Escaping Groundhog Day

Abstract

Existing approaches to reinforcement learning rely on a fixed state-action space and reward function that the agent is trying to maximize. During training, the agent is repeatedly reset to a predefined initial state or set of initial states. For example, in the classic RL Mountain Car domain, the agent starts at some point in the valley, continues until it reaches the top of the valley and then resets to somewhere else in the same valley. Learning in this regime is akin to the learning problem faced by Bill Murray in the 1993 movie *Groundhog Day* in which he repeatedly relives the same day, until he discovers the optimal policy and escapes to the next day. In a more realistic formulation for an RL agent, every day is a new day that may have similarities to the previous day, but the agent never encounters the same state twice. This formulation is a natural fit for robotics problems in which a robot is placed in a room in which it has never previously been, but has seen similar rooms with similar objects in the past. We formalize this problem as learning to act in a *domain*. A domain is defined by the tuple (X, A, P), where X is a state representation, A is an action set, and P is a probability distribution of MDPs with different state spaces, reward functions, and transition dynamics, but the same state representation X and action set A. The agent samples an MDP from some unknown distribution of related MDPs and must learn to behave well under the distribution of MDPs. The agent observes samples from the MDP distribution and at test time is evaluated on a new set of MDP samples drawn from the same distribution, focusing the evaluation on the agent's ability to generalize.

Keywords: Meta-learning, transfer learning, learning to plan,

1 Introduction

Existing approaches to reinforcement learning rely on a fixed state-action space and reward function that the agent is trying to maximize. Often during training, the agent is reset to a predefined initial state or set of initial states. For example, in the classic RL Mountain Car problem [1], the agent is tasked with driving an underpowered car out of a valley and the agent learns to do so over a series of episodes. At the start of an episode, the agent is placed in some random location in the valley and end when the agent drives out of the valley. After the episode ends a new episode begins and learning continues. Learning in this problem, and many other RL problems, is akin to the learning problem faced by Bill Murray in the 1993 movie *Groundhog Day* in which he repeatedly—and unrealistically—relives the same day, until he discovers the optimal policy and escapes to the next day. Instead, a more natural formulation for many problems is that the agent samples an MDP from some unknown distribution of related MDPs and must learn to behave well under the distribution of MDPs. In our analogy, the agent iteratively experiences new days that may have similarities to the previous days, but require acting in states it has not previously observed. Given a new MDP drawn from the same distribution, we would like the agent to learn to solve it as quickly as possible. This formulation is a natural fit for robotics problems in which a robot is placed in a room it has never previously seen, but has seen similar rooms with similar objects in the past.

We formalize this problem as learning to act in a *domain*. A domain is defined by the tuple (X, A, P), where X is a state representation, A is an action set, and P is a probability distribution of MDPs with different state spaces, reward functions, and transition dynamics, but the same state representation X and action set A. In principle, P could have such a large variation that nothing learned in one MDP drawn from it is useful for another; however, we are interested in domains in which there strong commonalities. For example, in the real world, no two rooms may be the same, but physical and social constraints entail similarity in the mechanics of objects, such as light switches and door knobs; we would like are agent to learn in this type of scenario.

There are two types of learning problems that can be addressed in this setting: (1) learning to plan and (2) learning to learn. In learning to plan, the agent always knows the transition dynamics for each MDP; however, the agent can exploit its knowledge from solving previous MDPs from the distribution to more efficiently find good solutions in new MDPs. In learning to learn, the agent does not know the transition dynamics for each MDP, but can use knowledge from learning in previous MDPs to learn a portable model or bias its exploration through the state space.

By formalizing the learning problem in this way, we are able to focus research on generalization to states never previously seen rather than overfitting to a single set of states and reward function. Focusing on generalization suggests a set of evaluation metrics in which at training time the agent learns from i.i.d. samples from the MDP distribution and at test time is evaluated on a new set of i.i.d. MDP samples drawn from the same distribution. In this work, we formally describe our learning problem and evaluaiton metrics for it, present two approaches that address it, and review how existing work fits in with our problem formulation.

2 Problem Definition

- overall problem definition
- RL instantiation (true transition dynamics unknown)
- planning instantiation (true transition dynamics provided)
- evaluation metrics

3 Approaches

3.1 Action Priors

3.2 Meta-RL

4 Related Work

4.1 Model-based RL

4.2 Bayesian RL

4.3 Transfer Learning

4.4 Skill learning

5 Conclusion

References

[1] S.P. Singh and R.S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1):123–158, 1996.