Robot Navigation Policy Evaluation Based on Minimax OPE on POMDP Algorithm

Yi Wei, Yunfu Deng

[†]University of Wisconsin - Madison

Introduction

We apply one of the state-of-the-art off-policy evaluation methods, i.e. Minimax OPE algorithm, to the robot navigation problem in Labyrinth-Escape Environment with the Partially Observable MDP setting, which is an instrumental and important Environment for robot path planning tasks in robotics area. Thus we offer an efficient and accurate policy evaluation tool for robot navigation problems without curse of history.

Problem Motivation and Connection to DP

Navigation and Path Planning is a fundamental problem of mobile robots. And MDP is a powerful mathematical tool for this problem because of its strong representation and experience learning abilities. What's more, by the nature of the navigation problem, robots can only get the information from sensing equipment. Thus a large number of effective structures and algorithms are modeled as Partially Observable MDP.

In this area, since the experimentation is expensive and risky, we need to utilize the historical dataset well. Therefore, after we train a new policy, we need to evaluate it with accessible dataset, which is the aim of the off-policy evaluation (OPE). OPE is concerned with estimating the mean reward of a given decision policy, known as the evaluation policy, using historical data generated by a potentially different policy, known as the behavior policy.

Previously, Off-Policy Evaluation methods like "sequential-importance-sampling" are mainly based on the whole horizon of data. So it will suffer "curse of history", the error would grow exponentially in the whole horizon. To address this issue , a state-of-the-art estimation algorithm, Minmax OPE on POMDP based on Future-Dependent Value Functions [4], has been proposed theoretically. And we hope to apply it in the robotics navigation problem and offer an application strucuture.

Model Setup

Setup: We model the problem as an infinite-horizon discounted POMDP.

We denote the state space by $\mathcal{S}$, the action space by $\mathcal{A}$, the observation space by $\mathcal{O}$, the emission kernel by $\mathbb{O} : \mathcal{S} \to \Delta(\mathcal{O})$ (the conditional distribution of the observation given the state), the state transition kernel by $\mathbb{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, (the conditional distribution of the next state given the current state-action pair), and the reward function: $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Under POMDP setup, exact $\mathbb{T}, \mathbb{O}, r$ are unknown.

Then we denote memory-less policies by $\pi : \mathcal{O} \to \Delta(\mathcal{A})$, the corresponding behavior policy and evaluation policy by π^b and π^e, the reward at stage t by R_t, the discounted factor by $\gamma \in [0, 1)$. Thus the objective of OPE for this setup is to evaluate:

$$J(\pi^e) := \mathbb{E}_{\pi^e}[\sum_{t=0}^{\infty} \gamma^t R_t]$$

Dataset: We denote the M_H-step historical observations and actions obtained prior to some observation O_t at time t by $H = (O_{t-M_H:t-1}, A_{t-M_H:t-1})$, and the M_F-step future observations after some observation O at time t by $F' = (O_{t+1:t+M_F}, A_{t+1:t+M_F-1})$. Additionally $F = (O_{t:t+M_F-1}, A_{t:t+M_F-2})$. Then $\mathcal{F}$ denotes the domain of F: $(\mathcal{O} \times \mathcal{A})^{M_F-1} \times \mathcal{O}$, and $\mathcal{H}$ denotes the domain of H: $(\mathcal{O} \times \mathcal{A})^{M_H}$

We obtain offline data D generated by π^b. And we divide it into 2 datasets: D_{tra} and D_{ini}, where D_{tra} contains $\{(H^{(i)}, O^{(i)}, A^{(i)}, R^{(i)}, F^{(i)})\}_{i=1}^{N}$, and use (H, O, A, R, F') to denote a generic data tuple with respect to some given H and O. And D_{ini} contains $\{O^{(i)}_{0:M_F-1}, A^{(i)}_{0:M_F-2}\}_{i+1}^{N'}$, which is generated starting from a random initial state by some distribution until we observe $O^{(i)}_{M_{F-1}}$ and $A^{(i)}_{M_{F-1}}$.

Theoretical Foundation

Definition 1 (Future-dependent value functions). Future-dependent value functions $g_V \in [\mathcal{F} \to \mathbb{R}]$ are defined such that the following holds almost surely

$$\mathbb{E}[g_V(F)|S_0] = V^{\pi^e}(S_0)$$

where $V^{\pi^e}(S_0) := \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k | S_0]$. The expectation is taken w.r.t the D generated by π^b.

Definition 2 (Learnable future-dependent value functions). Define $u(O, A) := \pi^e(A|O)/\pi^b(A|O)$. Learnable future-dependent value functions $b_V \in [\mathcal{F} \to \mathbb{R}]$ are defined such that the following holds almost surely,

$$\mathbb{E}[u(O, A)\{R + \gamma b_V(F')\} - b_V(F)|H] = 0$$

Then with these two definitions, the theoretical result[4] shows that these two functions are equivalent. Then we can use the latter one, i.e. the off-policy Bellman Equation in POMDP to learn the function values of π^e.

Algorithm

After exploring the equivalence of future-dependent value functions and learnable future-dependent value functions, by the property of the learnable future-dependent functions b_V:

$$\mathbb{E}[L(b_V, \xi)] = 0$$

for any $\xi : \mathcal{H} \to \mathbb{R}$, where $L(q, \xi) := [u(A, O)\{R + \gamma q(F') - q(F)\}\xi(H)]$, we can derive a Minimax algorithm to learn the desired estimated function for $V^{\pi^e}(S_0)$.

```
Algorithm 1 Minimax OPE on POMDPs
Require: Dataset $\mathcal{D}$ , function classes $\mathcal{Q} \subset [\mathcal{F} \to \mathbb{R}]$, $\Xi \subset [\mathcal{H} \to \mathbb{R}]$, hyperparameter $\lambda \geq 0$
  $\hat{b}_V = argmin_{q \in \mathcal{Q}} \max_{\xi \in \Xi} \mathbb{E}_{D_{tra}}[\{u(A, O)\{R + \gamma q(F')\} - q(F)\}\xi(H) - \lambda \xi^2(H)]$
  return $\hat{J}_{\pi^e} = \mathbb{E}_{D_{ini}}[\hat{b}_V(f)]$
```

Experiment Design: Environment

Since the real robot navigation experiment needs complex setting and simulators, which is costly and risky, researchers usually evaluate their trained policies first in some simple environments. Thus our project offers a simple application strucuture for these tasks. Then we can extend the result in the real complex environment in the future.

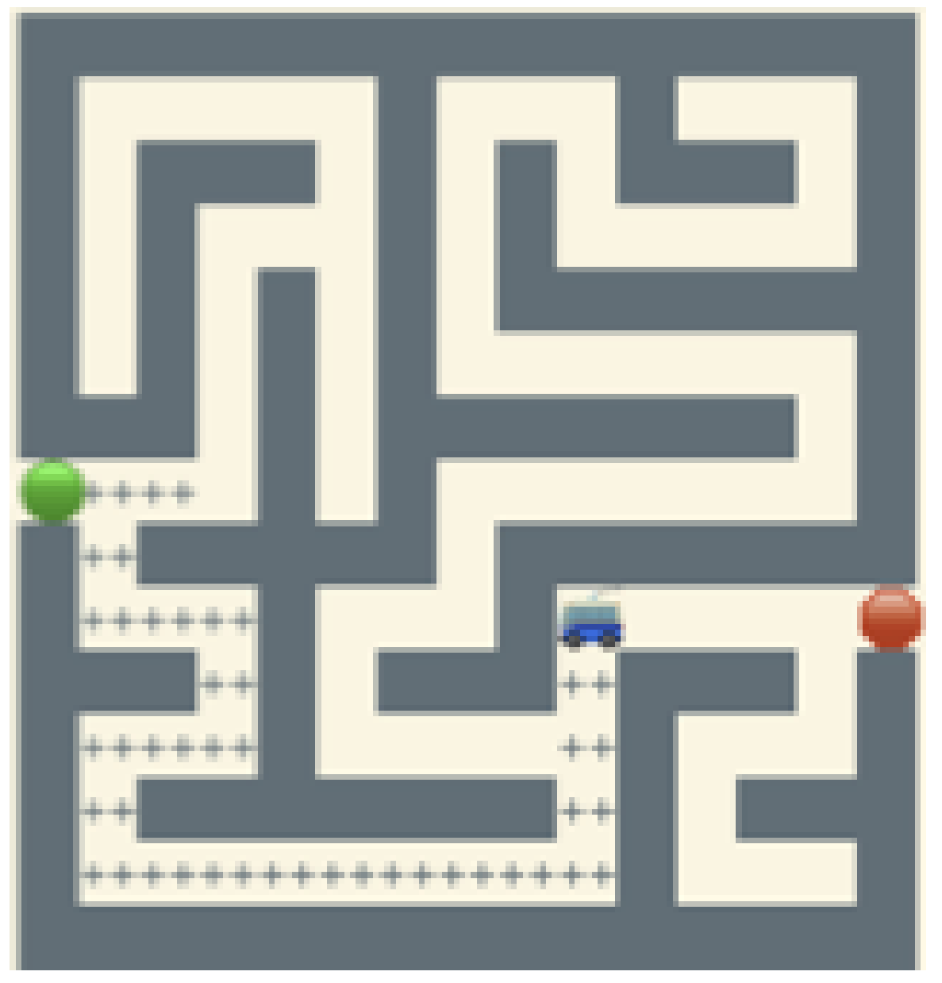

Fig. 1: Labyrinth-Escape Environment

We use the Labyrinth-Escape Environment provided by PopGym [1], which is a benchmark library of POMDP environments. The environment will randomly generate a maze with given size, and the agent can only observe the surrounding grids and cannot know the real state(position in the maze). Thus it's a partially observable setting.

Experiment Design: Policy Evaluation

To get behavior and evaluation policies, we use Double Q-Learning algorithm[5] to train a policy. Then we apply this policy in two data sets: one contains the set of pairs of states and actions(S, A), and the other contains the set of pairs of observations and actions(O, A). By applying this policy in the two datasets, we define two ϵ-greedy policies. Then the former one is the behavior policy and the latter one is the evaluation policy.

For the off-policy evaluation, we choose two-layer neural networks for the function classes $\mathcal{Q}$ and RKHSs for function classes Ξ in the above algorithm. After computation, we compare the result with two other off-policies. One is the Sequential Importance Sampling method [2]:

$$\hat{J}_{\pi^e} = \mathbb{E}_{\pi^b}[\sum_{t=0}^{T-1}\{\prod_{k=0}^{t}\frac{\pi^e(a_k|o_k)}{\pi^b(a_k|o_k)}\}R_t]$$

And the other is the Minimax OPE method for fully observable MDP[3]. In this method we just substitute $\mathcal{F}$ and $\mathcal{H}$ by $\mathcal{O}$ (observation space) in the above algorithm.

Hopeful Result

1. Apply this algorithm successfully in the Labyrinth-Escape environment, and get a convergent and more accurate value function than other two off-policy evaluation methods.
2. Explore the appropriate future and history steps $(M_H$ and $M_F)$ under the simple environment of robot navigation task, and give a reasonable explanation (or guess).

References

[1] Morad, Steven, et al. "POPGym: Benchmarking Partially Observable Reinforcement Learning." arXiv preprint arXiv:2303.01859 (2023).

[2] Precup, Doina. "Eligibility traces for off-policy policy evaluation." Computer Science Department Faculty Publication Series (2000).

[3] Uehara, Masatoshi, Jiawei Huang, and Nan Jiang. "Minimax weight and q-function learning for off-policy evaluation." International Conference on Machine Learning. PMLR, (2020).

[4] Uehara, Masatoshi, et al. "Future-dependent value-based off-policy evaluation in pomdps." arXiv preprint arXiv:2207.13081 (2022).

[5] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Proceedings of the AAAI conference on artificial intelligence. Vol. 30. No. 1. (2016).

[6] Levine, Sergey, et al. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." arXiv preprint arXiv:2005.01643 (2020).

[7] Zhu, Kai, and Tao Zhang. "Deep reinforcement learning based mobile robot navigation: A review." Tsinghua Science and Technology 26.5 (2021).

[8] Uehara, Masatoshi, Chengchun Shi, and Nathan Kallus. "A review of off-policy evaluation in reinforcement learning." arXiv preprint arXiv:2212.06355 (2022).