Action Discovery for Reinforcement Learning

(Extended Abstract)

Bikramjit Banerjee
School of Computing
The University of Southern Mississippi
118 College Dr. # 5106
Hattiesburg, MS 39406, USA
Bikramjit.Banerjee@usm.edu

Landon Kraemer
School of Computing
The University of Southern Mississippi
118 College Dr. # 5106
Hattiesburg, MS 39406, USA
Landon.Kraemer@eagles.usm.edu

Introduction

The design of reinforcement learning solutions to many problems artif cially constrain the action set available to an agent, in order to limit the exploration/sample complexity. While exploring, if an agent can discover new actions that can break through the constraints of its basic/atomic action set, then the quality of the learned decision policy could improve. On the f ipside, considering all possible non-atomic actions might explode the exploration complexity. We present a potential based solution to this dilemma, and empirically evaluate it in grid navigation tasks. In particular, we show that the sample complexity improves signif cantly when basic reinforcement learning is coupled with action discovery. Our approach relies on reducing the number of decision-points, which is particularly suited for multiagent coordination learning, since agents tend to learn more easily with fewer coordination problems (CPs). To demonstrate this we extend action discovery to multi-agent reinforcement learning. We show that Joint Action Learners (JALs) indeed learn coordination policies of higher quality with lower sample complexity when coupled with action discovery, in a multi-agent box-pushing task.

Reinforcement learning (RL) problems are modeled as *Markov Decision Processes* or MDPs [4]. An MDP is given by the tuple $\{S, A, R, T\}$, where S is the set of environmental states that an agent can be in at any given time, A is the set of actions it can choose from at any state, $R : S \times A \mapsto \Re$ is the reward function, i.e., $R(s, a)$ specif es the reward from the environment that the agent gets for executing action $a \in A$ in state $s \in S$; $T : S \times A \times S \mapsto [0, 1]$ is the state transition probability function specifying the probability of the next state in the Markov chain. The agent's goal is to learn a policy $\pi : S \mapsto A$ that maximizes the sum of discounted future rewards from any state s, given by,

$$V^{\pi}(s) = E_T[R(s, \pi(s)) + \gamma R(s', \pi(s')) + \gamma^2 R(s'', \pi(s'')) + \ldots]$$

where $s, s', s'', \ldots$ are samplings from the distribution T following the Markov chain with policy π, and $\gamma \in (0, 1)$ is the discount factor.

A common method is to learn an action-value function, $Q(s, a)$, using an on-policy method called Sarsa, given by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

where $\alpha \in (0, 1]$ is the learning rate, r_{t+1} is the actual environmental reward and $s_{t+1} \sim T(s_t, a_t, .)$ is the actual next state resulting from the agent's choice of action a_t in state s_t.

Cite as: Action Discovery for Reinforcement Learning (Extended Abstract), Bikramjit Banerjee and Landon Kraemer, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1585-1586

Action discovery

In reinforcement learning problems, the atomic action set, A_0, is usually f xed. However, in many cases new actions that are neither included in A_0, nor precluded by the agent's capabilities, may be able to improve the agent's performance by (a) reducing the number of steps to the goal, or the total solution cost, (b) reducing the cost of exploration by connecting topologically distant states with new actions, and (c) making the goal-directed behavior more natural, i.e., less constrained from a design perspective.

We renounce the innate meaning of an action, and assume it to simply stand for a vehicle of state transition. As such, we represent an action by $a_{ss'}$ to mean that the *intended* purpose of this action is to transition from state s to state s'. To accomodate nondeterminism in the effect of an action, we can now redef ne the transition function T as $T(s, a_{ss'}, s'')$ to stand for the probability that if the agent acts with the intention of transitioning from s to s', then it ends up in state s''. Therefore, $T(s, a_{ss'}, s')$ is the probability of success of this action. In this paper, however, we focus on the deterministic cases, i.e., where $T(s, a_{ss'}, s')$ is either 1, or the action $a_{ss'}$ is infeasible due to physical limitations of the agent or the environment, for all s, s'. It is useful to deal with both possibilities uniformly, with a cost function.

We assume that for a given domain, a cost function $c : S \times S \mapsto \Re$, is always available, such that $c(s, s')$ gives the cost of executing an action that would take an agent from state s to state s', i.e., $a_{ss'}$. If $c(s, s') < \infty$, this simply means that there is some action (whether atomic or newly discovered) that takes the agent from state s directly to state s'. However, if $c(s, s') = \infty$, then no such action exists. For actions outside the atomic action set (A_0), and having a f nite cost, we do not assume that a reward sample for such an action is available unless this action is actually executed. Hence the f rst time that such an action is discovered the reward is *estimated* on the basis of the actual rewards r_1, r_2.

Clearly, accepting every newly discovered action into the set of actions will be expensive for learning. For instance, in a grid of size $n \times n$, there may be $O(n^2)$ such new actions, per state, i.e., potentially $O(n^4)$ actions to contend with. Accomodating such a large number of actions will impact the exploration and reduce the learning rate. Fortunately, many of these actions may be needless to explore, e.g., if they lead away from the goal. It is possible to estimate the *value potential* of a state, Φ, precisely for this purpose. Potential functions, $\Phi(s)$, have been used before, to shape rewards and reduce the sample complexity of reinforcement learning [3]. In this paper, we use such functions to informatively select among newly discovered actions.

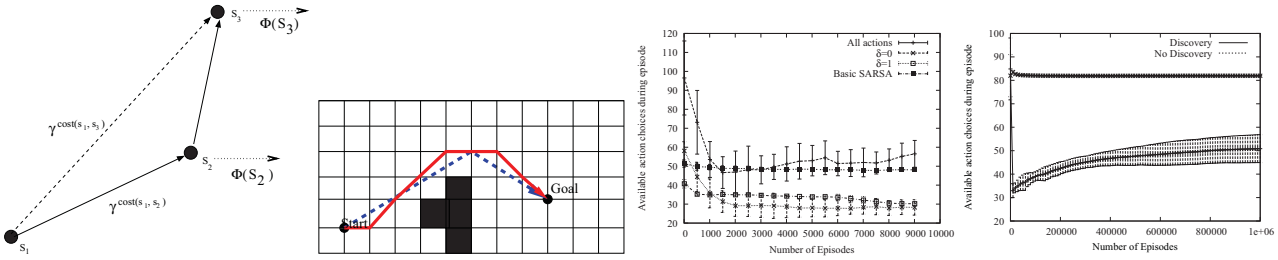

Figure 1: a Illustration of the action discovery criterion. b Navigation map for single agent experiment. Sarsa nds the solid path while Sarsa AD nds the dotted path. c Sample complexities in single agent experiment. d Sample complexities in multi agent box pushing experiment.

To illustrate our heuristic selection procedure for newly discovered actions, consider an agent that has transitioned through successive states s_1, s_2, and s_3, during some episode, t (Figure 1(a)). The actions that it has executed to make these transitions may be atomic actions (i.e., in A_0), or previously discovered new actions, in the current set $A_t(.)$. At state s_3, the agent determines if there exists an action that could have transitioned it directly from s_1 to s_3, i.e., whether $c(s_1, s_3) < \infty$. If this is true and this action did not exist in $A_t(s_1)$, then a new action has been discovered. This action, $a_{s_1 s_3}$, is worth exploring (and hence added to $A_t(s_1)$) if

$$\gamma^{c(s_1,s_3)}\Phi(s_3) > (1+\delta)\gamma^{c(s_1,s_2)}\Phi(s_2)$$

where δ is a slack variable guiding the degree of conservatism in accepting new actions. Note that new actions merely facilitate reaching the goal, but they are not necessary for the agent to reach the goal. We call the above version of Sarsa, Sarsa-AD (Sarsa with Action Discovery).

Usually sample/experience complexity in RL is measured by the number of decisions that the agent has to make in each episode. The problem with this measure in the context of our work is that it is not only affected by learning, but also by action discovery. Clearly, Sarsa-AD will learn to make fewer decisions than Sarsa, by virtue of action discovery. However, Sarsa-AD makes fewer decisions at the expense of increasing the number of choices (i.e., available actions) at each decision point. Therefore, a more ref ned measure of sample complexity for Sarsa-AD would be the sum of the number of choices available across all decision points in each episode. Figure 1(c) shows the above measure of sample complexity (95% conf dence intervals over 20 runs) for Sarsa-AD (with various δ), compared to basic Sarsa, on the navigation task in the map shown in Figure 1(b). We see a statistically signif cant advantage of Sarsa-AD over Sarsa, especially for low δ.

Multi agent Learning with Action Discovery

Our results so far indicate a benef cial impact of action discovery on exploration complexity even though it comes at a cost to decision complexity, so much so that the overall sample complexity is signif cantly lower than in regular reinforcement learning. However, a sterner test for this hypothesis is in a multi-agent system where the decision complexity grows exponentially with the number of agents, creating the possibility that any augmentation of the action set (by discovery) may dominate the sample complexity.

In order to test the hypothesis that action discovery is benef cial to the sample complexity (combined over all agents) in a multi-agent learning (MAL) task, we adopt the Joint Action Learning algorithm [2]. For JALs, the decision complexity is clearly exponential in the number of agents, n, since each agent maintains a Q-value for each joint-state s and the entire joint-action vector $\langle a_1, a_2, \ldots, a_n \rangle$. Since we intend to test the impact of action discovery on what Boutilier calls coordination problems (CPs) [1], in particular whether the number of coordination problems are reduced or increased, we cleanly separate the atomic action sets of agents, so that every decision point is a coordination problem. In our experiments we consider two agents pushing a box on a plane, so we allow one agent to exert a force along the x-axis only (we call it the x-agent), and the other along the y-axis only (the y-agent. By removing overlap in the directionalities of the forces, we ensure that the agents do not trivially coordinate at some decision points. This serves the purpose of isolating the impact of action discovery on CPs, with the impact on accidental coordination being removed. Note however, that this is only meant for our experimental set-up, and it is not necessary to preclude overlaps in the agents' atomic action sets. Also, agents can achieve such clean separation of their action sets by prior agreement in cooperative domains.

We allow each agent to test for feasibility of a new action using the same method as in Sarsa-AD. If a new action passes the test, then all agents discover that action and append their action sets in that joint-state, by the appropriate *component* of the discovered action. Therefore, if an action (x', y') is discovered, the x-agent appends x' as a new action in its own list of actions in that state, and also includes y' as a new action of the other agent in that state. The y-agent performs the corresponding actions as well. This means that with each discovery, the joint action table of each agent grows at the rate of $O(|A|^{n-1})$ where A is the size of the action set of each agent. Given such a phenomenal growth in decision complexity, it is unclear if action discovery will benef t multi-agent learning.

Figure 1(d) shows the total sample complexity (95% conf dence intervals over 20 runs) of JAL Sarsa learning with and without action discovery, when two agents learn to coordinate in pushing a block from one corner to the diagonally opposite corner on a square map with a square obstacle in the center. Surprisingly, action discovery improves the sample complexity in JAL as well. This clearly demonstrates that the impact of action discovery on the number of CPs (which is reduced) outweighs the impact on decision complexity (which is worsened), such that the net sample complexity is signif cantly lower with action discovery. The result reaff rms our f nding that action discovery is indeed a potent tool for reinforcement learner(s) to improve sample complexity of learning, through the counter-intuitive process of worsening the decision complexity.

1. REFERENCES

[1] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 478–485, 1999.

[2] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 746–752, Menlo Park, CA, 1998. AAAI Press/MIT Press.

[3] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proc. 16th International Conf. on Machine Learning*, pages 278–287. Morgan Kaufmann, 1999.

[4] R. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.