Temporal Abstraction in RL

How can an agent represent stochastic, closed-loop, temporally-extended courses of action? How can it act, learn, and plan using such representations?

- HAMs (Parr & Russell 1998; Parr 1998)
- MAXQ (Dietterich 2000)
- Options framework (Sutton, Precup & Singh 1999; Precup 2000)

Outline

- Options
- MDP + options = SMDP
- SMDP methods
- Looking inside the options

Markov Decision Processes (MDPs)

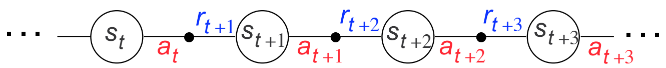
S: set of states of the environment

A(*s*): set of actions possible in state *s,* for all *s*∈*S*

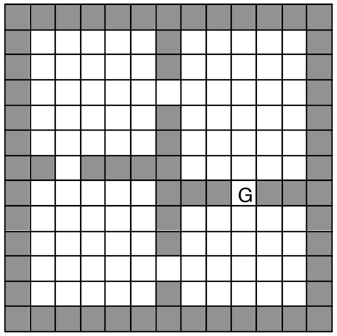
$$P^a_{ss'} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad \forall s, s' \in S, a \in A(s)$$

$$R^a_{ss'} = \mathrm{E}\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \quad \forall s, s' \in S, a \in A(s)$$

γ: discount rate



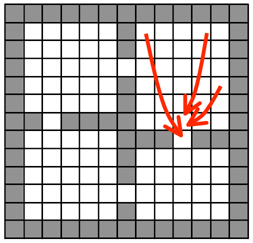
Example



- Actions
  - North, East, South, West
  - Fail 33% of the time
- Reward
  - +1 for transitions into G
  - 0 otherwise
- γ = 0.9

Options

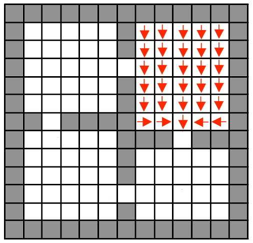
- A generalization of actions
- Starting from a finite MDP, specify a way of choosing actions until termination
- Example: go-to-hallway



Markov options

A Markov option can be represented as a triple $o = < I, \pi, \beta >$

- $I \subseteq S$ is the set of states in which o may be started
- $\pi : S \times A \rightarrow [0,1]$ is the policy followed during o
- $\beta : S \rightarrow [0,1]$ is the probability of terminating in each state



Examples

- `Dock-into-charger`
  - I : all states in which charger is in sight
  - π : pre-defined controller
  - β : terminate when docked or charger not visible
- `Open-the-door`
  - I : all states in which a closed door is within reach
  - π : pre-defined controller for reaching, grasping, and turning the door knob
  - β : terminate when the door is open

One-Step options

A primitive action $a \in \cup_{s \in S} A_s$ of the base MDP is also an option, called a one-step option.

- $I = \{s : a \in A_s\}$
- $\pi(s,a) = 1, \forall s \in I$
- $\beta(s) = 1, \forall s \in S$

Markov vs. Semi-Markov options

- Markov option: policy and termination condition depend only on the current state
- Semi-Markov option: policy and termination condition may depend on the entire history of states, actions, and rewards since the initiation of the option
  - Options that terminate after a pre-specified number of time steps
  - Options that execute other options

Semi-Markov Options

Let H be the set of possible histories (segments of experience)

$$\langle s_t, a_t, r_{t+1}, s_{t+1}, \ldots, s_T \rangle$$

A semi-Markov option may be represented as a triple $o =< I,\pi,\beta >$

- $I \subseteq S$ is the set of states in which o may be started
- $\pi : H \times A \to [0,1]$ is the policy followed during o
- $\beta : H \to [0,1]$ is the probability of terminating in each state

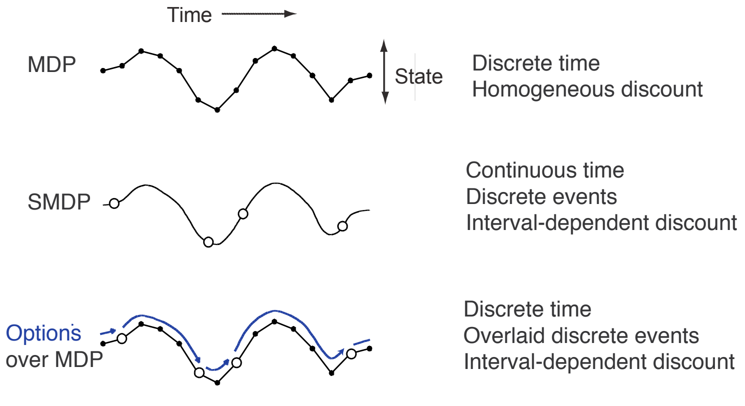
Value functions for options

$$Q^{\mu}(s,o) \overset{\text{def}}{=} E\{r_{t+1} + \gamma r_{t+2} + \ldots \mid o \text{ initiated in } s \text{ at time } t, \; \mu \text{ followed after termination}\}$$

$$Q^{*}_{O}(s,o) \overset{\text{def}}{=} \max_{\mu \in \Pi(O)} Q^{\mu}(s,o)$$

Set of all policies selecting only from options in O

Options define a Semi-Markov Decision Process (SMDP)



A discrete-time SMDP overlaid on an MDP
Can be analyzed at either level

SMDPs

- The amount of time between one decision and the next is a random variable τ
- Transition probabilities

$$P(s',\tau \mid s,a)$$

- Bellman equations

$$V^*(s) = \max_{o\in A_s}\left[R(s,a) + \sum_{s',\tau}\gamma^\tau P(s',\tau \mid s,a)V^*(s')]\right]$$

$$Q^*(s,a) = R(s,a) + \sum_{s',\tau}\gamma^\tau P(s',\tau \mid s,a)\max_{o'\in A_{s'}} Q^*(s',a')$$

Option models

$$R_s^o = E\{r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{\tau-1} r_{t+\tau} \mid$$
$$o \text{ is initiated in state } s \text{ at time } t \text{ and lasts } \tau \text{ steps}\}$$

$$P_{ss'}^o = \sum_{\tau=1}^{\infty}\gamma^\tau \underline{p(s',\tau)}$$

Probability that o terminates in s' after τ steps when initiated in state s

They generalize the reward and transition probabilities of an MDP in such a way that one can write a generalized form of the Bellman optimality equations.

Bellman optimality equations

$$V_O^*(s) = \max_{o\in O_s}\left[R(s,o) + \sum_{s'}P(s' \mid s,o)V_O^*(s')]\right]$$

$$Q_O^*(s,o) = R(s,o) + \sum_{s'}P(s' \mid s,o)\max_{o'\in O_{s'}} Q_O^*(s',o')$$

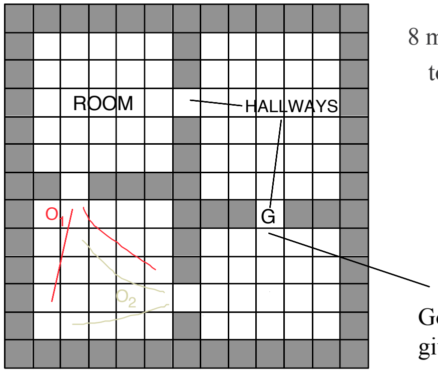
Bellman optimality equations can be solved, exactly or approximately, using methods that generalize the usual DP and RL algorithms.

DP backups

$$V_{k+1}(s) = \max_{o\in O_s}\left[R(s,o) + \sum_{s'}P(s' \mid s,o)V_k(s')]\right]$$

$$Q_{k+1}(s,o) = R(s,o) + \sum_{s'}P(s' \mid s,o)\max_{o'\in O_{s'}} Q_k(s',o')$$

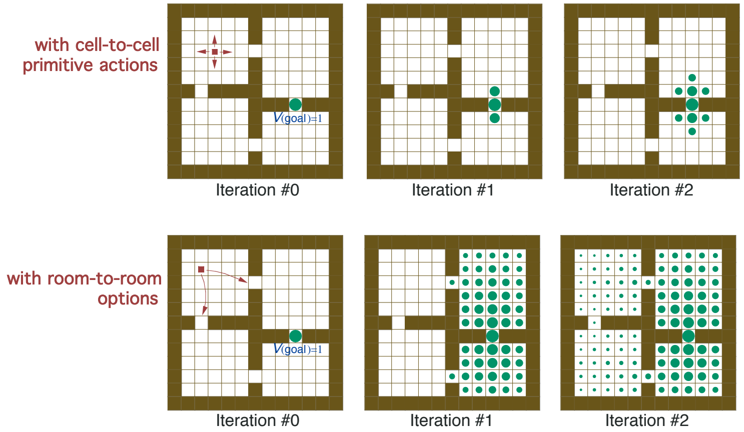
Illustration: Rooms Example



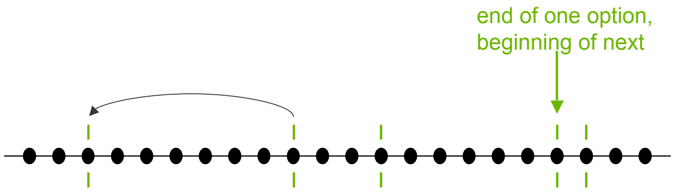
8 multi-step options, to each room's 2 hallways

Goal states are given a terminal value of 1

Synchronous value iteration



SMDP Q-learning backups



- At state *s*, initiate option *o* and execute until termination
- Observe termination state s´, number of steps τ, discounted return r

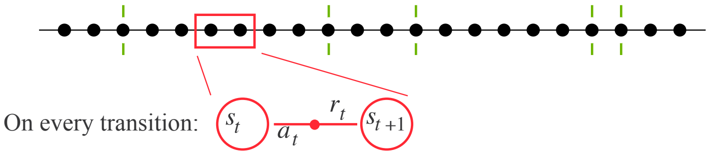
$$Q_{k+1}(s,o) = (1-\alpha_k)Q_k(s,o) + \alpha_k\left[r + \gamma^t \max_{o'\in O_{s'}} Q_k(s',o')\right]$$

Looking inside options

SMDP methods apply to options, but only when they are treated as opaque indivisible units. Once an option has been selected, such methods require that its policy be followed until the option terminates. More interesting and potentially more powerful methods are possible by looking inside options and by altering their internal structure.

—Precup (2000)

Intra-option Q-learning



On every transition:

Update every Markov option o whose policy could have selected a_t according to the same distribution $\pi(s_t, \cdot)$:

$$Q_{k+1}(s_t,o) = (1-\alpha_k)Q_k(s_t,o) + \alpha_k\left[r_{t+1} + \gamma U_k(s_{t+1},o)\right],$$

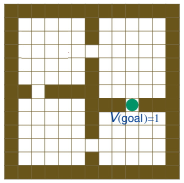
where

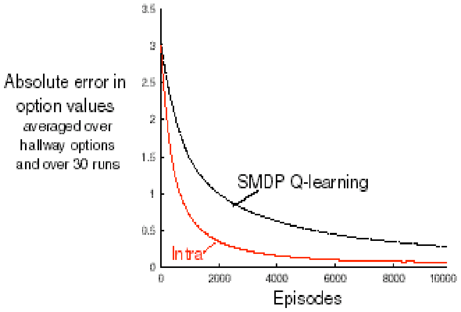
$$U_k(s,o) = (1-\beta(s))Q_k(s,o) + \beta(s)\max_{o'\in O} Q_k(s,o')$$

is an estimate of the value of state-option pair (s,o) upon arrival in state s.

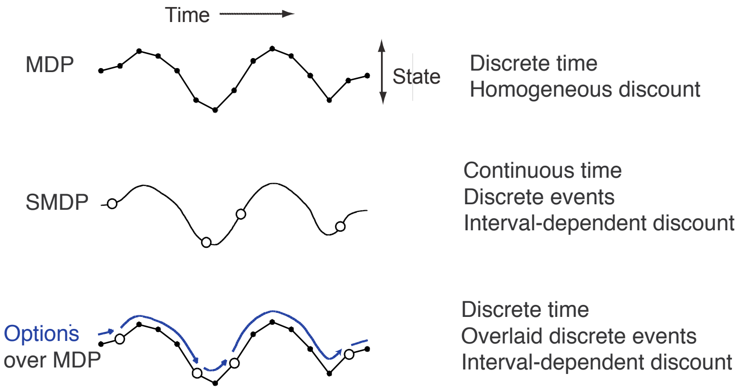
Illustration: Intra-option Q-learning

Random start, goal in right hallway, choice from actions and options, 90% greedy





Summary



Time

MDP

State

Discrete time
Homogeneous discount

SMDP

Continuous time
Discrete events
Interval-dependent discount

Options over MDP

Discrete time
Overlaid discrete events
Interval-dependent discount

A discrete-time SMDP overlaid on an MDP
Can be analyzed at either level

What else?

- Intra-option learning of option models
- Early termination of options
- Improving option policies (given its reward function)
- Learning option policies given useful subgoals to reach (e.g. hallways in the sample problem)

Which states are useful subgoals?

States that …

- have a high reward gradient or are visited frequently
(Digney 1998)
- are visited frequently only on successful trajectories
(McGovern & Barto 2001)
- change the value of certain variables
(Hengst 2002; Barto et al. 2004; Jonsson & Barto 2005)
- lie between densely connected regions
(Menache et al. 2002; Mannor et al. 2004; Simsek & Barto 2004; Simsek, Wolfe & Barto 2005)

References

- D. Precup. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2000.
- R. S. Sutton, D. Precup, and S. P. Singh. Between MDPs and Semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.
- A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341 – 379, October 2003.