Advanced Topics in Reinforcement Learning

Lecture 20: Hierarchical RL I

Josiah Hanna
University of Wisconsin — Madison

Announcements

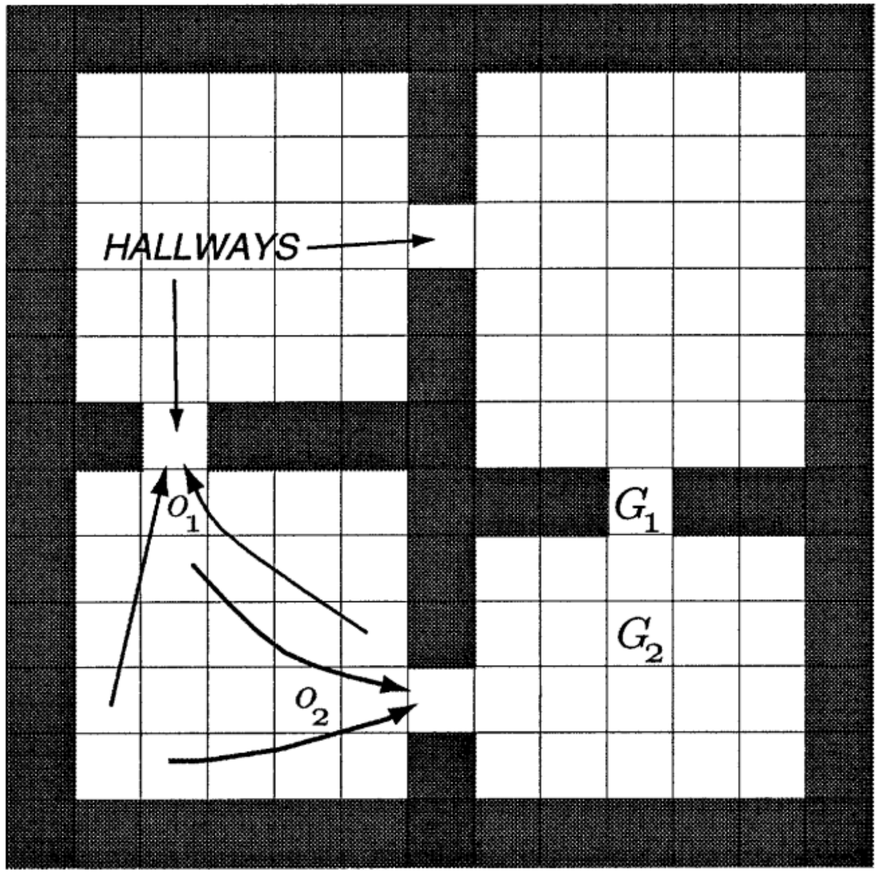
- Homework 4 due Thursday.
- Next week: Ethics, Reproducibility, and Evaluation

Motivation for Abstraction

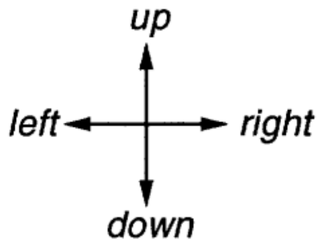
- People complete tasks by planning, learning, and acting at different level of abstraction.
  - Aids credit assignment and exploration.
- Behaviors are modular and re-used across tasks.
  - Transfer learning; subtask learning
- Different states may be functionally the same.
- Abstraction as perception.

Types of Abstraction

- State Abstraction
- Temporal Abstraction



4 stochastic primitive actions



Fail 33% of the time

8 multi-step options

(to each room's 2 hallways)

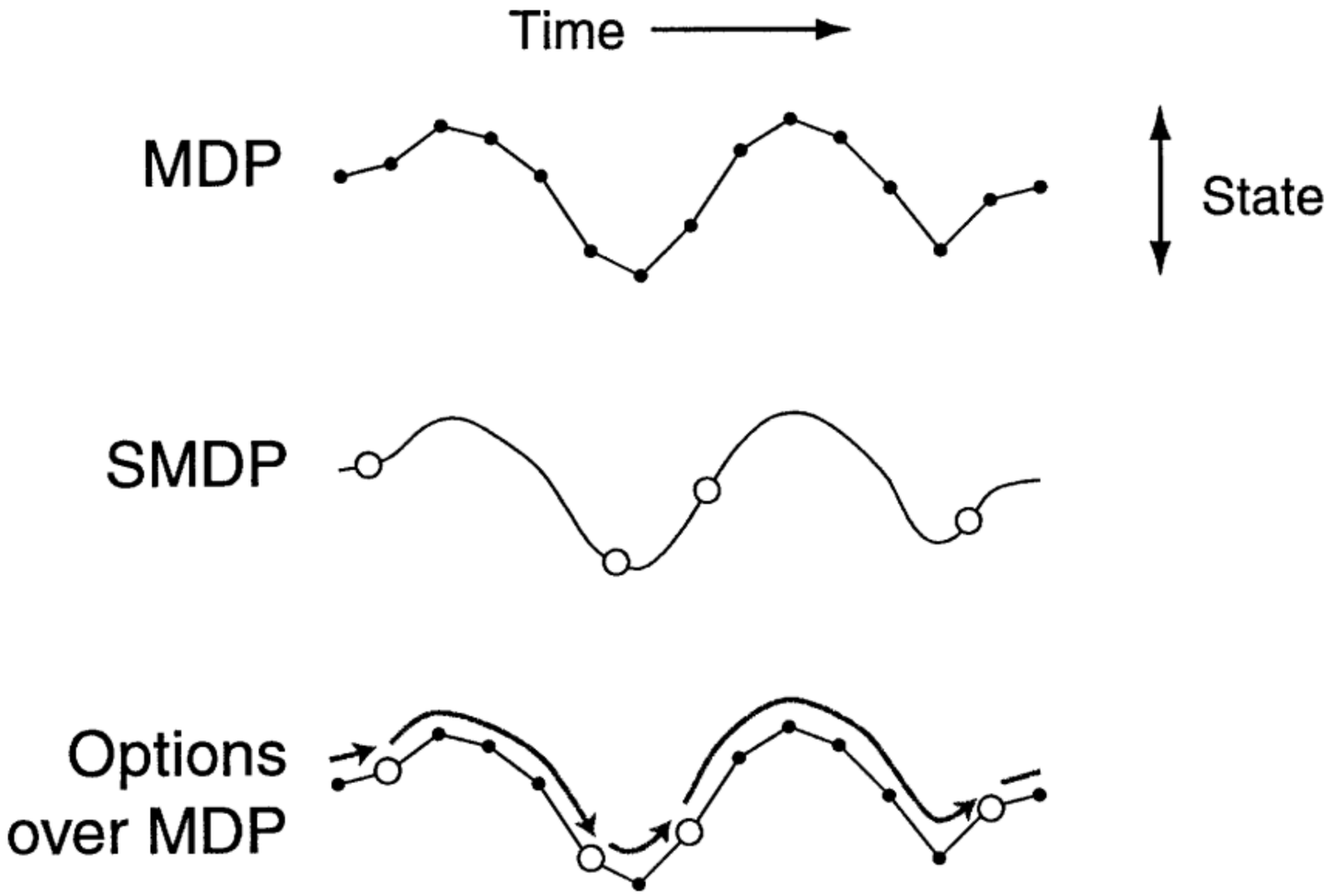
Semi-MDPs

- A formalism that allows actions to last for varying amounts of time.
  - $\mathcal{S}$: same state set.
  - $\mathcal{O}$: Option set.
  - $p(s', r, t \mid s, o)$: Transition probability (includes time).
  - $r(s', s, t)$: Reward function (includes time).
  - γ: discount factor.

Options Framework

- Minimal change to MDP formalism to permit temporal abstraction.
- An option is:
  - $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$: a policy.
  - $\beta : \mathcal{S} \rightarrow [0,1]$: termination probability.
  - $\mathcal{I} \subset \mathcal{S}$: initiation set.
- If agent is in $s \in \mathcal{I}$, then can execute option o which means following π until termination which occurs with probability $\beta(S_t)$ at each step t.
- $\mathcal{E}(\pi, s, t)$: the event that option π was initiated at time t in state s.

Options Framework



Policies

- Markov policies over options: $\mu : \mathcal{S} \times \mathcal{O} \rightarrow [0,1]$.
- Policies over options determine an underlying flat policy which outputs primitive actions.
  - Flat policy is (usually) non-Markovian. Why?
    - Action selection at each time-step depends on what option is being executed and thus history.
    - Not fully non-stationary policies; only depend on history since current option was selected.

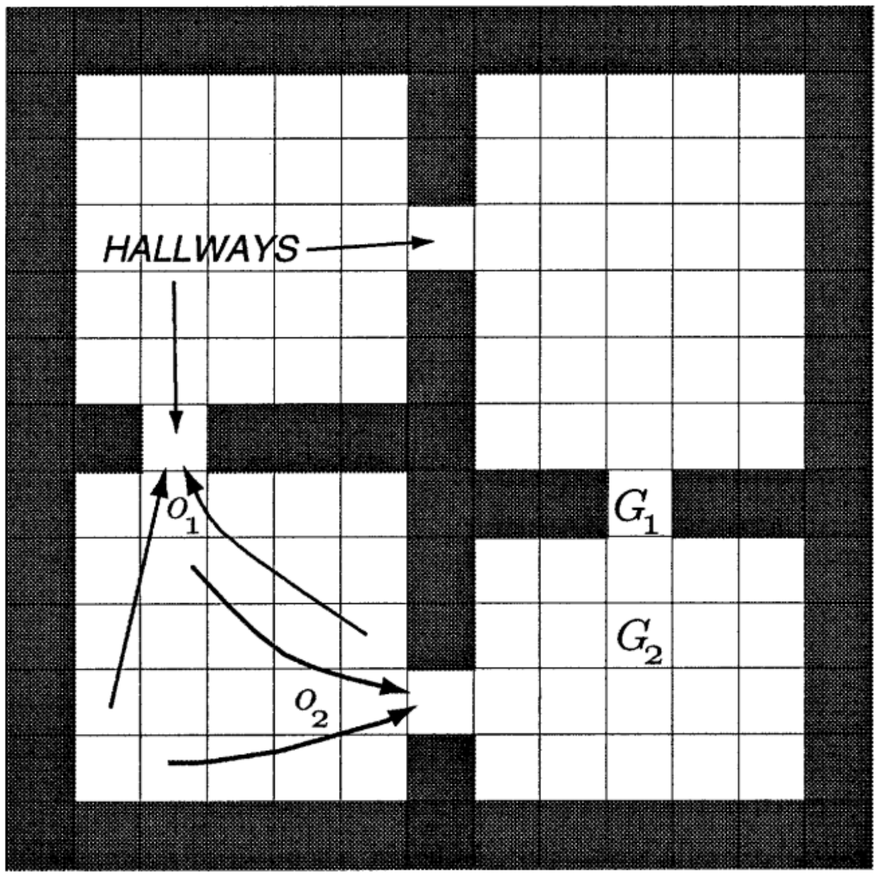
Multi-Time Models and Value Functions

- $r_s^o = \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^k R_{t+k} \,|\, \mathcal{E}(o, s, t)]$
- $p_{ss'}^o = \sum_{k=1}^{\infty} p(s', k)\gamma^k$
- Action-value functions become *option-value* functions:
  - $q_\mu(s, o) = \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \,|\, \mathcal{E}(o\mu, s, t)].$
  - $q_\mu(s, o) = r_s^o + \sum_{s'} p_{ss'}^o \sum_{o' \in \mathcal{O}} \mu(s', o') q_\mu(s', o')$

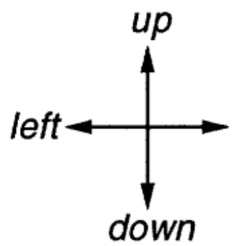
Learning and Planning

- With recursive action-value definitions, learning and planning easily extend from MDPs:
  - Use value iteration with a known or learned $p^o_{ss'}$ and r^o_s.
  - Use Q-learning for model-free learning.

Examples



4 stochastic primitive actions

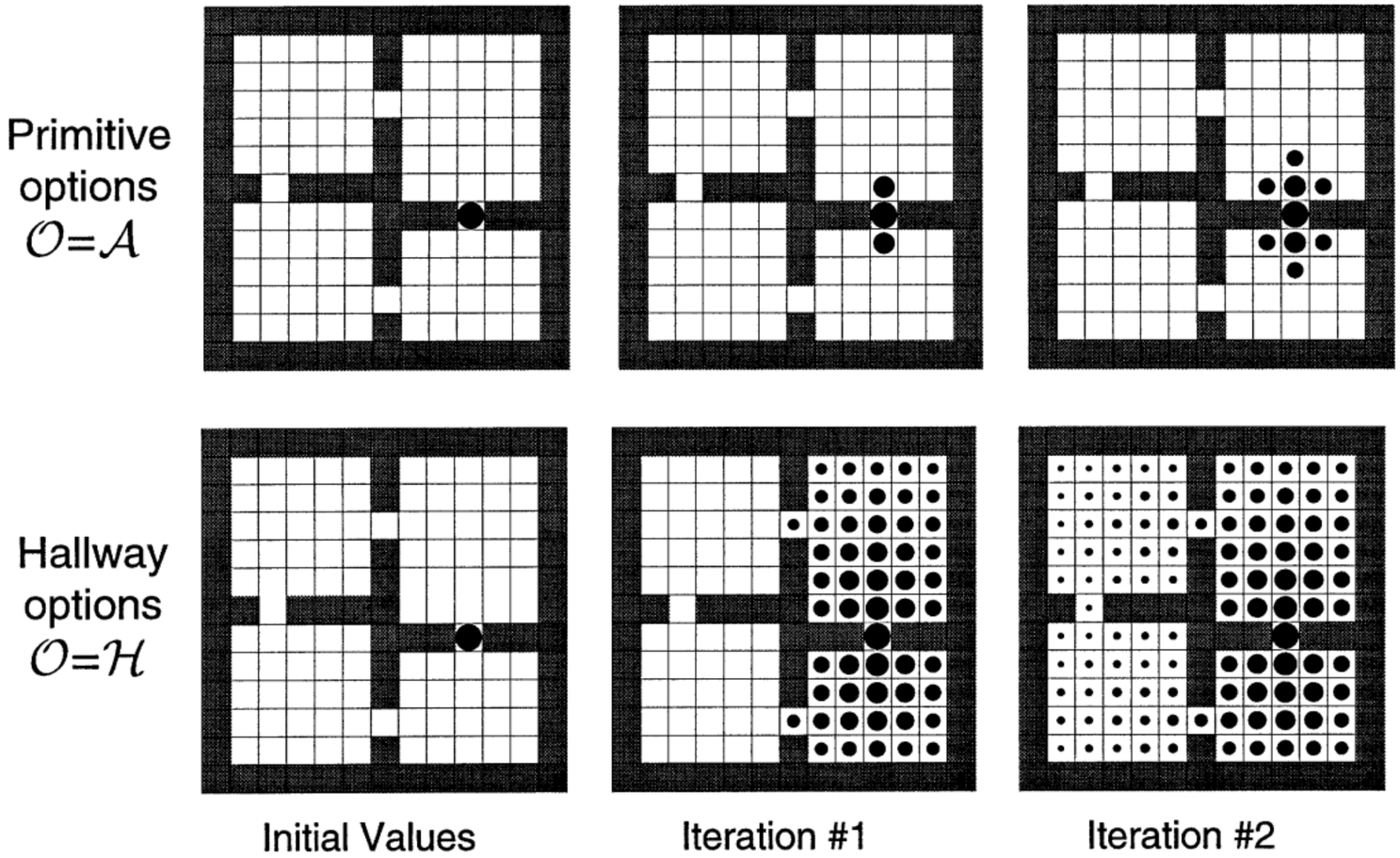


Fail 33% of the time

8 multi-step options

(to each room's 2 hallways)

Examples


Primitive options $\mathcal{O}=\mathcal{A}$
Hallway options $\mathcal{O}=\mathcal{H}$
Initial Values
Iteration #1
Iteration #2

Intra-Option Learning

- Key benefit of the options framework over semi-MDPs is the ability to “open up” and modify options.
  - Interrupt an option before termination if another option has higher predicted value (Theorem 2 of reading guarantees improvement).
  - Learn about or improve options:
    - Build multi-time models for planning (section 5).
    - Learn option-values more efficiently (section 6).
    - Improve option policy (section 7).

Where do options come from?

- Manually defined based on domain knowledge.
- Learned from experience
  - Can specify sub-goal rewards and learn an option (i.e., a policy) that maximizes sub-goal reward.
  - Option-Critic architecture learns options based on original problem reward.

Option Critic Architecture. Bacon et al. 2018.

Options Today

- The options framework is perhaps the most well-known formalism for hierarchical RL.
  - Not the only choice!
- Alternative skill formalism in some works:
  - $\pi_{\texttt{high}}(z \mid s)$ and $\pi_{\texttt{low}}(a \mid s, z)$.
- Intra-option learning and reasonable sub-goals are still open challenges with no widely accepted one method.

Option Critic Architecture. Bacon et al. 2018.

Summary

- Temporal abstraction facilitates exploration and credit-assignment.
- Options framework enables extending concepts, theory, and algorithms from MDPs to temporal abstraction.
- Key questions:
  - How are options acquired?
  - How much benefit can we derive from first learning options and then learning the optimal policy over options?

Action Items

- Homework 4 due Thursday!