Advanced Topics in Reinforcement Learning

Lecture 19: Multi-agent Learning I

Josiah Hanna
University of Wisconsin — Madison

Based on Slides from Stefano Albrecht's MARL lectures

Announcements

- Midterm course evaluation (link on Piazza) — due tonight!
- Work on final projects.
- Start reading “Deep Reinforcement Learning from Human Preferences”

Final Projects

- Check project page: https://pages.cs.wisc.edu/~jphanna/teaching/25fall_cs839/project.html
- One ask that is to be added: lightning talks on last day of class.
- Clarify about asks ahead of time!

Learning Outcomes

After today, you will be able to:

1. Explain the goal of using state abstraction in RL.

2. Formulate multi-agent RL problems.

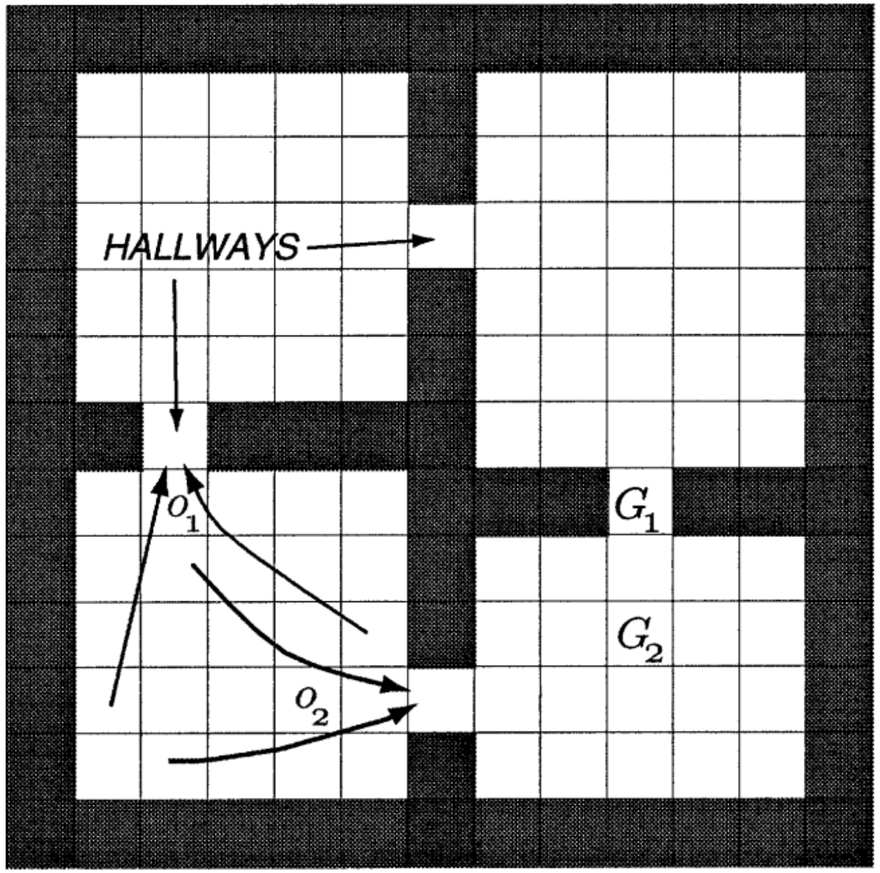
3. Compare and contrast single- and multi-agent RL in terms of challenges and solution concepts.

Abstraction Review

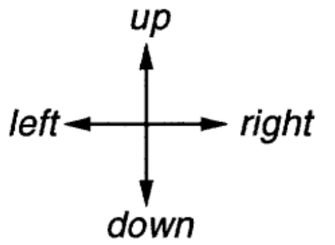
- People complete tasks by planning, learning, and acting at different level of abstraction.
  - Aids credit assignment and exploration.
- Behaviors are modular and re-used across tasks.
  - Transfer learning; subtask learning
- Different states may be functionally the same.

Types of Abstraction

- Temporal Abstraction
- State Abstraction



4 stochastic primitive actions



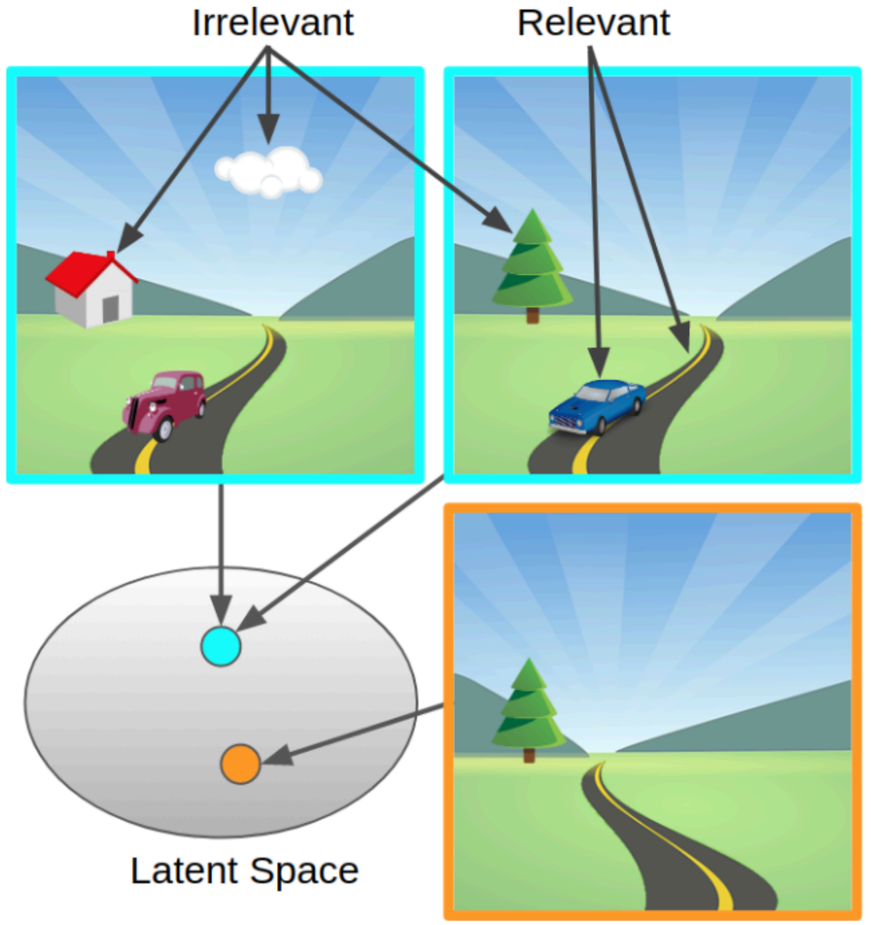
Fail 33% of the time

8 multi-step options

(to each room's 2 hallways)

State Abstraction

- Real life problems have many states
  - But differences between states may be superficial and not affect optimal decision-making.
- A state abstraction removes irrelevant state information to promote generalization and faster learning.



Credit: Amy Zhang

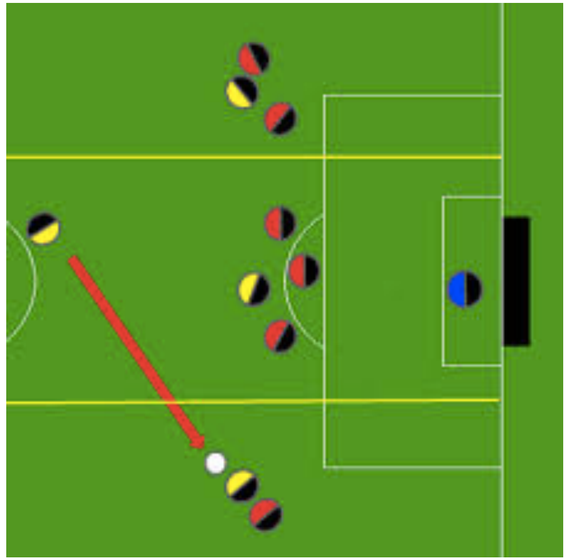
Deep Bisimulation for Control. Zhang et al. 2018.

State Abstraction

- Define a state abstraction function as $\phi : \mathcal{S} \rightarrow \overline{\mathcal{S}}$.
- $\phi(s) \in \overline{\mathcal{S}}$ is an abstract state. We call $\mathcal{S}$ the ground state-space.
- The state abstraction function partitions the ground state-space into disjoint sets.
- Induces a new MDP $\langle \overline{\mathcal{S}}, \mathcal{A}, \bar{r}, \bar{p} \rangle$:
  - $\bar{r}(\bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} w(s) r(s, a)$ $w(s)$ is a state weighting function.
  - $\bar{p}(\bar{s}' \mid \bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} \sum_{s' \in \phi^{-1}(\bar{s}')} w(s) p(s' \mid s, a)$

State Abstraction

- Weighting function is necessary to define a Markov reward and transitions.



Types of State Abstraction

- Many choices for ϕ. What properties should ϕ have?
  - Depends on what “irrelevant” means!
- Model-irrelevance: if two states, s_1 and s_2 are grouped together ($\phi(s_1) = \phi(s_2)$), then s_1 and s_2 have identical rewards and probabilities of leading to any other abstract state.
  - This property is also called bisimulation.
- $\pi^\star$-irrelevance: if two states are grouped together then the optimal action is the same in both.
- Some other choices in between: q_π- irrelevance, $q_\star$-irrelevance, $a^\star$-irrelevance.

Towards a Unified Theory of State Abstraction for MDPs. Li et al. 2006.

Deep State Abstraction

- Partitioning the state space may be difficult with high-dimensional state spaces.
- Instead, learn state abstractions with multi-layer neural networks.
- Define $\phi(s)$ as the non-linear function defined by the first k layers of a network.
- Much recent work on attempting to learn ϕ that exhibits abstraction properties.
  - "MICo: Improved representations via sampling-based state similarity for Markov decision processes." Castro et al. 2022.
  - "Learning Invariant Representations for Reinforcement Learning without Reconstruction." Zhang et al. 2021.
  - "DeepMDP: Learning Continuous Latent Space Models for Representation Learning." Gelada et al. 2019.

Yiheng’s Presentation

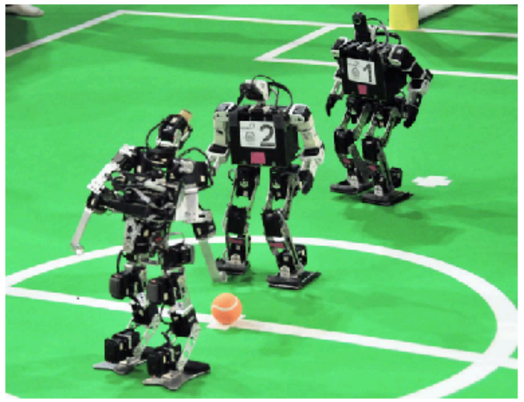
Hindsight Experience Replay

- Andrychowicz et al. 2017
- Slides

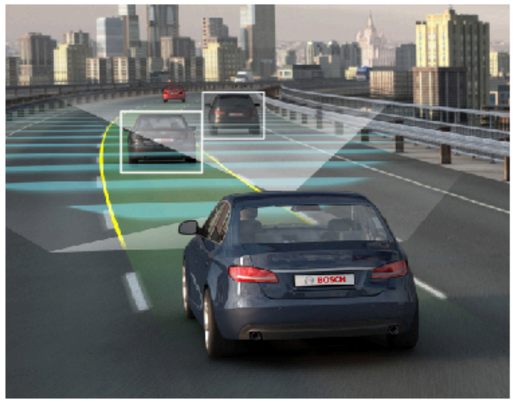
Multi-Agent Systems

Games

Robot soccer



Autonomous cars



Negotiation/markets

Wireless networks

Smart grid

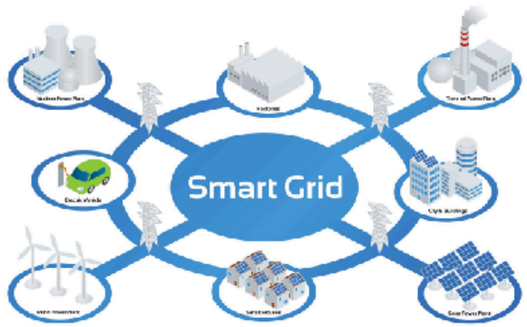


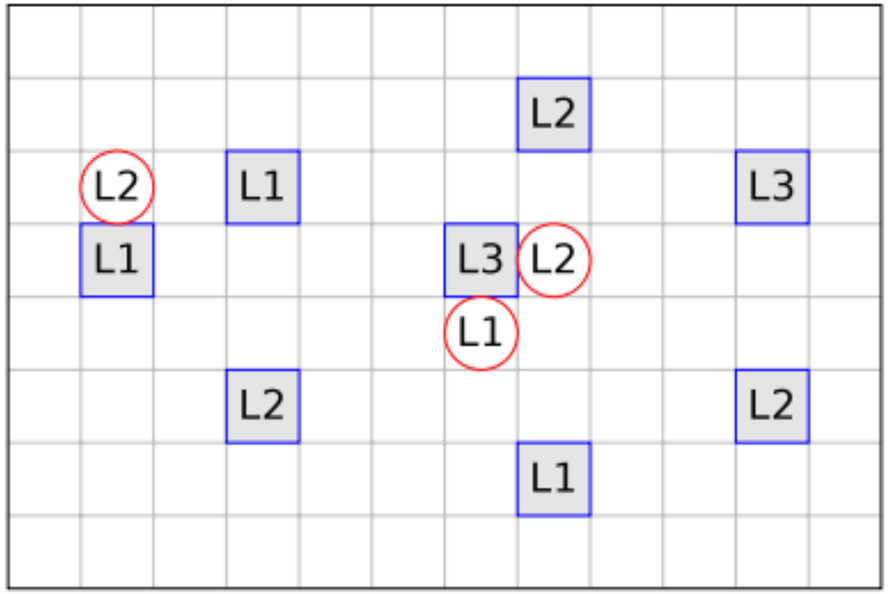
Image Credit: Stefano Albrecht

Challenges in Multi-Agent Learning

- Multi-agent credit assignment.
- Curse of multiple agents.
- Non-stationarity in learning.
- Different agents may have different objectives.
    - Need new solution concepts.

Multi-agent Credit Assignment

- All single-agent RL algorithms must solve temporal credit assignment.
  - Which actions contributed to eventual rewards received.
- Now each agent's rewards depend on what other agents do.
  - Did my action contribute when a reward was received?



Non-Stationarity in Multi-Agent Learning

- So far we have assumed that the environment's transition dynamics are stationary (unchanging over time).
- With learning, the environment appears non-stationary from the view of individual agents.
- Thus, the true action-values for any policy are also non-stationary.

Curse of Many Agents

- What if we just learn a policy that outputs an action for all agents?
  - Size of action space grows (possibly exponentially with number of agents).
  - Size of state space might grow.
  - Application communication constraints.
- Multi-agent RL decomposes a large RL problem into smaller, coupled problems.
- …but agents must coordinate action choices.

Stochastic Games

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

Interaction in Stochastic Games

- Begin in state s_0.
- At time t:
  - Each agent chooses action according to $\pi(A_t = a \mid S_t)$.
  - Each agent receives reward $r_i(S_t, A_t^1, \ldots A_t^n)$.
  - 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, $\pi = (\pi_1, \ldots, \pi_n)$.
- 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.
- Solution concept pursued in “Markov games as a framework for multi-agent reinforcement learning.”

Nash Equilibrium

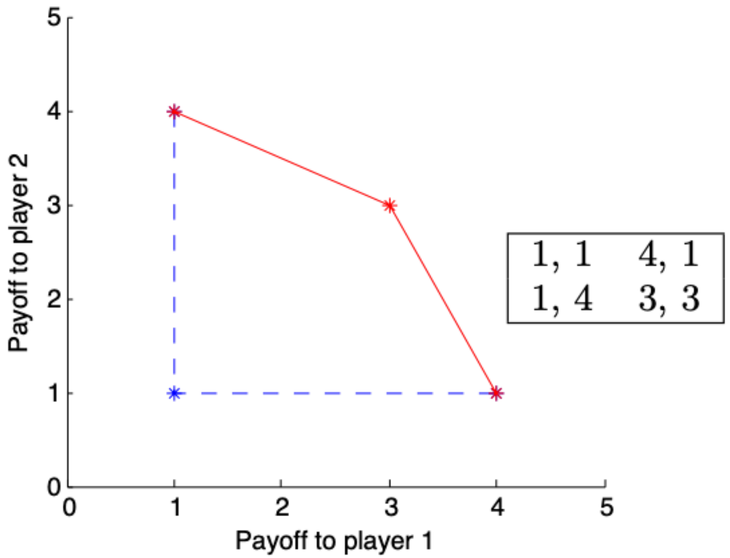
- A policy profile is a Nash equilibrium if no agent has an incentive to change their policy.
- Formally, profile π is a Nash equilibrium if $\forall i, \pi' \; v^i_{\pi'}(s) \leq v^i_{\pi}(s)$ where π' is identical to π except for agent i's policy.
- Assumes all agents are rational.

	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 $\forall i, v^i_{\pi'}(s) \geq v^i_{\pi}(s)$ and $\exists i, v^i_{\pi'}(s) > v^i_{\pi}(s)$.



Dexter’s Presentation

Human-level play in the game of Diplomacy by combining language models with strategic reasoning

- Bakhtin et al. 2022
- Slides

Summary

- Multi-agent RL aims to scale RL to environments with multiple, possibly learning agents.
- New challenges in MARL:
  - Credit assignment
  - Non-stationarity
- New solution concepts:
  - Minimax optimality, Pareto optimality, Nash equilibrium

Action Items

- Read “Deep Reinforcement Learning from Human Preferences”
- Good luck on your final project!