Discovering Portable Options through Automated Mapping

University of Maryland, Baltimore County, Computer Science and Electrical Engineering Department
Nicholay Topin, Nicholas Haltmeyer, *Shawn Squire, John Winder, James MacGlashan*[1]
[1]Department of Computer Science, Brown University
Dr. Marie desJardins, Computer Science and Electrical Engineering

Overview

Goal for artificial agents: Learn the most efficient process for completing a task in a given domain

- Corollary: Reuse and transfer learned knowledge
- Previous work assumed that a mapping was provided or that all domains were identical
- Our contributions:
  - Automatically map across domains with different objects and attributes
  - Leverage prior knowledge by identifying commonalities between source and target domains
  - Provide novel techniques for scoring mappings and abstracting domains
- Our method outperforms Pickett and Barto's PolicyBlocks (2002) and MacGlashan's Transfer Options (2013)

Background

- A task can be solved more efficiently if previous knowledge is reused, such as by using *options*, sequences of actions found to be used often; options provide several benefits:
  - Can be transferred if there is a known relation between the new task and the one where the knowledge was gained
  - Are extracted from a *policy* (mapping from states to actions that an agent uses to determine which action to take, either learned or created by hand)
- The *domain* (environment) in which a task is performed is modelled as an *Object-Oriented Markov Decision Process* (OO-MDP): each global state is identified by the union of all of the states of all of the objects present
- This allows generalization between states that can be treated as identical and facilitates learning in complex environments
- Existing methods discover options within a single domain (PolicyBlocks (PB)) or discover options for new domains if the domain relationship is known (Transfer Options (TOPs))

Approach

- Ability to automatically create a mapping improves transfer
- OO-MDP representation of a domain allows the identification of the least relevant objects (least important to preserving the behavior of a policy)
- This involves two core procedures: *abstraction* and *scoring*:
  - Abstraction converts a policy for one domain to a policy for another—objects are added or removed to accomplish this
  - Scoring is used to determine the order in which to remove or add objects (leads to the closest approximation to the original possible after each step)
- This framework, Portable Option Discovery (POD), for finding a mapping can be applied to existing methods, including PB and TOPs

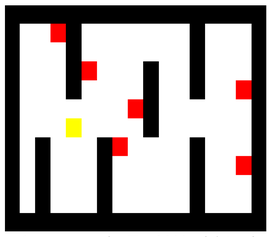
Applications

Taxi World

- Grid world with passengers, walls, goal cell, and agent
- The goal is to drop off every passenger onto the goal
- The agent can move north, south, east, or west
- Can also pick up a passenger if in the same cell as one and drop one off if one is currently being carried

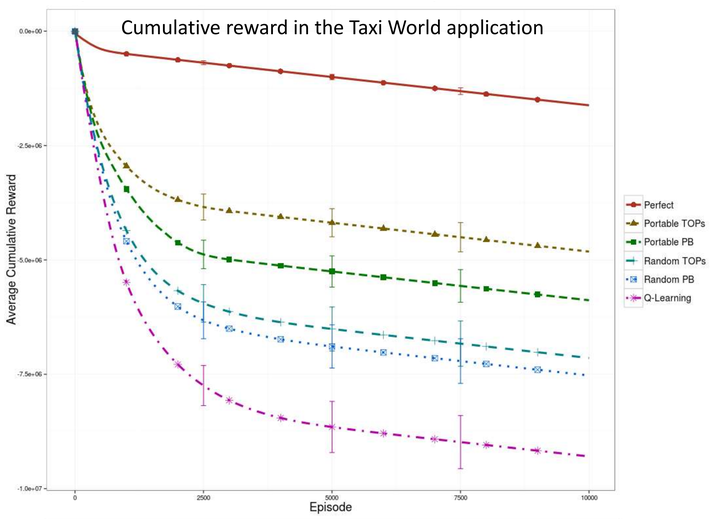
Block Dude

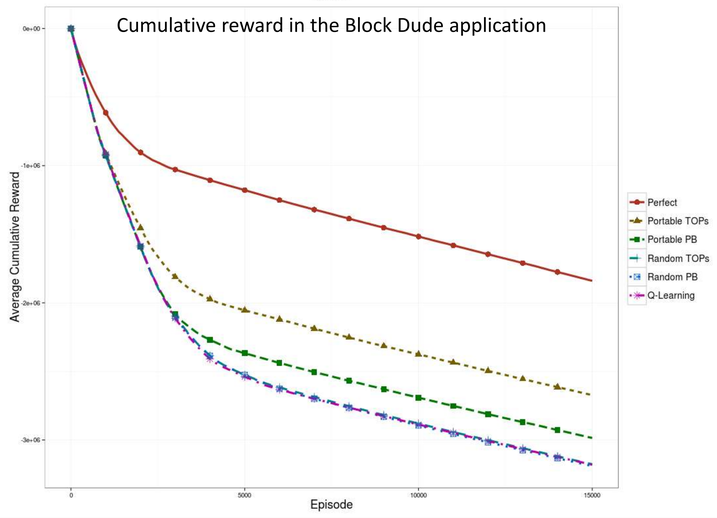
- Based on the game of the same name
- Grid world with exit cell, floors of various heights, movable blocks, and agent—everything falls downwards
- The goal is to reach the exit cell
- The agent can move left or right or climb up a single step
- An adjacent block can be picked up if nothing is on top
- Carried blocks can be placed next to agent if cell is open



An example Taxi World task

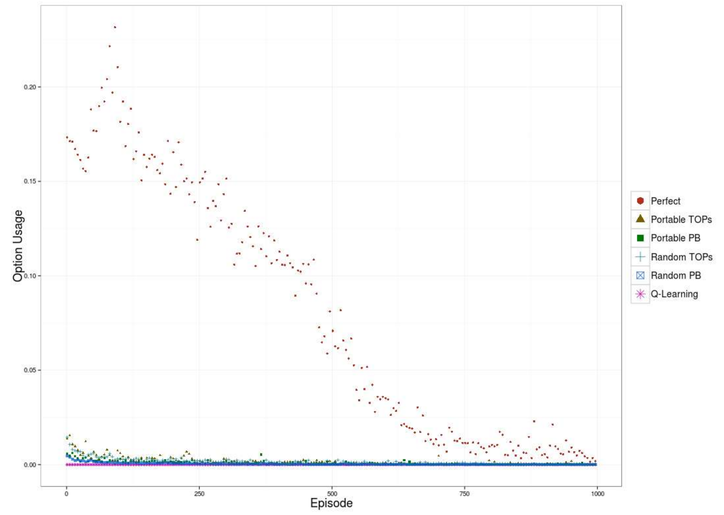
An example Block Dude task





Results

- In Taxi World, transfer is beneficial because the agent is able to reuse knowledge about static elements in the application domain (locations of walls and the goal)
- Performance in Block Dude benefits because the agent is able to reuse knowledge of overcoming obstacles in the application domain regardless of the object locations
- Both PTOPs and PPB perform better than the random mappings
- In Block Dude, random mappings perform only slightly better than Q-learning, since many of the possible mappings do not facilitate meaningful reuse of the options.



Option usage percentage in the Taxi World application.

Conclusion

- Introduced the POD framework to perform automated option discovery in OO-MDP domains
- Extended two existing algorithms to create two new methods (PPB and PTOPs)
- Demonstrated that POD's heuristic mapping selection permits these methods to be applied automatically in object-oriented domains, significantly outperforming standard Q-learning and random mapping selection

Future Work

- Extending beyond tabular learning would be a major improvement; a table of state-action pairs mapped to rewards is a significant limitation for domains that are extremely large or continuous
- Value function approximation (VFA) can be used, allowing the agent to estimate a state's value from similar states
- Extending POD in this way would allow algorithms using our framework to operate in more complex applications

This work was supported by NSF's Division of Information and Intelligent Systems on awards #1065228 and #1340150 (REU supplement). We would like to thank the other co-PIs on the project, Michael Littman and Smaranda Muresan, for many fruitful discussions.