Learning Representations in Model-Free Hierarchical Reinforcement Learning

Jacob Rafati[*] and **David C. Noelle**
Electrical Engineering and Computer Science
Computational Cognitive Neuroscience Laboratory
Univeristy of California, Merced. 5200 North Lake Road, Merced, CA 95343. USA.

Abstract

Common approaches to Reinforcement Learning (RL) are seriously challenged by large-scale applications involving huge state spaces and sparse delayed reward feedback. Hierarchical Reinforcement Learning (HRL) methods attempt to address this scalability issue by learning action selection policies at multiple levels of temporal abstraction. Abstraction can be had by identifying a relatively small set of states that are likely to be useful as subgoals, in concert with the learning of corresponding skill policies to achieve those subgoals. Many approaches to *subgoal discovery* in HRL depend on the analysis of a model of the environment, but the need to learn such a model introduces its own problems of scale. Once subgoals are identified, skills may be learned through *intrinsic motivation*, introducing an internal reward signal marking subgoal attainment. We present a novel *model-free* method for subgoal discovery using incremental unsupervised learning over a small memory of the most recent experiences of the agent. When combined with an intrinsic motivation learning mechanism, this method learns subgoals and skills together, based on experiences in the environment. Thus, we offer an original approach to HRL that does not require the acquisition of a model of the environment, suitable for large-scale applications. We demonstrate the efficiency of our method on a variant of the *rooms* environment.

1 Background

The Reinforcement Learning (RL) problem involves learning behaviors through interaction with an *environment* (Sutton and Barto 2017). At any given time step, the *agent* receives a representation of the environment's *state*, $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible states, and, on that basis, the agent selects an *action*, $a \in \mathcal{A}$, where $\mathcal{A}$ is the set of all available actions. One time step later, as a consequence of the agent's action, the agent receives information from the environment, consisting of a *reward*, $r \in \mathbb{R}$, and also the resulting new state of the agent. Each cycle of interaction is called a transition *experience*, $e = (s, a, r, s')$. At each time step, the agent implements a mapping from states to possible actions, $\pi : \mathcal{S} \to \mathcal{A}$, called its *policy*. The goal of the RL agent is to find an *optimal policy* that maximizes the expected value of the *return*, G (e.g., the cumulative sum of future rewards). Temporal Difference (TD) learning is a class of *model-free* RL methods that attempt to learn a policy without learning a model of the environment. It is often useful to define a *value* function, $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, to estimate the expected value of return when taking a given action in a given state and then following the policy π. When the state space is large, it is common to use a function approximator, $Q(s, a; w)$, to estimate the value function, q_π. Artificial neural networks are often used as such function approximators. Q-learning is a TD algorithm that attempts to find the optimal value function, characterized by parameters, w, by minimizing a loss function, $L(w)$, which is defined as the expectation of squared *TD error* over a recent transition *experience memory*, $\mathcal{D}$:

$$L(w) \triangleq \mathbb{E}_{e \sim \mathcal{D}} \Big[\big(r + \gamma \max_{a'} Q(s', a'; w) - Q(s, a; w) \big)^2 \Big].$$

2 Representations in Model-Free HRL

Problem statement

The reinforcement learning problem suffers from serious scaling issues. *Hierarchical Reinforcement Learning* (HRL) is an important computational approach intended to tackle problems of scale by learning to operate over different levels of *temporal abstraction* (Sutton, Precup, and Singh 1999).

One of the common approaches to temporal abstraction is to identify a set of useful states as *subgoals*. One major open problem in HRL is automatic *subgoal discovery*. Existing methods for subgoal discovery require a model of the environment, such as the state transition probability model and knowledge of the reward function (Şimşek, Wolfe, and Barto 2005). Learning a model of the environment is a difficult problem, however, particularly for large-scale tasks.

In comparison, *model-free* HRL does not require learning a model of the environment. Still, producing accurate value function approximators generally involves learning good internal representations of states. In our previous work, we have studied methods for learning such internal representations during model-free reinforcement learning (Rafati and Noelle 2015; 2017). We now seek to address major open problems in the integration of internal representation learning, automatic subgoal discovery, and intrinsic motivation learning, all within the model-free HRL framework.

[*] Phone: +1 (415) 964-8086, URL: http://rafati.net/

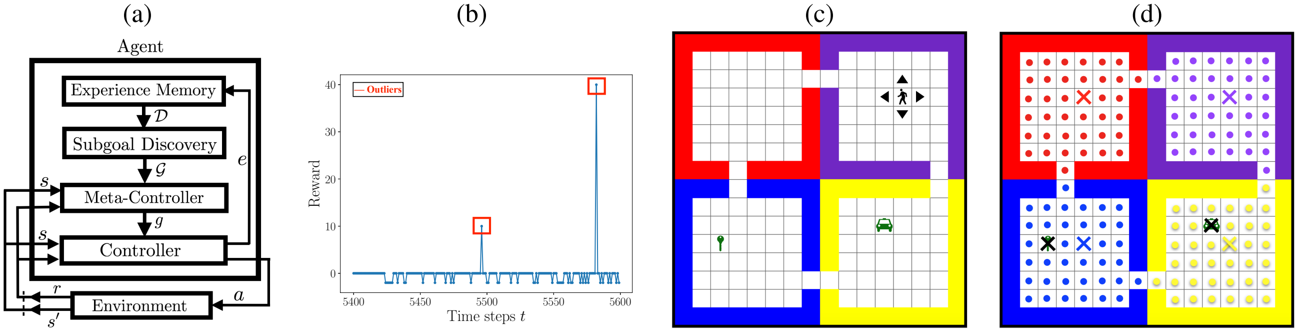

Figure 1: (a) Information flow in the unified model-free HRL framework. (b) Reward over an episode, with anomalous points coresponding to the key ($r = +10$) and the car ($r = +40$). (c) The *rooms* task with a key and a car. (d) The results of the unsupervised subgoal discovery algorithm, with *anomalies* marked in with black Xs and *centroids* with colored ones.

General Method

Inspired by Kulkarni et al. (2016), we use two levels of hierarchy for learning the representations for value function approximation.

The *meta-controller* observes the state, s, from the environment and chooses a *subgoal*, g. Subgoal selection is made from a set of discovered subgoals, $\mathcal{G}$, augmented with a set of random states to support exploration. The choice is made based on a meta-value function, $Q(s, g; w)$, with the subgoal exibiting the highest predicted value generally being selected. TD Learning is used to shape the meta-value function approximator parameters, w, based on reward.

The *controller* receives an input tuple (s, g), and it selects actions based on a policy derived from its value function, $\tilde{Q}(s, g, a; \tilde{w})$. TD Learning is used to learn each *subtask* corresponding to a subgoal, with learning driven by intrinsic reward delivered when the specified subgoal is attained, shaping the value function approximator parameters, $\tilde{w}$.

Value function approximators are generally implemented as multi-layer artificial neural networks augmented to encourage the learning of sparse internal representations of states (Rafati and Noelle 2017).

The transition experience (s, g, a, r, s') is stored in the *experience memory*, $\mathcal{D}$. The *subgoal discovery* mechanism exploits the underlying structure in the experience memory using unsupervised anomaly detection, as well as clustering of similar experiences, in order to add candidate subgoals to the discovered set, $\mathcal{G}$. The information flow between the major components of our proposed model-free HRL framework is depicted in Figure 1(a).

3 Experiment: Rooms Task

We evaluated our unsupervised subgoal discovery method, along with intrinsic motivation learning, on a variant of the *rooms* task, shown in Figure 1(c). The agent is rewarded if it navigates in this grid environment to a key object in one of the rooms and then moves to a car object in some other room. The controller learns to navigate from any location s to any other location g through intrinsic motivation learning, following a policy derived from $\tilde{Q}(s, g, a; \tilde{w})$. An iterative supervised anomaly detection algorithm detects experience outliers, largely from the stream of rewards. (See Figure 1(b).) An iterative version of K-means clustering over experiences identifies abstracted states. The *centroids* of clusters form candidate subgoals, with the transitions between abstracted states capturing *doorways*, as shown in Figure 1(d).

4 Contributions and Future Work

We propose and implement a novel model-free method for subgoal discovery using incremental unsupervised learning over a small memory of the most recent experiences of the agent. When combined with an intrinsic motivation learning mechanism, this method learns subgoals and skills together, based on experiences in the environment. Thus, we offer an original approach to HRL that does not require the acquisition of a model of the environment, suitable for large-scale applications. Planned experiments will test our method on large-scale RL problems, such as portions of the difficult Atari 2600 game *Montezuma's Revenge*.

References

Şimşek, O.; Wolfe, A. P.; and Barto, A. G. 2005. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany*.

Kulkarni, T. D., et al. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems, Barcelona, Spain*.

Rafati, J., and Noelle, D. C. 2015. Lateral inhibition overcomes limits of temporal difference learning. In *37th Annual Meeting of the Cognitive Science Society, Pasadena, CA*.

Rafati, J., and Noelle, D. C. 2017. Sparse coding of learned state representations in reinforcement learning. In *Cognitive Computational Neuroscience, New York City*.

Sutton, R. S., and Barto, A. G. 2017. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1):181 – 211.