State Visitation Correction Robustness in Off-policy Reinforcement Learning

Mayank Prashar*, Jochem van Lith *, Joost Jansen*

March 8, 2024

Abstract

With Reinforcement Learning (RL) expanding to diverse domains such as healthcare and robotics, its application is limited by the cost and potential risk of exploring new policies in safety-critical applications. Hence, the ability to use prior data to evaluate new policies is becoming increasingly important; commonly known as the Off-Policy Evaluation (OPE) problem. Therefore, this research investigates the impact of the difference between the evaluation policy π_e and the behaviour policy π_b on the bias and variance of OPE estimators. The difference between policies is quantified using the state-visitation difference; a metric that also accounts for the environment. Experiments conducted using the Spectrum of Off-Policy Estimators (SOPE) on the Graph and GridWorld environments found that in general, distribution-based importance sampling (SIS) performs best with larger state-visitation difference. The findings can aid in enhancing existing techniques and understanding the limitations of estimators when used in specific application contexts.

1 Introduction

Reinforcement Learning (RL) has demonstrated success in applications such as AlphaGo [10] and Atari [3], and is now being developed for use in more domains such as healthcare, recommendation systems, and robotics. However, in such applications, it can be expensive, impractical, or even hazardous to explore new policies [12]. As a result, the ability to use prior data to evaluate new policies is becoming increasingly important. This is also known as counterfactual reasoning, where we must reason about some outcome while only having data from some alternative outcome [6]. In the RL setting, this is commonly known as the off-policy evaluation (OPE) problem. This involves estimating the expected return of a policy π_e (the evaluation policy) under the distribution of state-action trajectories that π_b (the behaviour policy) induces.

Consequently, having a π_e that is too dissimilar to the π_b would result in unreliable estimates of its value. This is because the data collected under π_b may have different propensities to visit certain states or take certain actions that reflect their interpretation of the environment. For example, suppose π_b is much more conservative than π_e, this would result in limited data and inaccurate predictions of the expected value of states visited by π_e. This can have serious consequences in applications that are safety-critical such as autonomous driving or healthcare. Therefore, this paper investigates how the difference between the π_e with respect to the π_b affects the bias and variance of OPE estimators. In other words:

To what extent does the difference between the evaluation policy (π_e) and the behaviour policy (π_b) affect the bias and variance in the Spectrum of Off-Policy Estimators (SOPE)?

and the following sub-questions:
(i) What parameters in the estimators affect this change the most?
(ii) How can we determine if a policy is too different before evaluation?

*Authors are master's students at the Department of Cognitive Robotics (CoR) and Department of Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology, 2628 CD Delft, Netherlands. E-mail address: {M.Prashar, J.A.E.vanLith, J.J.Jansen-2 } @student.tudelft.nl
* Equal contribution

To address the above, experiments are performed in a simple graph and gridworld environment using the Kullback-Leibler (KL) divergence to compare the state-visitation distribution of the behaviour policy and the evaluation policy. The SOPE estimator [1] is employed for this analysis. As it's possible with SOPE to test a number of different estimators that lay on the spectrum between SIS [4] and PDIS [7] with one formula. Based on the results of the experiments, the following insights can be drawn:

- The best-performing estimator is highly problem specific
- For a large difference between policies, distribution-based importance sampling generally shows better results
- Better distribution ratio estimates can improve the performance of OPE estimators

The findings of this study hold significant relevance for the practical implementation of off-policy RL. Gaining insights into the comparative performance of different OPE algorithms under varying π_e and π_b conditions can guide future research toward enhancing existing techniques. Furthermore, it can contribute to a better comprehension of the limitations with current estimators and offers guidance on the acceptable deviation between π_e and π_b to have meaningful results, though this varies depending on the specific target application.

Section 2, presents the background information relevant to the experiments conducted. In Section 3, existing body of work in this domain is studied, presenting its significance and drawing relevant comparisons. Section 4 outlines the methodology to evaluate SOPE with varying KL divergence. Subsequently, Section 5 details the experimental setup and presents the results obtained. In Section 6, a summary of the findings and insights are described.

2 Background

This section provides the necessary background material and notation related to the off-policy evaluation (OPE) problem.

2.1 Markov Decision Process

A Markov decision process (MDP) is defined as a tuple $M = (S, A, T, R, \gamma)$, where S is the state space of the system, A is the set of actions the system can perform, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function where $T(s, a, s') = Pr(s'|s, a)$, R is the reward function defined on the transitions of the system, and γ is the discount factor [11].

A policy π is a distribution over actions conditioned on the current state. Starting from an initial state S_1 drawn from the initial state distribution d_1, the policy samples an action A_t at each time step t from the distribution $\pi(\cdot|S_t)$. The environment provides a reward R_t with an expected value determined by the reward function for the given state-action pair $r(S_t, A_t)$. The environment also transitions to the next state S_{t+1} according to the probabilities defined by the transition function for the given state-action pair $T(S_t, A_t, S_{t+1})$.

A trajectory τ refers to a sequence of random variables representing the states S, actions A, and rewards R encountered when following a certain policy. The length of the trajectory is denoted by L, also referred to as the horizon length. We denote the distribution of trajectories under policy π by p_π.

2.2 Off-Policy Evaluation

The value of a policy π is determined by the expected discounted sum of rewards it generates: $J(\pi) := \mathbb{E}_{\tau \sim p_\pi}[\Sigma_{t=1}^{L} \gamma^{t-1} R_t]$. In the off-policy evaluation problem, we aim to estimate the expected discounted sum of rewards of π_e using only a batch of data $D := [\tau_i]_{i=1}^{m}$ gathered by a different behaviour policy π_b. The key challenge this raises is the data distribution mismatch between π_e and π_b. Each policy guides the agents' next state within the environment, resulting in different data distributions. When the policies differ greatly and there is a situation where π_b has no probability of taking an action where π_e does, $IS(D)$ cannot be used as the actions are never observed in D that the evaluation policy would take. Therefore, a key assumption is that the policies have full coverage:

Assumption 1 ***Coverage of Optimality***

$\pi_b(a|s) > 0 \ \ \forall\, a, s \ \ s.t.\ \pi_e(a|s) > 0$ ***or:***

$$\frac{\pi_e(a|s)}{\pi_b(a|s)} < \infty \ , \forall\, a \in A, \forall\, s \in S.$$

2.3 Estimators

Estimators play a crucial role in OPE. They are used to estimate the expected value of a function under π_e using samples generated from π_b. In this section, we will discuss the estimators commonly used in OPE.

Importance sampling (IS) Importance Sampling (IS) is a widely used technique in off-policy evaluation. It corrects for the discrepancy between π_e and π_b by re-weighting the samples. The general form of IS is the following:

$$\mathbb{E}[g(X)] = \mathbb{E}[\frac{p_X(Y)}{p_Y(Y)} \cdot g(Y)] \tag{1}$$

Here, X and Y represent random vectors of the same size with probability mass functions $p_X(x)$ and $p_Y(y)$ respectively, and $g(X)$ is the function g(Y) rescaled by the ratio $p_X(Y)/p_Y(Y)$. For IS to hold: $p_Y(x) > 0$ whenever $p_X(x) > 0$, which follows from Assumption 1. Various estimators that make use of this technique are described below:

Trajectory based IS Trajectory-based estimators estimate the value of π_e by considering entire trajectories sampled from π_b. These estimators use the idea of importance sampling to correct for the differences between the two policies.

A commonly used trajectory-based estimator is Per-Decision Importance Sampling (PDIS) [7]. It calculates the expected return of π_e by re-weighting the returns of individual trajectories based on the importance sampling ratio until the current time-step: $\rho_{1:t}$. Unlike traditional IS that computes the ratio for all steps $\rho_{1:T}$, PDIS uses the ratios only till the current time step as the later steps shall not influence the ratio at some step t. The PDIS estimator can be expressed using the following equation:

$$\text{PDIS}_n(D) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{n} \gamma^{t-1} \rho^i_{1:t} R^i_t \tag{2}$$

Here, N is the number of trajectories, n is the horizon length, and $\rho^i_{1:t} = \frac{\pi_e(A_t|S_t)}{\pi_b(A_t|S_t)}$ is the IS likelihood ratio from steps 1 till t. According to the findings of [8], employing this approach leads to an unbiased estimator of $J(\pi_e) = \mathbb{E}_{\tau\sim\pi_b}[PDIS(\tau)]$. However, despite its unbiased nature, this estimator is afflicted by a phenomenon known as the *"curse of horizon,"*[5]. The curse of horizon manifests in the exponential dependence of the estimator's variance on the length of the trajectory, resulting in a rapid increase in variance as the trajectory length grows.

Distribution based IS To remove the dependence on the trajectory length and thereby break the curse of the horizon, IS can be directly applied to the stationary state-visitation distributions of π_e and π_b [9]. Averaging these probabilities over time gives the stationary state-visitation distribution d^π for that policy. The ratio of these distributions for the behaviour and evaluation policies can be used as weights in IS, such that the dependence on the trajectory length is eliminated. One commonly used distribution-based estimator is the Stationary-distribution Importance Sampling (SIS) estimator. The SIS estimator can be expressed as follows:

$$\text{SIS}_n(D) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=n+1}^{\infty} \gamma^{t-1} w(S^i_{t-n}, A^i_{t-n}) R^i_t \tag{3}$$

where $w(s,a) = \frac{d^{\pi_e}(s,a)}{d^{\pi_b}(s,a)}$ is the distribution correction and n is the starting point of the distribution measure. This function eventually gives $J(\pi_e) = \mathbb{E}_{\tau\sim\pi_b}[SIS(\tau)]$. Unfortunately, estimating these distributions $w(s,a)$ accurately is often challenging and many techniques have been explored in recent works [9] [4] [2].

Trajectory and distribution IS To capture the unbiasedness of trajectory-based IS and the lower variance of distribution-based IS, Spectrum of Off-Policy Estimators (SOPE) was introduced [1]. This estimator interpolates between trajectory- and distribution-based methods: PDIS and SIS respectively:

$$\text{SOPE}_n(D) = \frac{1}{N} \sum_{i=1}^{N} \text{PDIS}_n(\pi_b) + \text{SIS}_n(\pi_b) \tag{4}$$

Here n represents the point at which SOPE transitions from using PDIS to SIS. Notably, when $n = 0$, SOPE reduces to SIS, which is biased but has a lower variance. Conversely, when $n = L$, SOPE becomes equivalent to PDIS, which is unbiased but has a higher variance. The choice of n determines the trade-off between bias and variance in the SOPE estimator. By employing the SOPE estimator, we can assess how the disparity between π_e and π_b affects the bias and variance of estimators. This approach

combines the strengths of both trajectory-based and distribution-based estimators, allowing us to examine the impact of these differences on estimation accuracy.

The equation 4 initially describes the unweighted variant of SOPE. However, going forward, we will adopt the weighted version of SOPE due to its computational advantages. This weighted version normalizes the weights of PDIS and SIS based on the total number of steps N, resulting in a biased but consistent PDIS. Consequently, the variance is reduced, requiring fewer repetitions to obtain accurate results. Therefore, to mitigate computational overhead, we will exclusively employ the weighted SOPE approach.

2.4 State-Visitation Difference

The state-visitation difference is a metric used to measure the discrepancy between π_e and π_b. This metric is defined by the Kullback-Leibler (KL) divergence between the state-visitation distributions of both policies. The state-visitation distribution takes both the policy and the environment into account, making it a meaningful property to compare policies intended to be deployed in the same environment. To compute this distribution, we define P^π to be the square matrix such that $P_\pi(s', s) = \sum_a T(s'|s, a)\pi(a|s)$. For a finite state and action space the state-visitation distribution is then the following:

$$\begin{aligned} d_\pi^{s_0} &= (1-\gamma)\sum_{t=0}^{\infty}[\gamma^t (P^\pi)^t]s_0 \\ &= (1-\gamma)(\mathbb{I} - \gamma P^\pi)^{-1} s_0 \end{aligned} \tag{5}$$

The state-visitation difference can be calculated as the KL divergence between the behavior policy distribution $d_{\pi_b}^{s_0}$ and the evaluation policy distribution $d_{\pi_e}^{s_0}$. The KL divergence is used as it not only captures the difference between the state-visitation distributions but also provides a statistical meaning relating to how much information in the two distributions is identical.

Definition 1 ***State-Visitation Difference***

$$KL(d_{\pi_e}^{s_0} \parallel d_{\pi_b}^{s_0}) = \sum_x d_{\pi_e}^{s_0} \log\left(\frac{d_{\pi_e}^{s_0}}{d_{\pi_b}^{s_0}}\right)$$

As the KL divergence is asymmetric, and to ensure consistency in the results, the same order is used in all experiments: evaluation with respect to behavior state-visitation distributions. This order is chosen as it also best represents the IS ratio as shown in subsection 2.3.

3 Related work

In the Caltech OPE Benchmarking Suite [13], the authors measure the mismatch between the evaluation policy and behaviour policy using the supremum ratio as shown in Equation 6 , which is independent of the environment. However, incorporating information about the environment, if available, can provide a more comprehensive understanding of the actual difference between policies by considering the effects of state transition dynamics and reward structure. As an example, with one policy, the agent may become trapped in a state; which cannot be anticipated by solely examining the policy without considering the environment. These are specific scenarios that should be captured in the policy mismatch, as the induced effect of the policy is what the agent encounters. Therefore, our primary objective is to explore the actual induced difference between policies, considering both the measure of mismatch independent of the environment and the effect due to the state transition dynamics and reward structure (environment).

$$sup_{a\in A, x\in X}\left(\frac{\pi_e(a|x)}{\pi_b(a|x)}\right)^T \tag{6}$$

Previous works [1, 4, 9] have chosen behaviour and evaluation policies arbitrarily when developing OPE estimators, and have focused majorly on mitigating the curse of the horizon. However, it has been shown that the policy mismatch can be an equally, if not more, crucial aspect influencing the performance of OPE estimators [13]. Therefore, it is crucial and intriguing to investigate the extent of performance differences across estimators when policies vary significantly. By considering both the measure of mismatch independent of the environment and the impact of the environment's characteristics, our study aims to provide a more comprehensive evaluation of OPE estimators and shed light on the variations in performance that can arise from policy differences and their state-visitation difference.

4 Method

As our research question states; we are interested in how the difference between π_e and π_b affects the bias and variance of estimators. We already defined the state-visitation difference in Section 2, which is a metric to quantify the difference between two policies while taking the environment into account. In this section we will explain how we can utilize this metric to evaluate estimators. The following steps describe our method:

1. Given an environment ENV and a behaviour policy π_b.
2. Derive the range of possible state-visitation differences between any valid policy and π_b in ENV (using Equation 5 and Definition 1).
3. Derive a set of π_e's corresponding to fixed steps within the state-visitation difference range (using Equation 5 and Definition 1).
4. For each π_e in the set, perform the following steps:
   (a) Rollout π_b to gather data. [1]
   (b) Let the estimator estimate the expected return of π_e using the gathered data.[1]
   (c) Run π_e online to find its ground-truth expected return.[1]
   (d) Compute the Mean Squared Error (MSE), bias and variance between the estimated expected return and the ground-truth expected return.
5. Plot the results against an ascending state-visitation difference.

Note that step 2 and 3 require the transition function T to be known, which is typically not the case in a real-world setting. However, in our experiments, we make use of synthetic environments where we do have access to T.

5 Experiments

In the last section, we have explained the general method to evaluate estimators for different state-visitation differences between π_e and π_b. We have applied this method to the Graph and GridWorld environments. These environments will be described next along with their results.

5.1 Graph Environment

We initially conduct experiments in the Graph environment Figure 1 due to its simplicity and ability to control for other design factors that are known to influence the performance of OPE methods including horizon length, reward sparsity, and environment stochasticity [13]. Here the environment stochasticity is referred to as slippage, and is the deviation from the agent's intended action, e.g. slippage = 0.25 means there is a 25% chance that the agent will take an unintended action. Further, we control for policy approximation errors by using a tabular policy instead of a parametric policy, as a more complex model of the policy network has been shown to affect the performance of OPE estimators [13]. To simplify our analysis, we also keep the policy state independent, i.e. the policy has the same probability of taking an action at each state. By controlling and keeping these variables constant we can measure solely how much the difference in state-visitation difference affects the performance of current OPE estimators by measuring the mean-squared error: the difference in the value between the estimator and ground truth, the variance, and bias of the estimator.

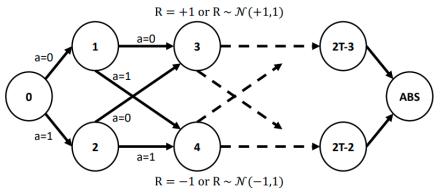

Figure 1: Graph Environment. Fixed horizon: T, Absorbing state: 2T. Agent starts at state 0 and can choose between two actions: 0 (Up), and 1 (Down) at each time step. Slippage = 0.25. Reward +1 for transitioning to an odd state and -1 for an even state.

To further illustrate the effect of the environment on policies, Figure 2 illustrates the effect of the state transition dynamics on policies in the graph environment. Most importantly, this shows that the environment can greatly change

[1] For stochastic environments and/or stochastic policies, this step should be repeated many times to get accurate results

the effect of policies as with larger graph environment lengths, the transition probabilities and structure of the environment result in having a greater influence on the state-visitation difference than the policy itself. This leads to the flattening (constant state-visitation difference) of the curve at higher graph environment lengths.

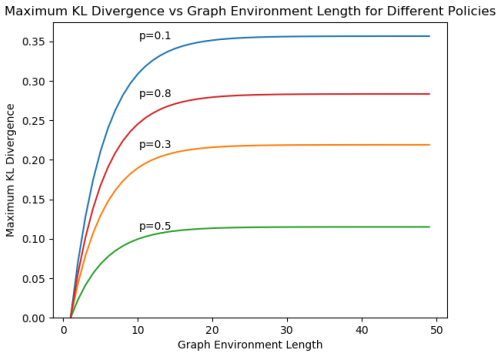

Figure 2: The change in maximum KL divergence vs. graph environment length for different policies (p)

Next, we select some behavior and evaluation policies and calculate the state-visitation difference for all combinations; thus covering the entire behavior and evaluation policy space. This results in the plot Figure 3, depicting the different combinations of behavior and evaluation policies with their respective state-visitation difference.

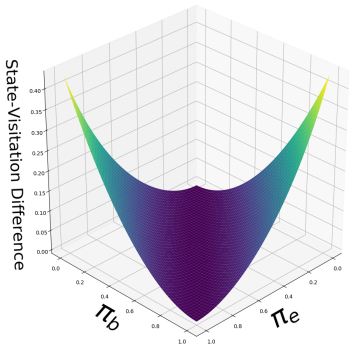

Figure 3: State-visitation difference for different combinations of π_b and π_e in the Graph environment with an horizon of 10. The policies are the probability of taking action: a = 0 and hold for all states (state independent policy)

Following this, we then evaluate the performance of SOPE [1] and other estimators (as a benchmark) with varying state-visitation difference to see its effect on the MSE, variance, and bias estimates.

5.1.1 Graph Environment Results

Figure 5 shows the results of weighted SOPE for different values of n. As mentioned before, n represents the trade-off between SIS and PDIS. When n is equal to 0, solely SIS is used. When n is equal to L (horizon length), only PDIS is used.

Generally the plot shows that n = 0 has a higher bias and a lower variance, which is what we would expect from SIS. On the other hand, n = L has a lower bias and a higher variance, which is also expected from PDIS. The intermediary values for n have biases and variances that lie in between the two, which illustrates the bias-variance tradeoff that SOPE is making.

Furthermore, we see a general trend of increasing MSE, bias and variance for all n with larger state-visitation difference. Again, this is expected because both estimators (SIS and PDIS) are negatively correlated with the increase in policy mismatch [13].

It is remarkable, however, that the variance of SIS (n = 0) remains more or less constant. This comes at the cost of a larger bias. Though for a very large state-visitation difference, SIS is able to outperform the other estimators as is visible in the MSE. This can be explained by the method used to approximate the estimated density ratios w. The current SOPE implementation uses the method from [9] to estimate the density ratios. In the paper, they use the Rao-Blackwellization method to reduce the variance of the density ratio estimate. This could be the reason why solely using SIS shows a much lower variance at the cost of a slight increase of bias.

In Figure 5 we picked the same π_b as proposed [1], but we would like to know if the results would be impacted when changing this. In Figure 6 we plotted the results for an estimator with n fixed to 10 for different behaviour policies. The original π_b is equal to 0.5. The main difference is in the range of the state-visitation difference; for 0.5 it is much shorter since evaluation policies can deviate less from it. Furthermore, the lines show a similar shape but not exactly. Apparently π_b is affecting the results even though we would expect a particular state-

visitation difference to encapsulate all the information about the state-visitation difference.

5.2 GridWorld

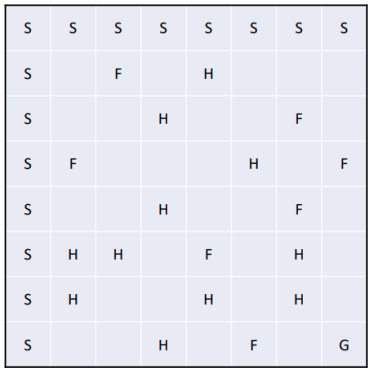

Figure 4: GridWorld Environment. Blank and 'S' spaces indicate a small negative reward of -0.01, 'S' are the possible starting states, 'F' indicates a reward of -0.005, and 'H' indicates a reward -0.5, 'G' is the goal state with reward +1. Slippage = 0.2

Continuing, we further conducted experiments on the gridworld environment from [13]. The gridworld allows introducing an additional level of uncertainty with a larger action space (4 possible actions) and state space, more variability in the reward structure, and the possibility of longer trajectory horizons i.e. that can create more complex policies. Further, gridworld is a standard and popular environment in the RL community. As the action and state space is larger, and the policy is dependent on the state, the experiment is modified accordingly to find two policies for some state-visitation difference. Specifying a probability for an action as done in the Graph experiment is not possible. Therefore, to set up random policies we use the ϵ-greedy approach and vary epsilon, also known as the probability of deviation to generate a number of policies with varying state-visitation difference. This allows us to specify a single value (the probability of deviation) to get different policies and works similarly to how the graph experiment is set up. Firstly the ϵ-greedy policy with $\epsilon = 0.001$ is used to find the best policy in the environment. We assume this is the best/optimal policy possible in the environment. To generate behavior and evaluation policies we make use of the optimal policy and add a probability of deviation (where 1 indicates maximum deviation, 0 indicates no deviation) to generate a different policy. As shown similarly in the graph environment, a plot of the deviation of the behavior and evaluation policies with respect to the ϵ-greedy policy, and its effect on the state-visitation difference is shown in Figure 9:

Figure 9 shows that the state-visitation difference changes similarly as shown for the Graph environment in Figure 3, but with a different magnitude in state-visitation difference. This increase in magnitude is expected as the state space is considerably larger than the Graph environment state space and allows for more varied policies and; thus more varied state visitations. After calculating the state-visitation difference for every behavior and evaluation policy pair, the SOPE estimator is run for various state-visitation difference steps.

5.2.1 Results GridWorld Environment

The results of the experiments in the GridWorld environment are shown in Figure 7 and Figure 8. Overall, the results presented follow the same trends as depicted by the results in the Graph environment. However, it is observed that the GridWorld results have a smaller magnitude of MSE, bias, and variance and concurrently a different relationship with the change in state-visitation difference. The smaller magnitude of error can be due to using 'higher' quality policies by the ϵ-greedy method. As all policies used in the experiment are derived from the best policy ($\epsilon = 0.001$) with some probability of deviation, all the policies generated still have a substantial state visitation in the trajectory of the best policy. For example, in Figure 4 the state-visitation distribution for a probability of deviation of 0.8 is shown. Even with a high deviation probability, the state visitation largely resembles the best policy. As the policies have higher state visitations in this region, this leads to more confident results of the distribution ratios in these regions; thus leading to lower errors in the estimator. Another notable difference is observed in the MSE of varying behavior policies in Figure 8. With $\pi_b = 0.1$ having the largest MSE with varying state-visitation difference. This demonstrates that having a good π_b does not translate to improved estimates. In fact, when π_b exhibits higher deviations, such as 0.5 and 0.9, the MSE is lower. This can be explained by the fact that the higher deviation π_b's

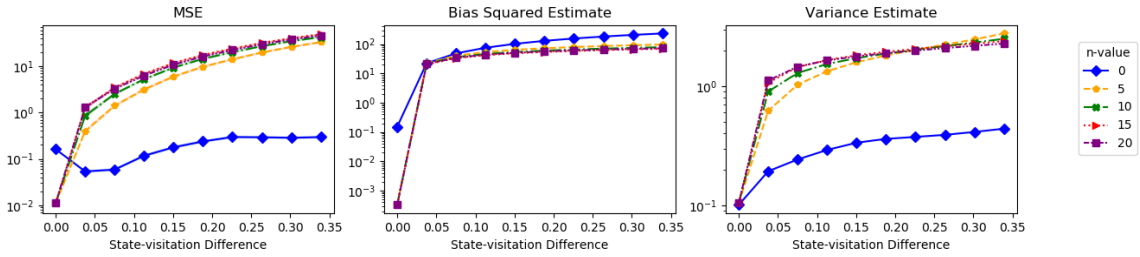

Figure 5: The results for the Graph environment under $\pi_b = 0.9$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

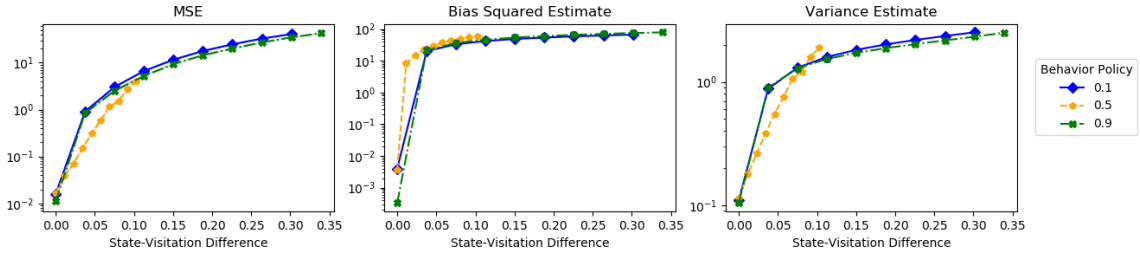

Figure 6: The results for the Graph environment with weighted SOPE n = 10. The plots represent different behaviour policies.

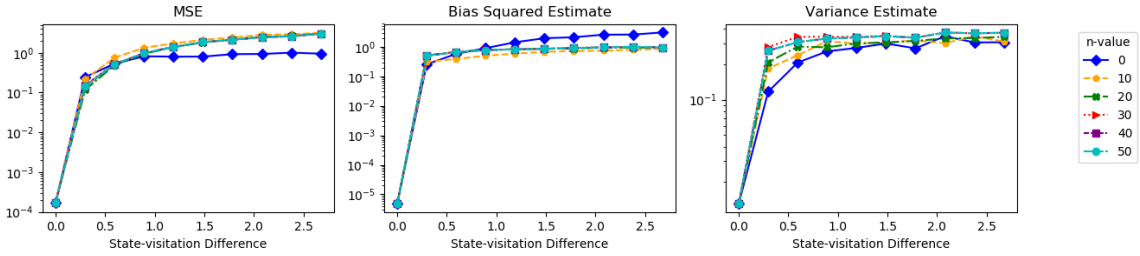

Figure 7: The results for the GridWorld environment under $\pi_b = 0.1$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS.

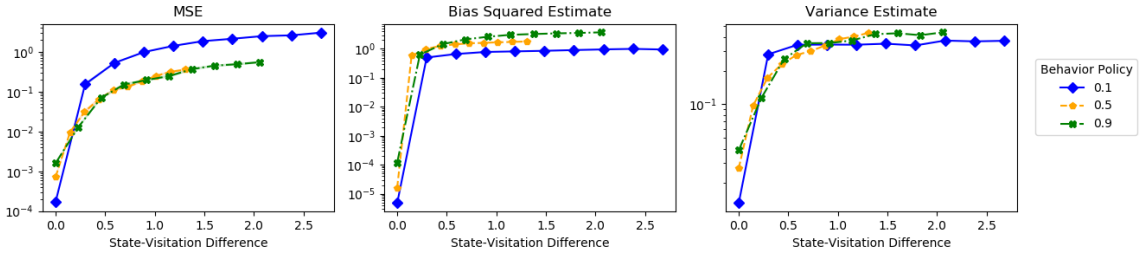

Figure 8: The results for the GridWorld environment with weighted SOPE n = 30. The plots represent different behaviour policies.

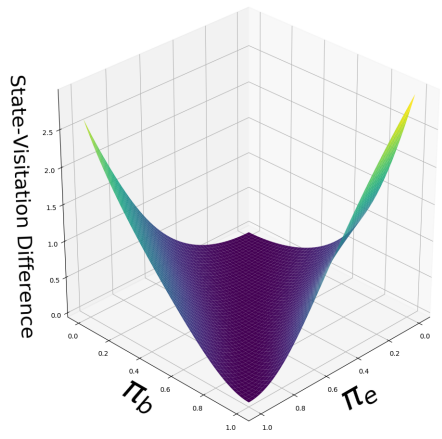

Figure 9: State-visitation difference for different values of deviation from ϵ-greedy policy in the GridWorld environment.

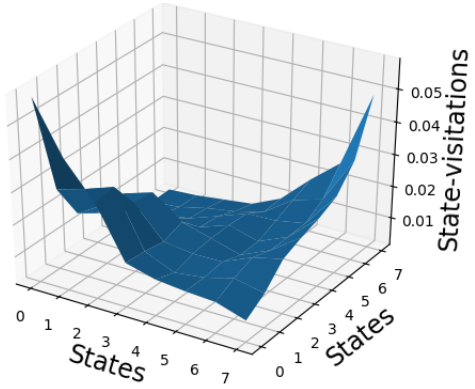

Figure 10: GridWorld State-visitation Frequency with a probability of deviation = 0.8. (0,0) refers to the top left in Figure 4

explore a greater number of states with a higher propensity, resulting in a more comprehensive understanding of the environment. Furthermore, the similarity in MSE between $\pi_b = 0.5$ and 0.9, suggests that there exists a saturation point where additional state-visitations do not necessarily improve the performance of estimators.

6 Conclusions

This paper studied to what extent the difference between the evaluation policy (π_e) and the behavior policy (π_b) affects the bias and variance of SOPE estimators. We quantified the difference between (π_e) and (π_b) by defining the state-visitation difference as the Kullback-Leibler divergence between the state-visitation distributions of the policies. This metric provided a systematic and intuitive way to compare the difference between state-visitation distributions, although it was not entirely robust to differing (π_b). Our results show that the choice of estimators is application specific. But generally, we showed that for a large state-visitation difference, SIS (stationary distribution-based importance sampling) performs best. Furthermore, estimators capable of estimating the distribution ratio accurately can perform significantly better.

However, it is crucial to acknowledge that the performance of current estimators relying on state visitation distributions is significantly influenced by the quality of the distribution estimators. Errors in the approximation of this ratio can cause estimators to perform worse than expected and finding accurate estimates of the distribution ratios is an important and currently active field of research.

Moreover, a limitation of our study is that our policies were not state-dependent. Even though in the Gridworld environment the policy depended on the state, the same constant probability of deviation was used. To make it truly state-dependant a varying deviation probability would be required. Further, the use of synthetic environments may not completely translate to the actual use cases of OPE, making it challenging to currently assure that our findings hold on more different and challenging environments.

Future work can include a comparison with more estimators and measuring the effect of what the distribution ratio estimators have on the OPE estimators. Further, in similar settings, state-dependent policies could be used to evaluate if this changes the performance of the estimators. More research towards 'hybrid' OPE estimators could be an interesting direction by integrating multiple currently existing estimators. Lastly, to enhance the computational efficiency in evaluating estimators in high-dimensional or continuous state/action spaces, it is necessary to revise the current approach presented. A possible solution would be to employ a function approximator to estimate the state-visitation difference. This would eliminate the need for an exhaustive search to find policies with a target state-visitation difference and allow to evaluate the robustness of the estimators

in high dimensional environments.

References

[1] Christina J. Yuan, Yash Chandak, Stephen Giguere, Philip S. Thomas, and Scott Niekum. Sope: Spectrum of off-policy estimators. *arXiv*, 2021.

[2] Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation, 2017.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[4] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *CoRR*, abs/1906.04733, 2019.

[5] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. *International Conference on Machine Learning*, 2016.

[6] Judea Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.

[7] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[8] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[9] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *arXiv*, 2018.

[10] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.

[11] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction, Second Edition*. The MIT Press, 2018.

[12] Philip S. Thomas, Bruno Castro Da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 529:999–1004, 2019.

[13] Cameron Voloshin, Hoang Minh Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *CoRR*, abs/1911.06854, 2019.

A Additional Results Graph Environment

A.1 Weighted SOPE

A.1.1 Different n-values

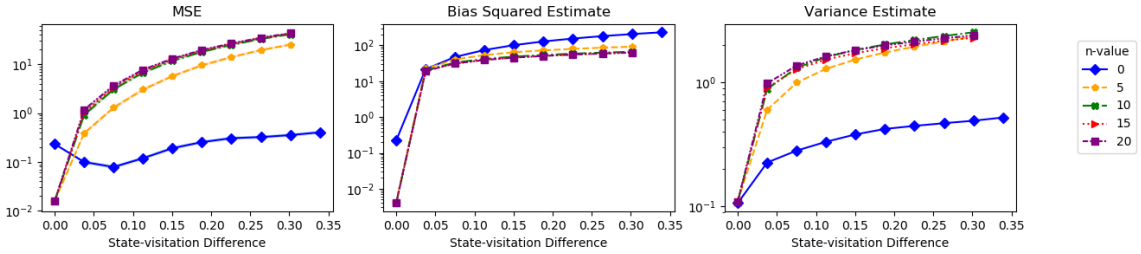

Figure 11: The results for the Graph environment under $\pi_b = 0.1$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

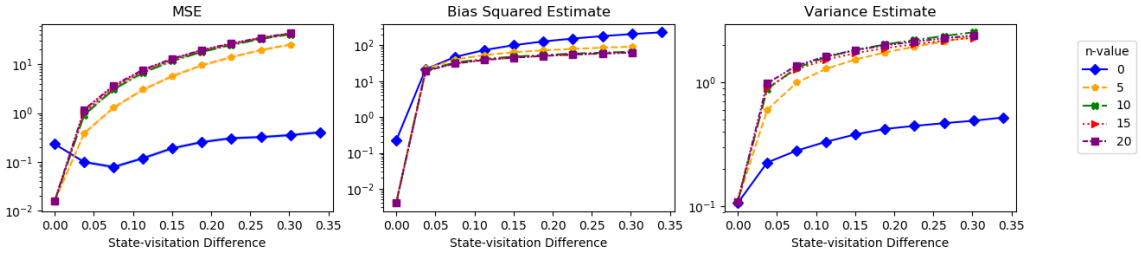

Figure 12: The results for the Graph environment under $\pi_b = 0.5$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

A.1.2 Different behaviour policies

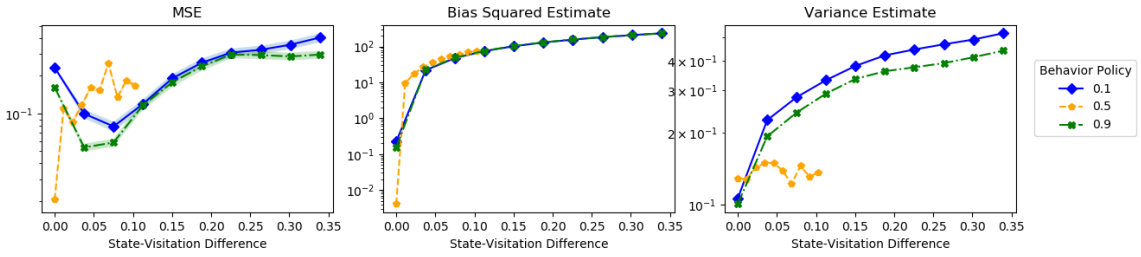

Figure 13: The results for the Graph environment with weighted SOPE n = 0. The plots represent different behaviour policies.

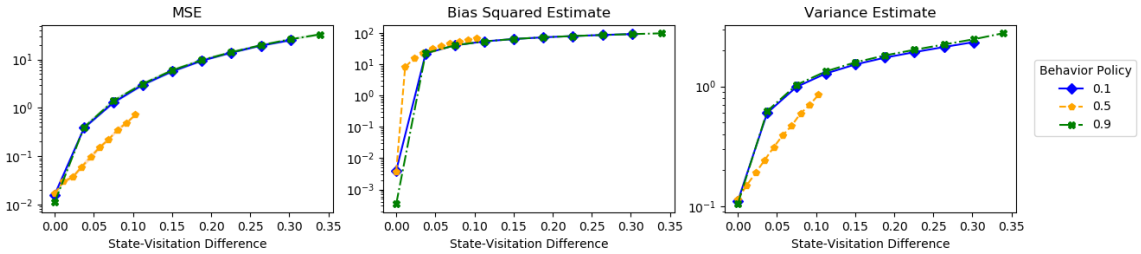

Figure 14: The results for the Graph environment with weighted SOPE n = 5. The plots represent different behaviour policies.

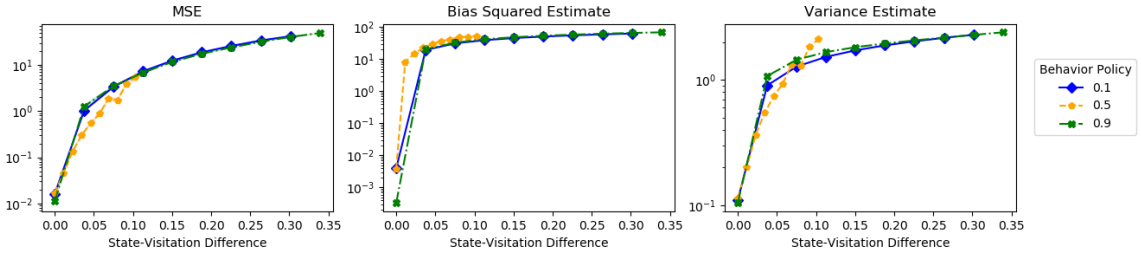

Figure 15: The results for the Graph environment with weighted SOPE n = 15. The plots represent different behaviour policies.

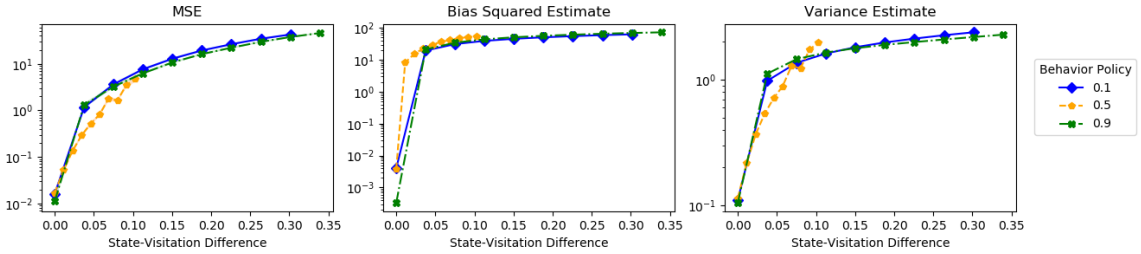

Figure 16: The results for the Graph environment with weighted SOPE n = 20. The plots represent different behaviour policies.

A.2 Unweighted SOPE

A.2.1 Different n-values

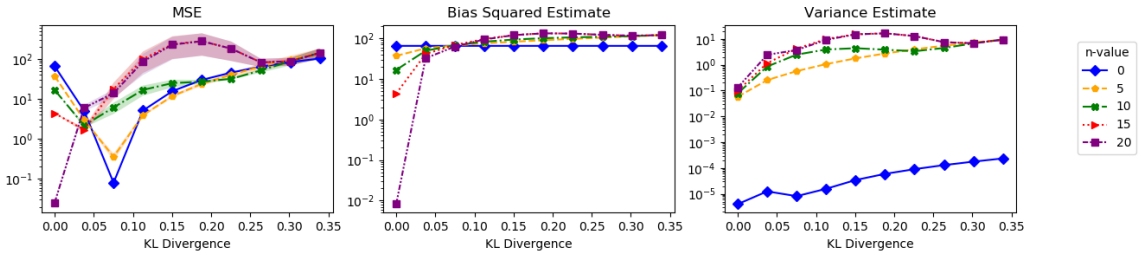

Figure 17: The results for the Graph environment under $\pi_b = 0.1$. The plots represent different n-values in unweighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

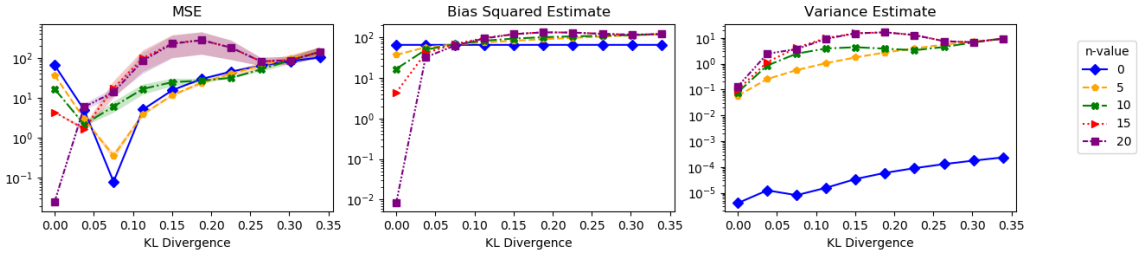

Figure 18: The results for the Graph environment under $\pi_b = 0.5$. The plots represent different n-values in unweighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

A.2.2 Different behaviour policies

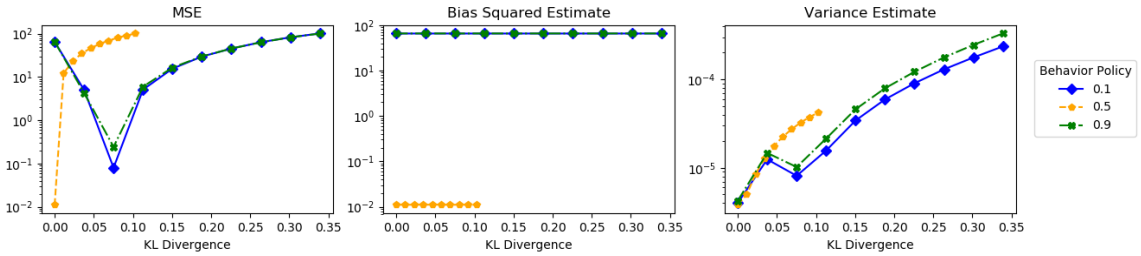

Figure 19: The results for the Graph environment with unweighted SOPE n = 0. The plots represent different behaviour policies.

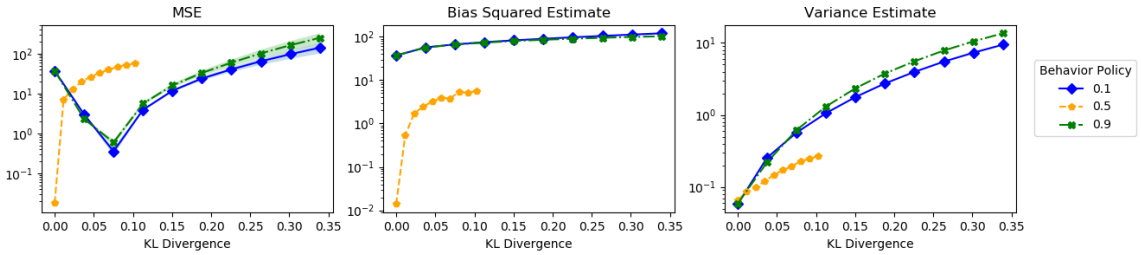

Figure 20: The results for the Graph environment with unweighted SOPE n = 5. The plots represent different behaviour policies.

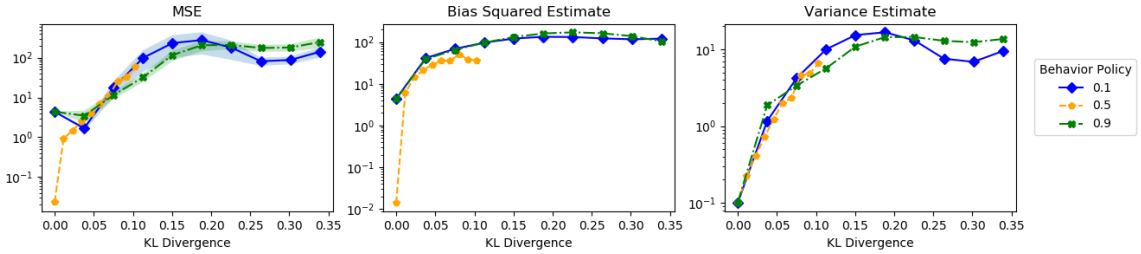

Figure 21: The results for the Graph environment with unweighted SOPE n = 15. The plots represent different behaviour policies.

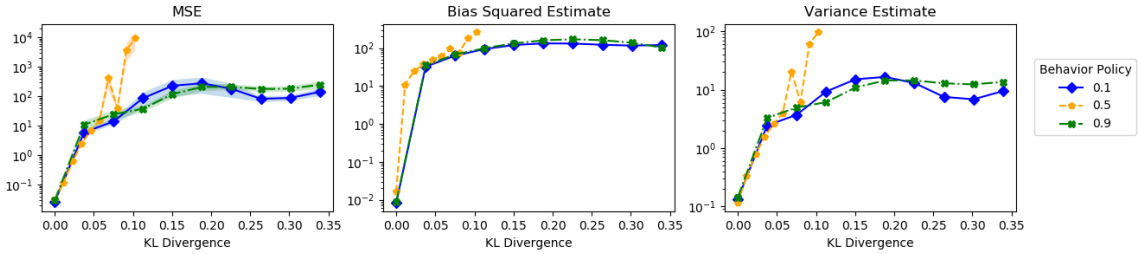

Figure 22: The results for the Graph environment with unweighted SOPE n = 20. The plots represent different behaviour policies.

B Additional Results GridWorld environment

B.1 Weighted SOPE

B.1.1 Different n-values

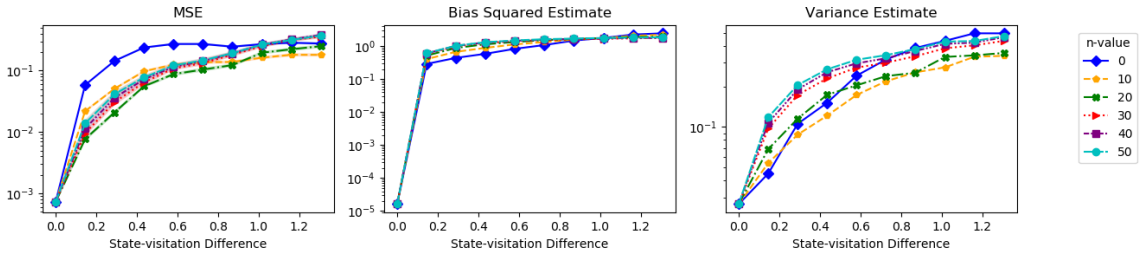

Figure 23: The results for the GridWorld environment under $\pi_b = 0.5$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

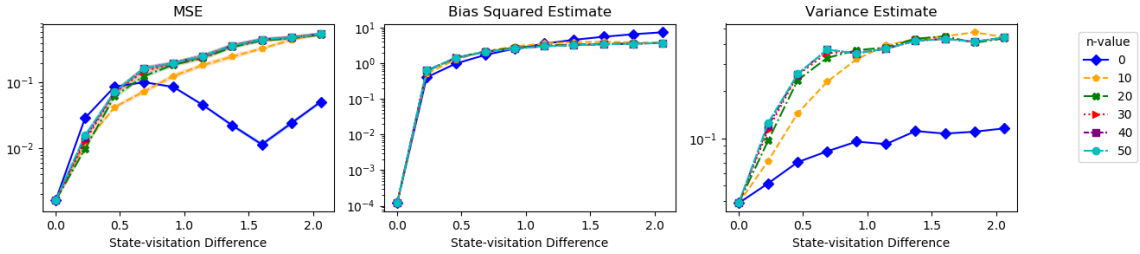

Figure 24: The results for the GridWorld environment under $\pi_b = 0.9$. The plots represent different n-values in weighted SOPE, where n = 0 is SIS and n = 20 is PDIS.

B.1.2 Different behavior policies

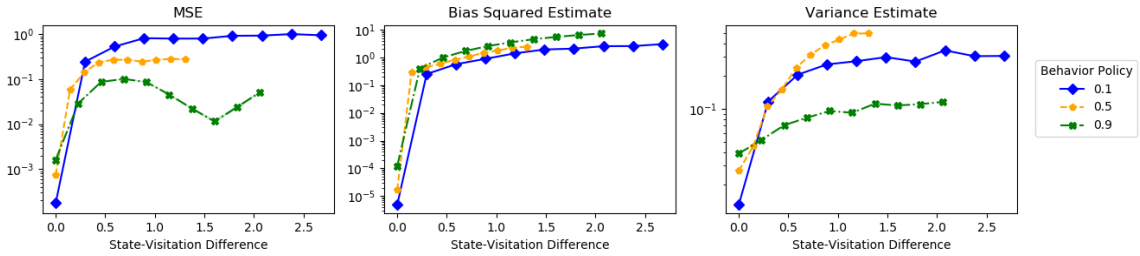

Figure 25: The results for the GridWorld environment with weighted SOPE n = 0. The plots represent different behaviour policies.

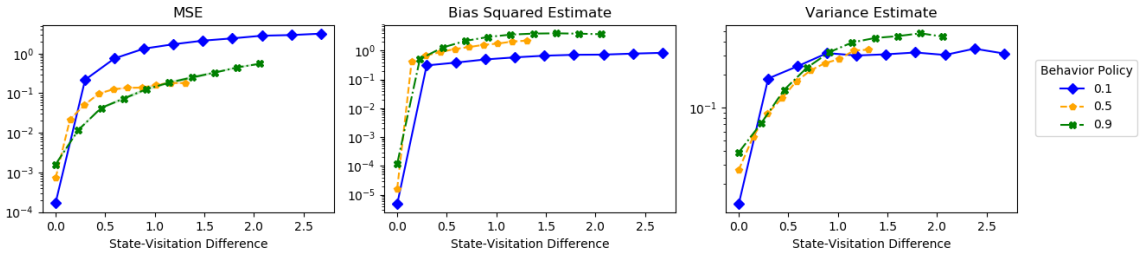

Figure 26: The results for the GridWorld environment with weighted SOPE n = 10. The plots represent different behaviour policies.

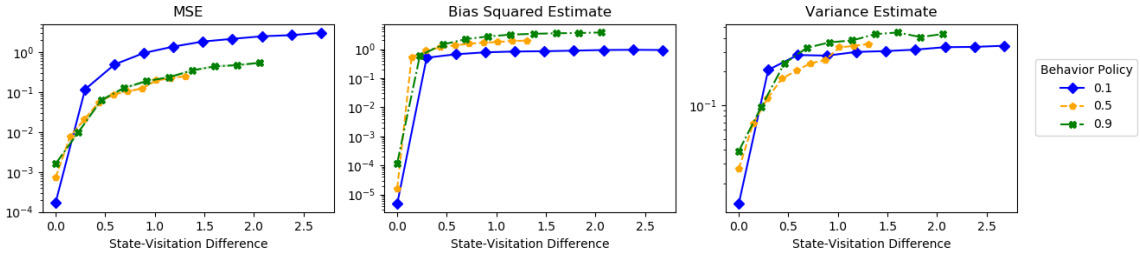

Figure 27: The results for the GridWorld environment with weighted SOPE n = 20. The plots represent different behaviour policies.

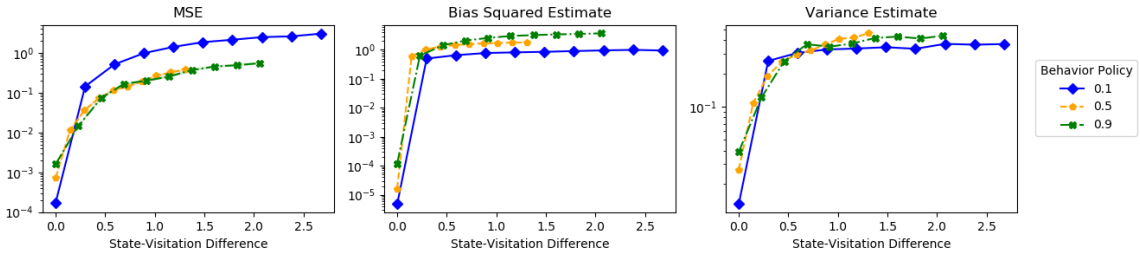

Figure 28: The results for the GridWorld environment with weighted SOPE n = 40. The plots represent different behaviour policies.

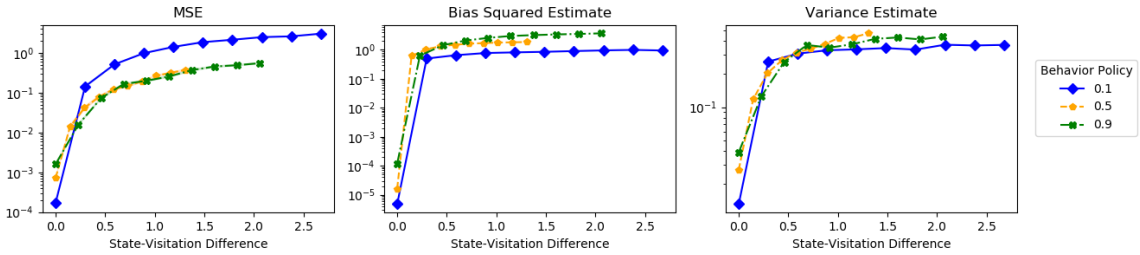

Figure 29: The results for the GridWorld environment with weighted SOPE n = 50. The plots represent different behaviour policies.