

Introduction

- ▶ How do we perform such a diverse set of complex tasks?
 - ▷ Given an MDP with options, $\mathcal{M}\langle \mathcal{S}, \mathcal{O}, \mathcal{P}, \cdot \rangle$, can we quickly learn any task (i.e. different \mathcal{R})?
- ▶ Most literature focuses on finding options to reach 'bottlenecks', which are common subgoals across tasks. The objective of these options is to aid in early exploration.
 - ▷ A. McGovern and A. G. Barto, "Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density," in ICML, 2001
 - ▷ I. Menache, S. Mannor, and N. Shimkin, "Q-Cut - Dynamic Discovery of Sub-Goals in Reinforcement Learning," in ECML, 2002.
 - ▷ Ö. Şimşek and A. G. Barto, "Skill characterization based on betweenness," in NIPS, 2008
- ▶ Our1 Hypothesis: The key is in finding a set of composable subtasks spanning the space of tasks.

Motivation: The Small World Phenomenon

- ▶ Kleinberg: "Individuals using local information are collectively very effective at actually constructing short paths between two points in a social network."

J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective" in ACM Theory of Computing, 2000
- ▶ Kleinberg constructed a family of networks for which the expected time to deliver a message from any source to any destination was $(\log |\text{size of network}|)^2$, using the inverse power law distribution.
 - ▷ Structural properties of the network are important
- ▶ Can we do the same for learning?
 - ▷ A **small-world RL domain** has the property that an agent using local information (e.g. the value function) can effectively reach a state of *maximal value*.

Generating Options according to P_r

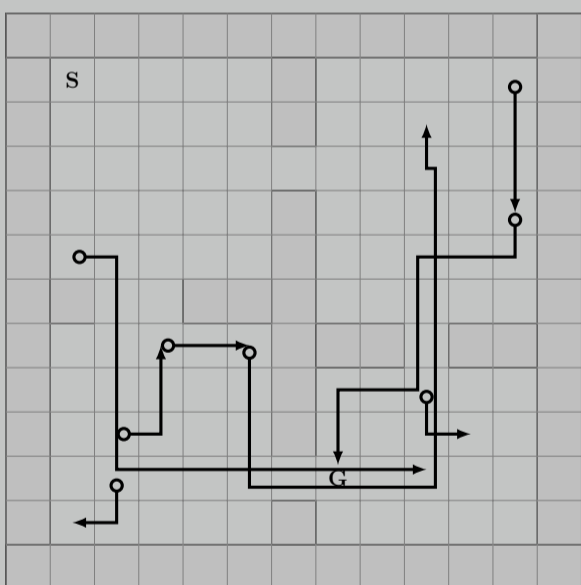


Figure: Some P_2 Options

Note: This construction adds just one additional action for each state, and thus does not blow up the agent's search space.

- ▶ Consider the state-interaction graph of \mathcal{M} .
- ▶ For each state $s \in \mathcal{S}$, select a single s' reachable from s according to the inverse power law distribution

$$P_r : p(s, s') \propto \|s - s'\|^{-r}.$$
- ▶ For each (s, s') pair, construct an option $o : \langle \mathcal{I}, \pi, \beta \rangle$ with $\mathcal{I} = \{s\}$, $\beta = \{s'\}$, and $\pi = \text{optimal policy to reach } s'$.

Theorem: $O((\log n)^2)$ Decisions

Assume \mathcal{M} to have states arranged in a k -dimensional lattice, with noisy (with parameter ϵ) primitive navigation actions \mathcal{A} , and rewards distributed between $[0, 1]$.

Using only the value of neighboring states, an agent with options \mathcal{O} generated by P_k , can reach a state of maximal value in $O((\log |\mathcal{S}|)^2)$ decisions.

- ▶ We relate the value of two states u and v , and their lattice distance,

$$\log \frac{V(v)}{V(u)} \approx \log \left(\sqrt{\frac{1-\epsilon}{\epsilon}} \right) \|u - v\| + c,$$

where $c \in [0, \frac{1}{1-\gamma}]$.

- ▶ Following Kleinberg's analysis, we show that using the optimal value function, the agent makes $O(\log |\mathcal{S}|)$ decisions to get exponentially closer to the maximal value state.

Results

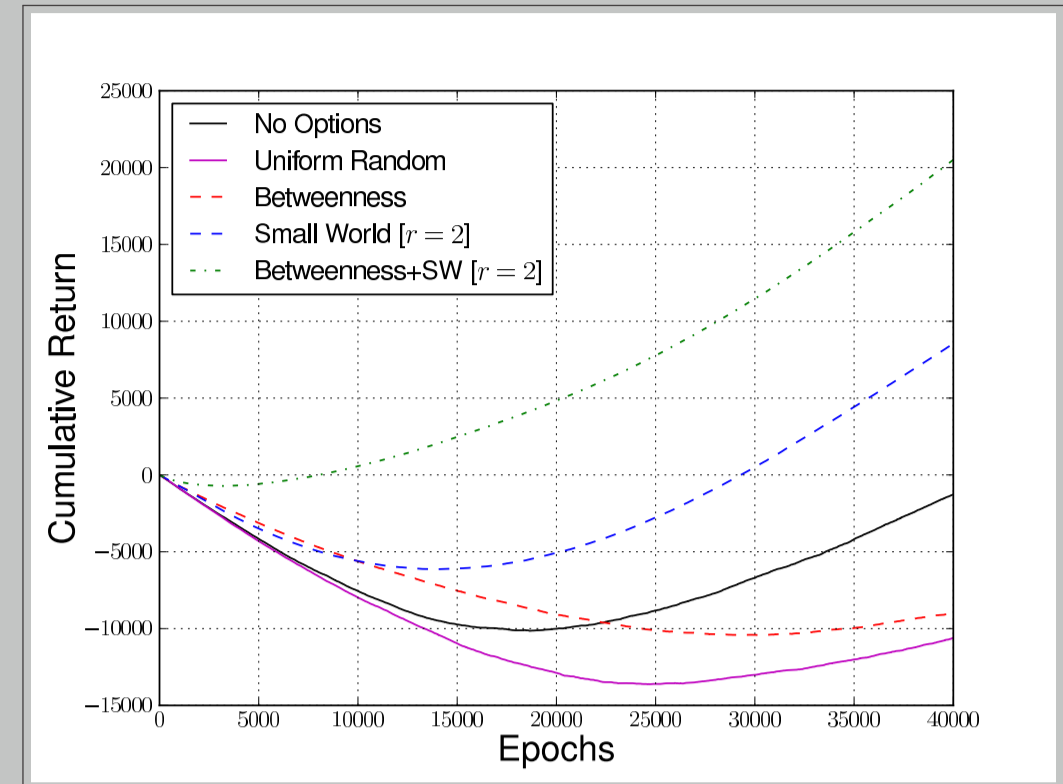


Figure: Rooms: Cumulative Return (with 200 options)

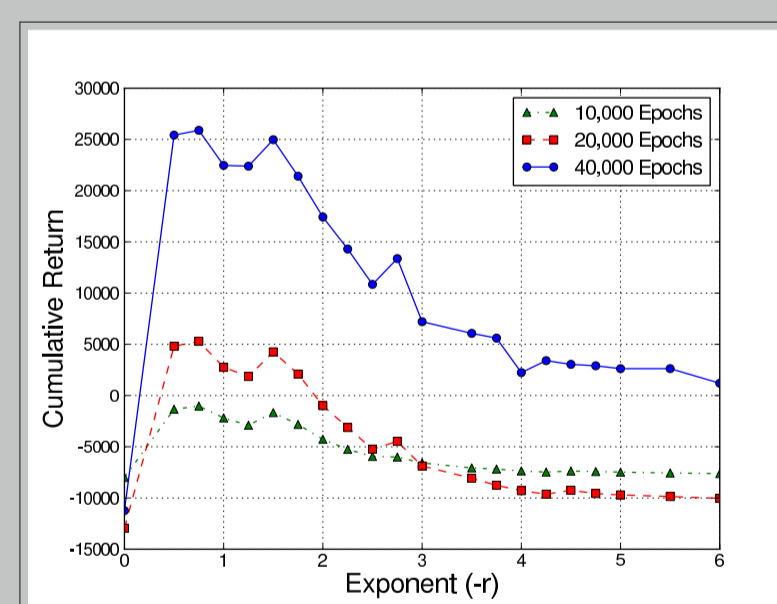
Scheme	Options (40,000 epochs)		Scheme	Options (40,000 epochs)	
	200	400		100	200
None	-31.82	-31.82	None	-16.90	-16.90
P_0	-31.23	-32.90	P_0	-17.68	-18.83
Betw.	-18.28	-24.38	Betw.	80.59	80.48
P_4	-14.24	-7.55	$P_{0.75}$	-7.55	0.66

Table: Arb. Nav.: Cumulative Return

Table: Taxi: Cumulative Return

- ▶ Experiments were run for 40,000 epochs using MacroQ. We compared options generated using a bottleneck based method (betweenness), randomly distributed options (P_0), and small world options ($P_{r>0}$).
- ▶ Bottleneck-based methods have a natural advantage in the Taxi domain, as goal states coincide with bottleneck states (the pick-up and put-down actions).
- ▶ Small-world options do very well on free-navigation tasks (Rooms or Arbitrary Navigation), even in the presence of bottlenecks (Rooms). Combining bottleneck-based options and small world options can outperform both (Rooms).

Role of the Exponent r (Rooms)



- ▶ The basic structure of the the Rooms state spaces is 2D, yet exponents around 1 perform optimally. This difference is likely due to obstacles (walls).
- ▶ The existence of a maximal value for the exponent, as well the behaviour for exponents greater than it, matches what is seen in the social networks scenario.

Conclusions

- ▶ We give an algorithm to generate a random collection of options \mathcal{O} such that any "task" in an MDP can be performed in $O((\log |\mathcal{S}|)^2)$ decisions.
- ▶ We find that these options significantly outperform bottleneck-based options and purely random options.

Future Work

- ▶ By using 'cheaply' learnt policies could the total training time for small world options compare to say that for the case of betweenness?
- ▶ Given the loose conditions for the theorem to hold, could function approximators be used in place of the complete MDP?
- ▶ Could the bounded number of decisions required translate to any theoretical guarantees on faster convergence?