Advanced Topics in Reinforcement Learning

Lecture 24: Multi-agent Learning II

Josiah Hanna
University of Wisconsin — Madison

Based on Slides from Stefano Albrecht's MARL lectures

Announcements

- Next week: Offline RL
- Final projects due < two weeks.
- Course evaluation is now available.

Yoon's Presentation

- Slides

Challenges in Multi-Agent Learning

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

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; still can work in practice.
  - Example: Alpha Go, 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.

Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? De Witt et al. 2020.

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: 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 $\pi_{\theta_i}(a \mid s)$ with gradient ascent on θ_i.

- $\nabla_{\theta_i} J(\theta) = (Q(s, a_1, .., a_i, .., a_n) - \boxed{\sum_a \pi_{\theta_i}(a \mid s) Q(s, a_1, .., a, .., a_n)}) \nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid s)$

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 $Q_i(s, a_1, \ldots, a, \ldots, a_n)$ for agent i and take actions with $\arg\max_a Q_i(s, a_1, \ldots, a, \ldots, a_n)$?
    - 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.
- Solution of normal form games is a policy profile, $\pi = (\pi_1, \ldots, \pi_n)$.
- Use this profile to prescribe how other agents will act in $\arg\max_a Q_i(s, a_1, \ldots, a, \ldots, a_n)$.

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

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)

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:
  - $V(s) = \max_{\pi \in \Delta(\mathscr{A}_1)} \min_{a_2} \sum_{a_1} \pi(a_1 \mid s) Q(s, a_1, a_2)$
  - $Q(s, a_1, a_2) \leftarrow Q(s, a_1, a_2) + \alpha(R + \gamma V(s') - Q(s, a_1, a_2))$

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.

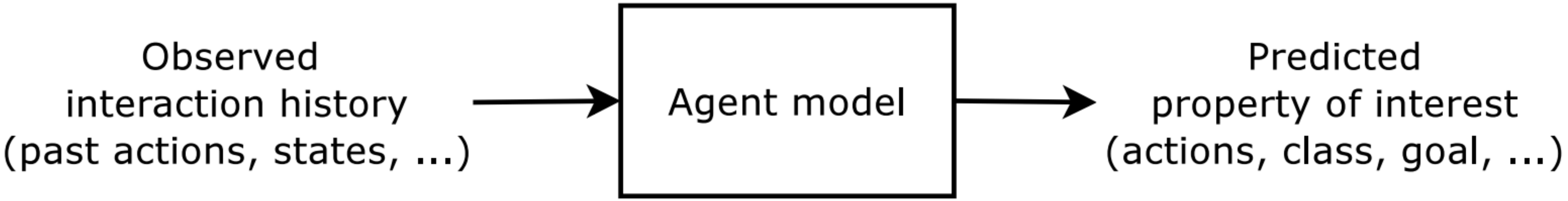

Figure Credit: Stefano Albrecht

Self-Play

- Where do opponents come from for training?
- “Markov Games” paper evaluated different training-evaluation combinations.
- Basic 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.

Matthew’s Presentation

- 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

Action Items

- Offline RL reading for next week.
- Good luck on your final project.