Information Maximizing Exploration and Control

YuXuan Liu [1]

Abstract

In the absence of dense reward signals, reinforcement learning algorithms often struggle to learn. Methods such as count-based exploration or information maximization attempt to alleviate this problem by incentivizing exploration in novel regions of the state space. In this project, we propose to explore an unsupervised method that learns to explore the state space in the absence of a known reward function.

1. Introduction

Deep reinforcement learning algorithms have shown a remarkable ability in learning complex control policies from high-dimensional visual input. However, much of the success of these methods have been in domains where rewards provide dense feedback in terms of which actions were optimal. In sparse-reward tasks, where no reward signal may be given for most states, naive exploration strategies such as ϵ-greedy or randomly initialized policies will struggle to learn.

In an attempt to address the sparse-reward problem, many prior methods use intrinsic motivation to encourage exploration. Houthooft et al. (2016) maximize information gain on a model of the environment dynamics to encourage exploration in regions of the state space the model cannot predict well. Tang et al. (2016) use hashing in a count-based exploration bonus. Bellemare et al. (2016) approximate count-based exploration with an explicit density model, while Fu et al. (2017) extend count-based exploration to implicit density models.

2. Method

Consider a Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0)$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $\mathcal{T}(s_{t+1}|s_t, a_t)$ the transition distribution, and ρ_0 the initial state distribution. We consider finite T horizon trajectories with $s_0 \sim \rho_0$, $a_t \sim \pi_\theta(s_t|a_t)$, $s_t \sim \mathcal{T}(s_{t+1}|s_t, a_t)$ where $\pi_\theta(s|a)$ is a parameterized policy. A trajectory $\tau := \{s_0, a_0, s_1, a_1 ...\}$ is defined as a collection of states and actions sampled according to the trajectory distribution induced by π_θ.

Without a priori knowledge of a reward, an agent should have mastery of the states it can reach. Thus to account for sparse rewards that can be located in any region of the state space, we optimize π_θ to maximize entropy over reachable states at time T:

$$\max_\theta \mathcal{H}(s_T) \tag{1}$$

We optimize θ using reinforcement learning with reward $R(s_T) = -\log p(s_T)$ where $p(s_T)$ is the probability the final state is s_T under the policy θ. We approximate $p(s_T)$ using a density model $p_\psi(s_T)$ with parameters ψ. We use a trust region policy optimization method (Schulman et al., 2015) to bound the policy changes across iterations so the final state distribution $p(s_T)$ does not change significantly across iterations. A maximum entropy policy such as this will serve as a good initialization for sparse reward problems when the reward function is not known beforehand.

While a maximum entropy policy will learn to explore the state space, it may be slow to adapt such a policy to new tasks. It may take several gradient updates to reduce the entropy of the policy and shift the trajectory distribution to regions of the state space relevant to a given task. To address this issue, we introduce a latent variable Z, which controls the state distribution of the policy. Namely, we consider the optimization:

$$\max_{\theta, Z} I(s_T; Z) \tag{2}$$

Note that the objective can decomposed into a sum of the maximum entropy term and an conditional entropy term:

$$I(s_T; Z) = \mathcal{H}(s_T) - \mathcal{H}(s_T|Z) \tag{3}$$

This objective still maintains that the policy should learn to explore the state space uniformly, however given a latent variable Z, the policy should consistently visit a single region of the state space.

[1]University of California, Berkeley. Correspondence to: YuXuan Liu <yuxuanliu@berkeley.edu>.

EE 229A Final Project.

Algorithm 1 Maximum Entropy Exploration

for iterations $= 1$ **to** N **do**
  Sample trajectories $\{\tau_1, \tau_2, ..., \tau_n\}$ using π_θ
  Policy gradient step $\theta \leftarrow \theta + \alpha \nabla \mathbb{E}_\tau[-\log P_\phi(s_T)]$
  Update density model $\phi \leftarrow \phi + \alpha \nabla \mathbb{E}_\tau[\log P_\phi(s_T)]$
end for

3. Experiments

In designing our experiments, we sought to address the following questions:

1. Does maximum entropy exploration learn policies that effectively explore the reachable state space?
2. Can we learn policies that are easily controllable with a latent variable Z?
3. How does the method scale to complex continuous state and action spaces?

3.1. 2D Block Manipulation

To evaluate the ability of our method to discover manipulation skills, we set up a 2D block manipulation environment. The agent, a 2D navigator, can pick up, move, and place different blocks within a bounding box. We used the Maximum Entropy Exploration Algorithm [1] with π_θ parameterized as a neural network and P_ϕ as a count-based estimator over a finite history.

The results of our experiments show that the maximum entropy exploration method was able to move multiple blocks to various places around the box, without an external reward signal that rewarded block moving. The distribution of final block positions is also fairly distributed across the state space as show in Figure 1.

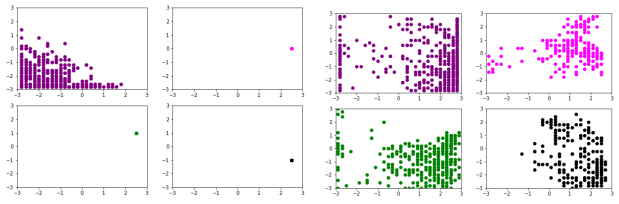

Figure 1. Left four plots: plot of final state distribution of the agent (purple) and three other blocks (magenta, green, black) using a randomly initialized policy. Right four plots: plot of final state distribution of the agent (purple) and three other blocks (magenta, green, black) using the policy trained with Maximum Entropy Exploration. The exploration policy learned to pick up and move blocks in the absence of an external reward.

Algorithm 2 Maximum Controllability

for iterations $= 1$ **to** N **do**
  Sample trajectories $\{\tau_1, \tau_2, ..., \tau_n\}$ using π_θ
  Policy gradient step
    $\theta \leftarrow \theta + \alpha \nabla \mathbb{E}_\tau[-\log P_\phi(s_T) + \log P_\psi(s_T|Z)]$
  Update density model
    $\phi \leftarrow \phi + \alpha \nabla \mathbb{E}_\tau[\log P_\phi(s_T)]$
    $\psi \leftarrow \psi + \alpha \nabla \mathbb{E}_\tau[\log P_\psi(s_T|Z)]$
end for

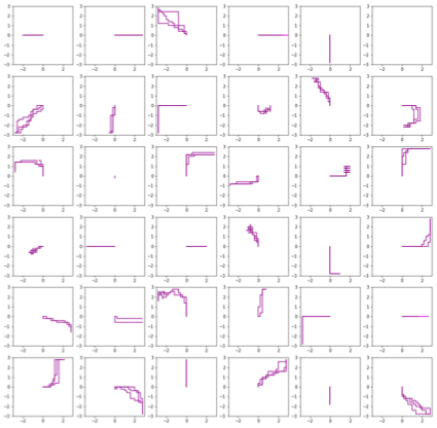

Figure 2. Each subplot shows the agent trajectory samples for given Z. The controllable policy consistently reaches a small region of the state space conditioned on Z.

3.2. Controllable Navigation

To evaluate the Maximum Controllability Algorithm [2], we ran the algorithm on the 2D manipulation environment. We used a fixed discrete, uniform distribution Z, with distributions P_ϕ and P_ψ as finite-history, count-based density estimators. π_θ is parameterized as a neural network. We found that, conditioned on a latent variable Z, the policy consistently visited a small region of the state space as show in Figure 2. The policy also covered a wide range of the state space over all possible Z.

3.3. Exploration Continuous Environments

In evaluating the ability of our method to scale to complex, continuous-control environments, we applied the Maximum Entropy Exploration Algorithm to the swimmer environment. The swimmer is a torque-controlled, 3-link robot with continuous actions and states that navigates on a plane.

To estimate continuous density, we parameterize P_ϕ as a variational autoencoder and π_θ as a neural network. We found that the swimmer learned to swim to a wide distribution of locations on the plane as shown in Figure 3.

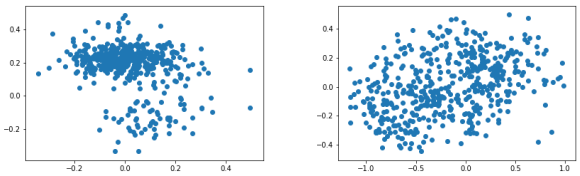

Figure 3. Left: distribution of final center-of-mass (COM) for swimmer with a randomly initialized policy. Right: distribution of final center-of-mass for swimmer a policy trained using Maximum Entropy Exploration. Notice the wider, more evenly spread distribution using the exploration policy.

References

Bellemare, Marc, Srinivasan, Sriram, Ostrovski, Georg, Schaul, Tom, Saxton, David, and Munos, Remi. Unifying count-based exploration and intrinsic motivation. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1471–1479. Curran Associates, Inc., 2016.

Fu, Justin, Co-Reyes, John D., and Levine, Sergey. EX2: exploration with exemplar models for deep reinforcement learning. *CoRR*, abs/1703.01260, 2017. URL http://arxiv.org/abs/1703.01260.

Houthooft, Rein, Chen, Xi, Chen, Xi, Duan, Yan, Schulman, John, De Turck, Filip, and Abbeel, Pieter. Vime: Variational information maximizing exploration. In Lee, D. D., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances In Neural Information Processing Systems 29*, pp. 1109–1117. Curran Associates, Inc., 2016.

Schulman, John, Levine, Sergey, Abbeel, Pieter, Jordan, Michael, and Moritz, Philipp. Trust region policy optimization. In Blei, David and Bach, Francis (eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1889–1897. JMLR Workshop and Conference Proceedings, 2015. URL http://jmlr.org/proceedings/papers/v37/schulman15.pdf.

Tang, Haoran, Houthooft, Rein, Foote, Davis, Stooke, Adam, Chen, Xi, Duan, Yan, Schulman, John, Turck, Filip De, and Abbeel, Pieter. #exploration: A study of count-based exploration for deep reinforcement learning. *CoRR*, abs/1611.04717, 2016. URL http://arxiv.org/abs/1611.04717.