} \min_{a_2} \sum_{a_1} \pi(a_1 | s) Q(s, a_1, a_2)$

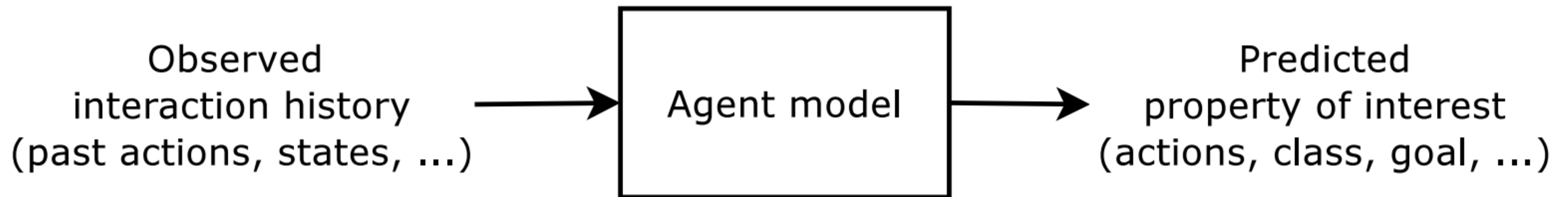
- $Q(s, a_1, a_2) \leftarrow Q(s, a_1, a_2) + \alpha(R + \gamma V(s') - Q(s, a_1, a_2))$

Minimax Q-learning

	MR		MM		QR		QQ	
	% won	games	% won	games	% won	games	% won	games
vs. random	99.3	6500	99.3	7200	99.4	11300	99.5	8600
vs. hand-built	48.1	4300	53.7	5300	26.1	14300	76.3	3300
vs. MR-challenger	35.0	4300						
vs. MM-challenger			37.5	4400				
vs. QR-challenger					0.0	5500		
vs. QQ-challenger							0.0	1200

Opponent Modelling

- Game-theoretic RL assumes that other agents will act rationally or worst-case.
- Instead we can try to predict what others might do and then play best response.



Policy Self-Play

- Where do opponents come from for training?
- Policy self-play uses the main agent's policy as the opponent's policy.
 - Idea: as the policy improves, the opponent also improves.
 - ...but might get stuck in cycles or chatter between different non-dominant policies.
- Can mitigate this by keeping around past versions of the opponent's policy and also training against those.

Alex's Presentation

- Mastering the Game of Stratego with Model-free Multi-agent Reinforcement Learning
- De Vylder et al. (2022)
- [Slides](#)

Summary

- Multi-agent RL aims to scale RL to environments with multiple, possibly learning agents.
- Often requires algorithm changes to overcome MARL challenges.
 - Centralized training / decentralized execution.
 - Game-theoretic RL.
 - Opponent modelling.
 - Self-play