The Utility of Temporal Abstraction in Reinforcement Learning

Nicholas K. Jong    Todd Hester    Peter Stone

Department of Computer Sciences
The University of Texas at Austin

The Seventh International Conference on Autonomous Agents and Multiagent Systems
Outline

1. Motivation: Hierarchical Reinforcement Learning

2. Experimental Results
   - Learning with Options
   - Options and Random Exploration
   - Other Applications of Options
Goal: Learn Agent Behaviors Autonomously

Reinforcement learning algorithms:
- Given experience with an unknown environment
- Estimates the value of states
- Learns a policy

Problem
How to learn more efficiently?
Goal: Learn Agent Behaviors Autonomously

Reinforcement learning algorithms:
- Given experience with an unknown environment
- Estimates the value of states
- Learns a policy

Problem
How to learn more efficiently?
Goal: Learn Agent Behaviors Autonomously

Reinforcement learning algorithms:
- Given experience with an unknown environment
- Estimates the value of states
- Learns a policy

Problem
How to learn more efficiently?
Goal: Learn Agent Behaviors Autonomously

Reinforcement learning algorithms:
- Given experience with an unknown environment
- Estimates the value of states
- Learns a policy

Problem
How to learn more efficiently?

Nicholas K. Jong, Todd Hester, Peter Stone
The Utility of Temporal Abstraction in Reinforcement Learning
Intuition: Decompose Tasks into Subtasks

- Standard RL assumes flat state and action spaces.
- Real-world applications have hierarchical structure.
  - Abstract actions
    - Represent sequences of primitive actions
    - Achieve subgoals
Intuition: Decompose Tasks into Subtasks

- Standard RL assumes flat state and action spaces.
- Real-world applications have hierarchical structure.
- Abstract actions
  - Represent sequences of primitive actions
  - Achieve subgoals
**Options**: analogous to macro-operators

- **Initiation set** (precondition)
- **Termination function** (postcondition)
- **Option policy** (implementation)

- Typically used to **augment** an action space
- Can be treated simply as **temporally extended actions**
The Most Popular Framework for Hierarchical RL

- **Options**: analogous to macro-operators
  - Initiation set (precondition)
  - Termination function (postcondition)
  - Option policy (implementation)

- Typically used to **augment** an action space
- Can be treated simply as **temporally extended actions**
The Most Popular Framework for Hierarchical RL

- **Options**: analogous to macro-operators
  - **Initiation set** (precondition)
  - **Termination function** (postcondition)
  - **Option policy** (implementation)

- Typically used to **augment** an action space
- Can be treated simply as **temporally extended actions**
**Motivation**

**Experimental Results**

**Summary**

The Most Popular Framework for Hierarchical RL

- **Options**: analogous to macro-operators
  - Initiation set (precondition)
  - Termination function (postcondition)
  - Option policy (implementation)

- Typically used to **augment** an action space
- Can be treated simply as **temporally extended actions**
The Benefits of Options

- Prior work: options are good
- Future work: where do the options come from?

Key Question
How precisely does the addition of options affect learning?
Outline

1. Motivation: Hierarchical Reinforcement Learning

2. Experimental Results
   - Learning with Options
     - Options and Random Exploration
     - Other Applications of Options
Replicating Results in Option Discovery

- Apply standard Q-learning with $\epsilon$-greedy exploration
- Introduce options after 20 episodes
  - One option for each of four given subgoals
  - Option policies learned from experience replay
  - Initiation set: states that can reach subgoal
Replicating Results in Option Discovery

- Apply standard Q-learning with $\epsilon$-greedy exploration
- Introduce options after 20 episodes
  - One option for each of four given subgoals
  - Option policies learned from experience replay
  - Initiation set: states that can reach subgoal

One of four options
The technique used to obtain the option policy can also be used to improve the value function without using options at all!

- Better baseline: just experience replay after 20 episodes
Options Can Degrade Learning Performance

- Isolating the effect of hierarchy
  - Give only subgoals (at start)
  - Learn option policies online
- Subgoals can **degrade performance** initially.
- Correct options can **severely degrade performance**!

![Graph showing degradation of performance](image-url)
Options Can Degrade Learning Performance

- Isolating the effect of hierarchy
  - Give only subgoals (at start)
  - Learn option policies online
- Subgoals can degrade performance initially.
- Correct options can severely degrade performance!

![Graph showing the comparison between Q-learning, subgoals from start, and options from start in terms of steps per episode over episodes.](image)
Outline

1. Motivation: Hierarchical Reinforcement Learning

2. Experimental Results
   - Learning with Options
   - Options and Random Exploration
   - Other Applications of Options
Options Change the Environment Structure

Random walk in original environment

Random walk in augmented environment
Restricting the Initiation Set

- Idea: Limit options to certain states
- Requires domain expertise

Initiation set of one option
Restricting the Initiation Set

- Idea: Limit options to certain states
- Requires domain expertise

Initiation set of one option

Nicholas K. Jong, Todd Hester, Peter Stone
The Utility of Temporal Abstraction in Reinforcement Learning
Restricting the Initiation Set

- Idea: Limit options to certain states
- Requires domain expertise

![Initiation set of one option](image)

![Graph showing steps per episode vs episodes for Q-learning and options](image)
Delaying Option Deployment

- Idea: wait until value function partially learned
- Somewhat brittle

---

Value function on option deployment

Nicholas K. Jong, Todd Hester, Peter Stone

The Utility of Temporal Abstraction in Reinforcement Learning
Delaying Option Deployment

- Idea: wait until value function partially learned
- Somewhat brittle

Value function on option deployment

Nicholas K. Jong, Todd Hester, Peter Stone
Delivering Option Deployment

- Idea: wait until value function partially learned
- Somewhat brittle

![Value function on option deployment](image)

<table>
<thead>
<tr>
<th>Episodes</th>
<th>Steps per episode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Nicholas K. Jong, Todd Hester, Peter Stone

The Utility of Temporal Abstraction in Reinforcement Learning
1 Motivation: Hierarchical Reinforcement Learning

2 Experimental Results
   - Learning with Options
   - Options and Random Exploration
   - Other Applications of Options
We can blame some of the performance degradation on random exploration.

- Alternative: optimism in the face of uncertainty
- Optimism offers solid theoretical benefits.
- Heuristic implementation: optimistic initialization of the value function

Thorough exploration eliminates the impact of options!
Observation

We can blame some of the performance degradation on random exploration.

- Alternative: optimism in the face of uncertainty
- Optimism offers solid theoretical benefits.
- Heuristic implementation: optimistic initialization of the value function

Thorough exploration eliminates the impact of options!
Options that Abstract Instead of Augment

- Remove primitive actions superceded by options.

Initiation set of one option

Availability of primitive actions

Nicholas K. Jong, Todd Hester, Peter Stone

The Utility of Temporal Abstraction in Reinforcement Learning
Observation

Q-learning may not be the best baseline algorithm for studying hierarchy.

- Q-learning uses each piece of experience exactly once.
- It therefore confounds data acquisition (exploration) with computation (planning).

See also

In ICML 2008: Jong and Stone, “Hierarchical Model-Based Reinforcement Learning: R-MAX + MAXQ”
Options do not always help reinforcement learning; in some cases, they *can severely hinder learning*.

Hierarchical methods impact learning by *biasing or constraining exploration*. 