Robot Navigation Off-Policy Evaluation Based on Minimax OPE on POMDP Algorithm

Yi Wei
Junior Undergraduate advised by Yunfu Deng, CS Phd in Robotics
ywei224@wisc.edu,

***Abstract*— This report explores the application of Minimax Off-Policy Evaluation (OPE) on Partially Observable Markov Decision Processes (POMDPs) algorithm, particularly in the context of robot navigation within the Labyrinth-Escape Environment. Addressing the challenge of effective policy evaluation without the curse of history, this study investigates the efficiency of Minimax OPE in leveraging future-dependent value functions. And we focus on the influence of Memory Horizon (M_H) and Memory Future (M_F) steps on the accuracy of policy evaluation.**

I. INTRODUCTION

In the ever-evolving field of mobile robotics, navigation stands as a cornerstone of autonomy, a challenge that has long captivated researchers and engineers alike. Robots, designed to traverse diverse and unpredictable environments, must interpret complex sensor data to navigate effectively. This demands advanced algorithms that can chart optimal paths, avoid obstacles, and adapt to new terrains in real-time. The sophisticated nature of these tasks necessitates a deep integration with dynamic programming, particularly Partially Observable Markov Decision Processes (POMDPs), which enable robots to make informed decisions even with incomplete information. Through this lens, our study seeks to further the capabilities of robotic systems, pushing the boundaries of autonomous navigation.

Experimental iterations in this domain are often costly and fraught with risks, making the effective use of historical data through Off-Policy Evaluation (OPE) methods not just advantageous, but essential. These methods aim to estimate the expected rewards of a novel policy, leveraging data from previously enacted policies without additional costly data collection.

Conventional OPE methodologies, however, face the 'curse of history', a phenomenon where errors increase exponentially with the length of data history considered. The Minimax OPE algorithm for POMDPs, predicated on Future-Dependent Value Functions, offers a promising advance to address this issue.

This paper builds upon the previous work which largely focused on modified fully observable MDPs with artificial disturbances, rather than on authentic POMDP environments. We aim to adapt the theoretical Minimax OPE model to the nuanced domain of robotic navigation within true POMDPs, circumventing historical constraints and offering a pragmatic framework for real-world application.

In addition to the methodological adaptation, our study scrutinizes the impact of Memory Horizon (M_H) and Memory Future (M_F) on policy evaluation accuracy, emphasizing the selected history and future memory steps' influence on learning outcomes. Consequently, this paper presents our novel contributions in the empirical analysis, marking a contribution in the field of robotics navigation.
Thus our novel contributions in this project are:

1) Construct a native Partially Observable environment for simple robot navigation and implement Minimax OPE for POMDP in this environment. But through the experiments we find that this algorithm doesn't work in this environment as well as it does in those originally fully observable environment with normal disturbance in the previous work.
2) Analize the influence for the memory horizon M_H and memory future M_F chosen in Labyrinth-Escape environment and the inspiration for robot navigation policy evaluation.

II. Theoretical Foundation

A. Problem Formulation

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 $A^{(i)}_{M_{F-1}}$.

B. Important Definition

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.

C. 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 $\lambda \geq 0$

1: $\hat{b}_V = argmin_{q \in \mathcal{Q}} \max_{\xi \in \Xi} \mathbb{E}_{D_{tra}}[\{u(A, O)\{R+ \gamma q(F')\} - q(F)\}\xi(H) - \lambda\xi^2(H)]$

2: **return** $\hat{J}_{\pi^e} = \mathbb{E}_{D_{ini}}[\hat{b}_V(f)]$

Note that the function classes Ξ, $\mathcal{Q}$, hyperparameter λ, memory horizon M_H and future memory steps M_F are not defined in this algorithm.

III. Environment Construction and Experiment

A. Environment

In this study, we construct the Labyrinth-Escape environment, designed to simulate complex decision-making scenarios in robot navigation within Maze structure from the mazelib based

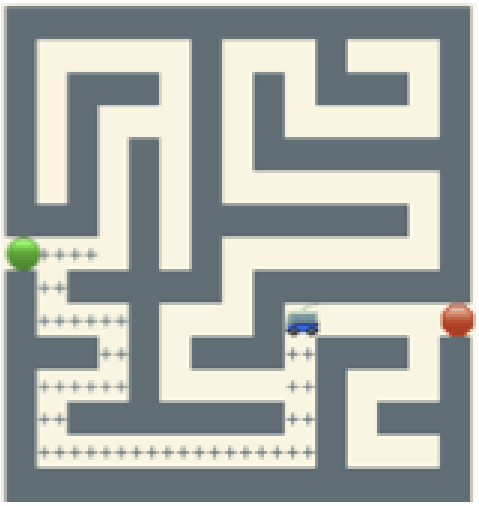

Fig. 1. Labyrinth Escape Environment.

on OpenAI Gym. Unlike environments with clear visibility of states, such as Cartpole, the Labyrinth-Escape presents an intricate state space with obstacles and varying pathways that challenge the agent's navigational capabilities.

To create the partial observability compared to other maze-like environments, given a n $\times$ n matrix maze, we define an attribute Cell of 5 types for each grid in the maze: START, GOAL, FREE, OBSTACLE and HIDDEN. The latent state in every stage is the map with the perfect information(No HIDDEN grids) for every grids in the maze. But the observation in every stage is the incomplete map of the maze. Only the grids that have been explored or within the 9 grids centered by the agent will be observed correctly. Other grids will remain HIDDEN. Thus it's a natrual partially observable environment. Furthermore the action space in every stage is up, down, right, and left. And we set the parameters to randomly generate the maze for given size and limit the max steps the agent can take.

As for the reward, for given max steps n the agent can take, the reward the agent gets to the GOAL is 1. And if the agent don't move(step to obstacle grids), the reward is $-\frac{2}{n}$. Otherwise the reward for exploring is $-\frac{1}{n}$.

B. Experiments

To check the validity and efficiency of the algorithm, we make an experiment to compare it with two other off-policy evaluation methods. One is the Sequential Importance Sampling method [2]:

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

, which is one of the classic OPE methods suffering curse of history, and the error will cumuluate to grow exponentially. 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.

Thus our experiment consists of 3 parts: the first part is to generate the required behavior policy and the evaluation policy. And we need to acquire the dataset generated by the behavior policy. The second part is to use these three methods to evaluate the policy value of the evaluation methods. And the third part is to use statistics method to estimate the error and compare their results.

Firstly, to get behavior and evaluation policies, we use Double Q-Learning algorithm[5] based on Pytorch library to train poor, median and expert policy sequentially to acquire a good policy for the environment. Here we use 10000 train steps. In each of those, We conduct 10 experiments and keep record of the rewards for each experiment. Then we compute the mean of the cumulative rewards. If the mean value is in $[-0.6, -0.3]$, the policy is logged as a poor policy. If the mean value is in $(-0.3, 0]$, the policy is logged as a median policy. And if the mean value is larger than 0, we get an expert policy as expected. Then we use Behavior Cloning method to 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). Based on the two policies trained above, we define two ϵ-greedy policies. Here we define ϵ to be 0.2. Then the former one is the behavior policy used to generate trajectory data and the latter one is the evaluation policy. This makes sense since in the context of off-policy evaluation, we assume the offline data generated by behavior policy is of good quality. Thus most of experiments conducted in the research of this area generate behavior policy on

the dataset containing latent states to get a good policy and relavent data.

Then with the behavior policy, we conduct simulation in this environment 2000, 5000, 10000 times, each of which contains at most the max steps the agent can take defined in the environment by the learner, to collect offline trajectory data containing the information of the observations, states, rewards, emission kernel, transition kernel, reward functions and initial state distribution. And we sort it into different data tuples described in the Part $II.A$.

Secondly, here we parameterize the function classes Ξ and $\mathcal{Q}$ for the naive OPE Minimax algorithm and OPE Minimax for POMDP algorithm both by a two-layer neural network with layer width equal to 32 and ReLU as activation function. And the functions are optimized using the algorithm above by a kernel loss function.
The loss function is given by:

$$\max_f L_V^2(g,f) = \mathbb{E}(\bar{R}^\pi(A|O) + \gamma \mathbb{E}_{a'\sim\pi}[g(a', O^+)] \cdot \pi^e(A|O) - \pi^e(A|O)g(A,O)) \cdot (\bar{R}^\pi(\bar{A}|\bar{O}) + \gamma \mathbb{E}_{a'\sim\pi}[g(\bar{a}', \bar{O}^+)]\pi^e(\bar{A}|\bar{O}) - \pi^\pi(\bar{A}|\bar{O})g(\bar{A},\bar{O}))K(A,O^-;\bar{A},\bar{O}^-).$$

where

$$K(x_1;x_2) := \exp(-\frac{2\left\|x_1 - x_2\right\|_2}{2\beta^2})$$

is the RBF kernel where β denotes the bandwidth parameter. Let m be the median of the l2-distance over the samples in the dataset, we define $\beta = \frac{m}{5}$ in the algorithm. And we choose the hyperparameter λ in the algorithm to be 0.5. And here we choose $M_H = 3$ and $M_F = 2$

Thirdly we compare the results of Sequential Importance Sampling method, Minimax OPE method for fully observable MDP and Minimax OPE method for POMDP. We use the statistics measure methods below:
Given n datasets $D_1, D_2, \ldots, D_n$, the estimators computed based on each dataset $\hat{V}_1, \ldots, \hat{V}_n$, and the true value V, we define the relative bias to be

$$\left| \frac{1}{n} \sum_{i=1}^{n} \frac{\hat{V}_i}{V} - 1 \right|. \tag{1}$$

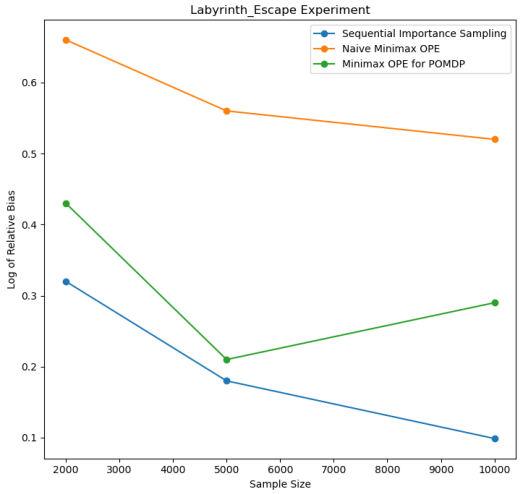

Fig. 2. Result of relative Bias in Labyrinth-Escape Environment

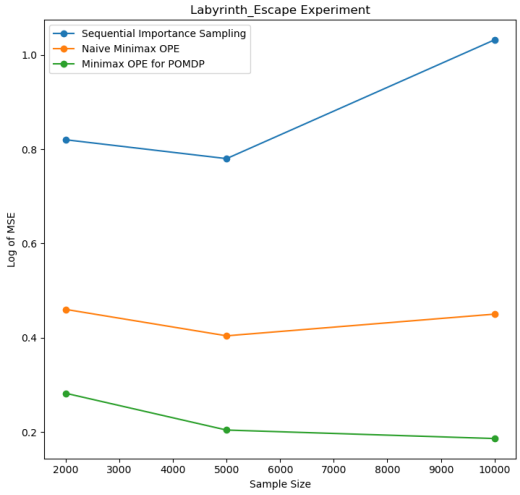

Fig. 3. Result of relative MSE in Labyrinth-Escape Environment

Define the relative mean squared error to be:

$$\left| \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\hat{V}_i - V}{V} \right)^2 \right|. \tag{2}$$

In our experiments, we use the above two definitions to measure the estimation error of different estimators.

C. Results

Our experiment results are presented in Figure 2 for the relative bias and Figure 3 for the relative MSE in the Labyrinth-Escape environment. Especially, we compare the relative MSE result(Figure 4) in the Cartpole environment, which is originally

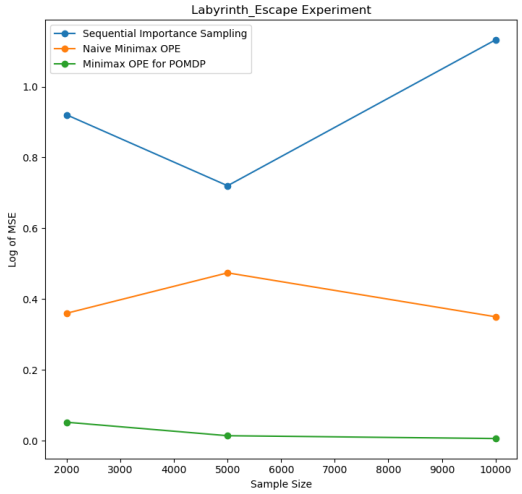

Fig. 4. Result of relative MSE in Cartpole Environment

fully observable but created with partial observability by adding normal disturbance to the latent state. Here the observation noise is $\mathcal{N}(0, 0.3^2)$. We use the setup in [9] to construct this cartpole environment. From the graph and data results, we yielded several insights regarding the behavior of bias and error in policy evaluation methods as the sample size changes:

1) As the sample size increases, the bias within estimations decreases. Particularly, Sequential Importance Sampling (SIS) exhibits stable and unbiased estimations, which is indicative of its superior performance in terms of bias behavior.
2) Despite its advantages in bias reduction, SIS is susceptible to the 'curse of history.' The error tends to accumulate and grow exponentially with larger sample sizes, diminishing the reliability of SIS for extensive data.
3) The performance of Minimax Off-Policy Evaluation (OPE) for Partially Observable Markov Decision Processes (POMDPs) was superior to the two baseline methods in terms of Mean Squared Error (MSE), suggesting a more stable estimation capability. showing its superior advantages with
4) Contrary to expectations and previous work, the Minimax OPE did not perform consistently across different environments. In the labyrinth environment, it yielded a larger MSE compared to the cartpole environment. This discrepancy could stem from the fact that environments with originally fully observable states, perturbed by normal disturbances, may be easier to estimate by the Minimax method. But different dynamic problem structure may damage its accuracy since in maze-like environment for robot navigation problem, the distribution of the emission kernel is usually hard to learn. This suggests that model-free policy evaluation methods might not be universally effective across different tasks.
The results underscore the importance of considering the specific structure of tasks when applying policy evaluation methods. For complex tasks, such as navigating labyrinth environments, a tailored approach that accounts for the unique environmental dynamics is crucial for accurate policy evaluation.

IV. Results for varying M_H and M_F

Now we analyze the influence of different memory horizon steps M_H and memory future steps M_F in Labyrinth-Escape environment. We remain the same settings of Minimax OPE method for POMDP. And we explore how to improve the accuracy and stability of policy evaluation.
We firstly fix M_F to be 0 and change M_H to be 1, 2 and 3. Then we fix M_H to be 1 and change M_F to be 0, 1 and 2. Our results are presented in Figure 5 to Figure 8 of the relative bias and MSE for varying M_H and M_F respectively. The results show that:

1) Memory Horizon (M_H) Diminished Importance: The experiments find that the performance doesn't change too much with varying M_H, and thus reveal that in simple robot navigation environment like Labyrinth-Escape, the current observations often encapsulate essential information from past explorations. This finding suggests that M_H, or the reliance on extensive historical data, is less critical than previously thought. We just need sufficient small number of history horizon for policy evaluation.
2) Critical Role of Memory Future (M_F) Steps: We found from the results that the accuracy

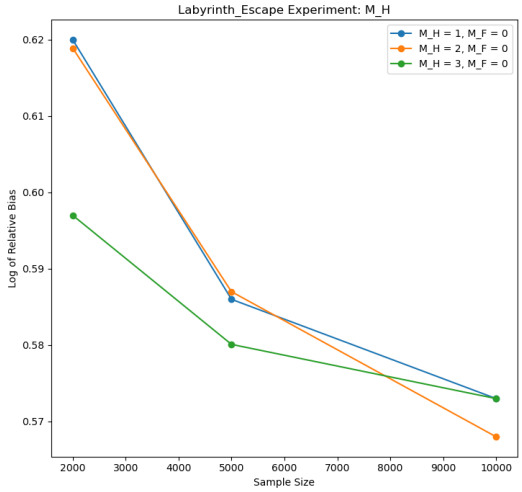

Fig. 5. Result of relative Bias with varying M_H steps

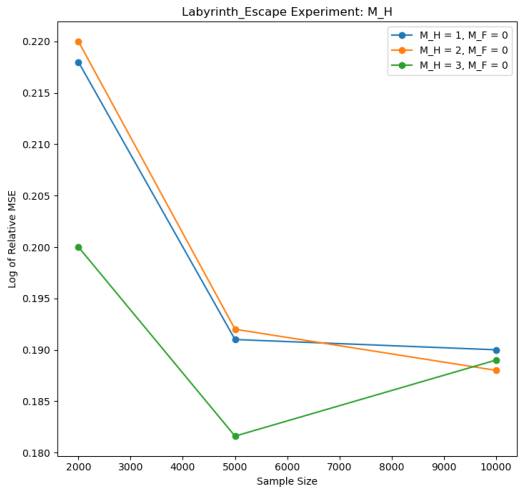

Fig. 6. Result of relative MSE with varying M_H steps

and stability of the Minimax OPE method improve with longer future steps. The improved performance may stem from the accurate estimation of the stage rewards by the future information.

3) Limitations of Large M_F Steps: A key takeaway from the previous theoretical work[10] is the risk associated with overly extending M_F steps. Excessively long M_F steps can lead to a phenomenon known as the "curse of history", where the accumulation of errors grows exponentially. Then the algorithm will encounter similar problems as Sequential Importance Sampling method.
4) Optimal Strategy - Balancing M_H and M_F: our study suggests an optimal strategy for simple robotic navigation policy evaluation with Minimax OPE algorithm : combining longer M_F steps with a shorter sufficient M_H. This approach aims to make good trade-off between the chosen data horizon and future memory steps to improve the performance of the algorithm and avoid exponential cumulative error problem in robot navigation. And it's one of the important directions in which we can extend our project's result.

V. Conclusion

Our project successfully adapted the Minimax Off-Policy Evaluation on POMDP algorithm to

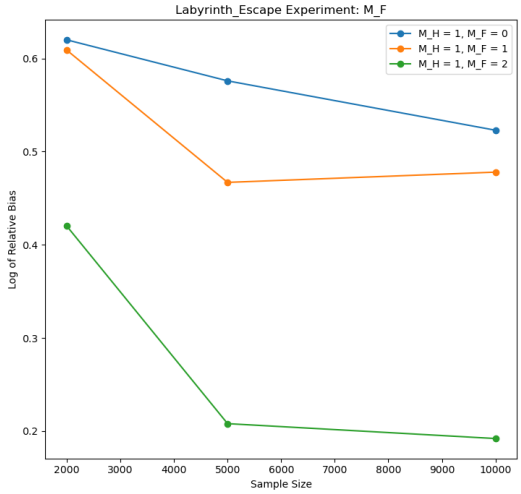

Fig. 7. Result of relative Bias with varying M_F steps

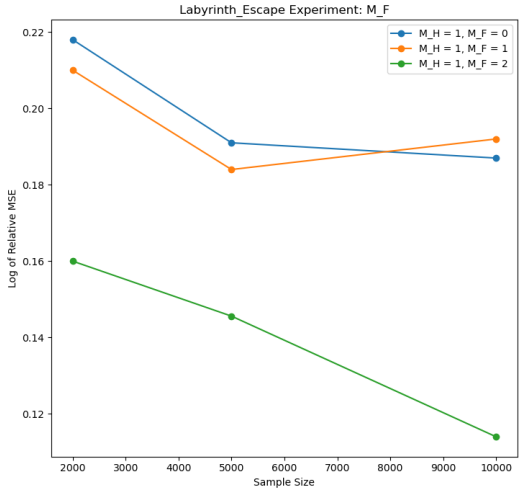

Fig. 8. Result of relative MSE with varying M_F steps

Labyrinth-Escape, a partially observable MDP environment. And this study explores how Memory Horizon (M_H) and Memory Future (M_F) steps influence the accuracy of policy evaluation. Results indicated that the performance of Minimax OPE varies significantly with these parameters, emphasizing their importance in robot navigation scenarios.

Furthermore, the experiments provided valuable insights into the applicability and limitations of the Minimax OPE POMDP algorithm in a native POMDP environment. Notably, the algorithm's performance differed from expectations, especially compared to that in environments with originally fully observable states that were disturbed, suggesting the need for further research and refinement for specific problem structure.

Our project also suggests other future research directions, such as exploring the optimal balance between M_H, M_F and total memory horizon steps, and extending the application of this approach to more complex navigation tasks and real environments.

Finally, the project contributes to the field of robotic navigation by providing a practical framework for policy evaluation in POMDPs and highlighting the critical factors that influence the effectiveness of such evaluation methods.

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).

[9] Shi, Chengchun, et al. "A minimax learning approach to off-policy evaluation in confounded partially observable markov decision processes." International Conference on Machine Learning. PMLR, 2022.

[10] Liu, Qiang, et al. "Breaking the curse of horizon: Infinite-horizon off-policy estimation." Advances in neural information processing systems 31 (2018).