Importance Sampling Policy Evaluation with an Estimated Behavior Policy

Josiah P. Hanna [1] **Scott Niekum** [1] **Peter Stone** [1]

Abstract

We consider the problem of off-policy evaluation in Markov decision processes. Off-policy evaluation is the task of evaluating the expected return of one policy with data generated by a different, *behavior* policy. Importance sampling is a technique for off-policy evaluation that re-weights off-policy returns to account for differences in the likelihood of the returns between the two policies. In this paper, we study importance sampling with an estimated behavior policy where the behavior policy estimate comes from the same set of data used to compute the importance sampling estimate. We find that this estimator often lowers the mean squared error of off-policy evaluation compared to importance sampling with the true behavior policy or using a behavior policy that is estimated from a separate data set. Intuitively, estimating the behavior policy in this way corrects for error due to sampling in the action-space. Our empirical results also extend to other popular variants of importance sampling.[1]

1. Introduction

Sequential decision-making tasks, such as a robot manipulating objects or an autonomous vehicle deciding when to change lanes, are ubiquitous in artificial intelligence. For these tasks, *reinforcement learning* (RL) algorithms provide a promising alternative to hand-coded skills, allowing sequential decision-making agents to acquire policies autonomously given only a reward function measuring task performance (Sutton & Barto, 1998). When applying RL to real world problems, an important problem that often comes up is *policy evaluation*. In policy evaluation, the goal is to determine the expected return – sum of rewards – that an *evaluation policy*, π_e, will obtain when deployed on the task of interest.

[1]The University of Texas at Austin, Austin, Texas, USA. Correspondence to: Josiah P. Hanna <jphanna@cs.utexas.edu>.

Real-world Sequential Decision Making workshop at ICML 2019. Copyright 2019 by the author(s).

[1]An extended version of this work has been accepted for presentation at ICML 2019 (Hanna et al., 2019).

In *off-policy* policy evaluation, we are given data (in the form of state-action-reward trajectories) generated by a second *behavior policy*, π_b. We then use these trajectories to evaluate π_e. Accurate off-policy policy evaluation is especially important when we want to know the value of a policy before it is deployed in the real world or have many policies to evaluate and want to avoid running each one individually. *Importance sampling* addresses this problem by re-weighting returns generated by π_b such that they are unbiased estimates of π_e (Precup et al., 2000). While the basic importance sampling estimator is often noted in the literature to suffer from high variance, more recent importance sampling estimators have lowered this variance (Thomas & Brunskill, 2016; Jiang & Li, 2016). Regardless of additional variance reduction techniques, all importance sampling variants compute the likelihood ratio $\frac{\pi_e(a|s)}{\pi_b(a|s)}$ for all state-action pairs in the off-policy data.

In this paper, we propose to replace $\pi_b(a|s)$ with its empirical estimate – that is, we replace the probability of sampling an action in a particular state with the frequency at which that action actually occurred in that state in the data. It is natural to assume that such an estimator will yield worse performance since it replaces a known quantity with an estimated quantity. However, research in the multi-armed bandit (Li et al., 2015; Narita et al., 2019), causal inference (Hirano et al., 2003; Rosenbaum, 1987), and Monte Carlo integration (Henmi et al., 2007; Delyon & Portier, 2016) literature has demonstrated that estimating the behavior policy can *improve* the mean squared error of importance sampling policy evaluation. Motivated by these results, we study the performance of such methods for policy evaluation in full Markov decision processes.

Specifically, we study an estimator that, given a dataset, $\mathcal{D}$, of trajectories, use $\mathcal{D}$ both to estimate the behavior policy and then to compute the importance sampling estimate. Though related to methods in the statistics literature, the so-called regression importance sampling method is specific to Markov decision processes where actions taken at one time-step influence the states and rewards at future time-steps. We show empirically that regression importance sampling *lowers* the mean squared error of importance sampling off-policy evaluation.

2. Preliminaries

This section formalizes our problem and introduces importance sampling off-policy evaluation.

2.1. Notation

We assume the environment is a finite horizon, episodic *Markov decision process* with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probabilities, P, reward function R, horizon L, discount factor γ, and initial state distribution d_0 (Puterman, 2014). A *Markovian* policy, π, is a function mapping the current state to a probability distribution over actions; a policy is *non-Markovian* if its action distribution is conditioned on past states or actions. For simplicity, we assume that $\mathcal{S}$ and $\mathcal{A}$, are finite and that probability distributions are probability mass functions.[2] Let $H \coloneqq (S_0, A_0, R_0, S_1, \ldots, S_{L-1}, A_{L-1}, R_{L-1})$ be a *trajectory*, $g(H) \coloneqq \sum_{t=0}^{L-1} \gamma^t R_t$ be the *discounted return* of trajectory H, and $v(\pi) \coloneqq \mathbf{E}[g(H)|H \sim \pi]$ be the expected discounted return when the policy π is used starting from state S_0 sampled from the initial state distribution. We assume that the transition and reward functions are unknown and that the episode length, L, is a finite constant.

In off-policy policy evaluation, we are given a fixed *evaluation policy*, π_e, and a data set of m trajectories and the policies that generated them: $\mathcal{D} \coloneqq \{H_i, {\pi_b}^{(i)}\}_{i=1}^m$ where $H_i \sim {\pi_b}^{(i)}$. We assume that $\forall\{H_i, {\pi_b}^{(i)}\} \in \mathcal{D}$, ${\pi_b}^{(i)}$ is Markovian i.e., actions in $\mathcal{D}$ are independent of past states and actions given the immediate preceding state. Our goal is to design an off-policy estimator, OPE, that takes $\mathcal{D}$ and estimates $v(\pi_e)$ with minimal mean squared error (MSE). Formally, we wish to minimize $\mathbf{E}_{\mathcal{D}}[(\mathrm{OPE}(\pi_e, \mathcal{D}) - v(\pi_e))^2]$.

2.2. Importance Sampling

Importance Sampling (IS) is a method for reweighting returns generated by a *behavior* policy, π_b, such that they are unbiased returns from the *evaluation* policy. Given a set of m trajectories and the policy that generated each trajectory, the IS off-policy estimate of $v(\pi_e)$ is:

$$\mathrm{IS}(\pi_e, \mathcal{D}) \coloneqq \frac{1}{m} \sum_{i=1}^{m} g(H^{(i)}) \prod_{t=0}^{L-1} \frac{\pi_e(A_t^{(i)}|S_t^{(i)})}{{\pi_b}^{(i)}(A_t^{(i)}|S_t^{(i)})}. \quad (1)$$

We refer to (1) – that uses the true behavior policy – as the ordinary importance sampling (OIS) estimator and refer to $\frac{\pi_e(A|S)}{\pi_b(A|S)}$ as the OIS weight for action A in state S.

The importance sampling estimator with OIS weights can be understood as a Monte Carlo estimate of $v(\pi_e)$ with a correction for the distribution shift caused by sampling trajectories from π_b instead of π_e. As more data is obtained, the empirical frequency of any trajectory approaches the expected frequency under π_b and then the OIS weight corrects the weighting of each trajectory to reflect the expected frequency under π_e.

[2] Unless otherwise noted, all results and discussion apply equally to the discrete and continuous setting.

3. Sampling Error in Importance Sampling

The ordinary importance sampling estimator (1) is known to have high variance. A number of importance sampling variants have been proposed to address this problem, however, all such variants use the OIS weight. The common reliance on OIS weights suggest that an implicit assumption in the RL community is that OIS weights lead to the most accurate estimate. Hence, when an application requires estimating an unknown π_b in order to compute importance weights, the application is implicitly assumed to only be approximating the desired weights.

However, OIS weights themselves are sub-optimal in at least one respect: the weight of each trajectory in the OIS estimate is inaccurate unless we happen to observe each trajectory according to its true probability. When the empirical frequency of any trajectory is unequal to its expected frequency under π_b, the OIS estimator puts either too much or too little weight on the trajectory. We refer to error due to some trajectories being either over- or under-represented in $\mathcal{D}$ as *sampling error*. Sampling error may be unavoidable when we desire an unbiased estimate of $v(\pi_e)$. However, correcting for it by properly weighting trajectories will, in principle, give us a lower mean squared error estimate.

The problem of sampling error is related to a Bayesian objection to Monte Carlo integration techniques: OIS ignores information about the closeness of trajectories in $\mathcal{D}$ (O'Hagan, 1987; Ghahramani & Rasmussen, 2003). This objection is easiest to understand in deterministic and discrete environments though it also holds for stochastic and continuous environments. In a deterministic environment, additional samples of any trajectory, h, provide no new information about $v(\pi_e)$ since only a single sample of h is required to know $g(h)$. However, the more times a particular trajectory appears, the more weight it receives in an OIS estimate even though the correct weighting of $g(h)$, $\Pr(h|\pi_e)$, is known since π_e is known. In stochastic environments, it is reasonable to give more weight to recurring trajectories since the recurrence provides additional information about the unknown state-transition and reward probabilities. However, ordinary importance sampling also relies on sampling to approximate the known policy probabilities.

Finally, we note that the problem of sampling error applies to any variant of importance sampling using OIS weights, e.g., weighted importance sampling (Precup et al., 2000), per-decision importance sampling (Precup et al., 2000), the doubly robust estimator (Jiang & Li, 2016; Thomas & Brun-

skill, 2016), and the MAGIC estimator (Thomas & Brunskill, 2016). Sampling error is also a problem for on-policy Monte Carlo policy evaluation since Monte Carlo is the special case of OIS when the behavior policy is the same as the evaluation policy.

4. Regression Importance Sampling

In this section we introduce the primary focus of our work: an estimator called regression importance sampling (RIS) that correct for sampling error in $\mathcal{D}$ by importance sampling with an estimated behavior policy. The motivation for this approach is that, though $\mathcal{D}$ was sampled with π_b, the trajectories in $\mathcal{D}$ may appear as if they had been generated by a different policy, $\pi_{\mathcal{D}}$. For example, if π_b would choose between two actions with equal probability in a particular state, the data might show that one action was selected more often than the other in that state. Thus instead of using OIS to correct from π_b to π_e, we introduce RIS that corrects from $\pi_{\mathcal{D}}$ to π_e.

We assume that, in addition to $\mathcal{D}$, we are given a policy class – a set of policies – Π where each $\pi \in \Pi$ is a distribution over actions conditioned on states: $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. The RIS estimator first estimates the maximum likelihood behavior policy in Π given $\mathcal{D}$:

$$\pi_{\mathcal{D}} := \operatorname*{argmax}_{\pi \in \Pi} \sum_{H \in \mathcal{D}} \sum_{t=0}^{L-1} \log \pi(a_t|s_t). \tag{2}$$

The RIS estimate is then the importance sampling estimate with $\pi_{\mathcal{D}}$ replacing π_b:

$$\text{RIS}(\pi_e, \mathcal{D}) := \frac{1}{m} \sum_{i=1}^{m} g(H_i) \prod_{t=0}^{L-1} \frac{\pi_e(A_t|S_t)}{\pi_{\mathcal{D}}(A_t|S_t)}$$

Analogously to OIS, we refer to $\frac{\pi_e(A_t|S_t)}{\pi_{\mathcal{D}}(S_t|A_t)}$ as the RIS weight for action A_t and state S_t. Note that the RIS weights are always well-defined since $\pi_{\mathcal{D}}$ never places zero probability mass on any action that occurred in $\mathcal{D}$.

Intuitively, RIS corrects for sampling error by down-weighting actions that were sampled more frequently than they would be in expectation and up-weighting actions that were sampled less frequently than they would be in expectation.

5. Empirical Results

We present an empirical study of the RIS estimator across several policy evaluation tasks. Our experiments are designed to answer the following questions:

1. What is the empirical effect of replacing OIS weights with RIS weights in sequential decision making tasks?
2. How important is using $\mathcal{D}$ to both estimate the behavior policy and compute the importance sampling estimate?

With non-linear function approximation, our results suggest that the standard supervised learning approach of model selection using hold-out validation loss may be sub-optimal for the regression importance sampling estimator. Thus, we also investigate the question:

4. Does minimizing hold-out validation loss set yield the minimal MSE regression importance sampling estimator when estimating $\pi_{\mathcal{D}}$ with gradient descent and neural network function approximation?

5.1. Empirical Set-up

We run policy evaluation experiments in several domains. We provide a short description of each domain here.

- **Linear Dynamical System:** This domain is a point-mass agent moving towards a goal in a two dimensional world by setting x and y acceleration. Policies are linear in a second order polynomial transform of the state features. We estimate $\pi_{\mathcal{D}}$ with least squares.
- **Simulated Robotics:** We also use two continuous control tasks from the OpenAI gym: Hopper and HalfCheetah.[3] In each task, we use neural network policies with 2 layers of 64 $\tanh$ hidden units each for π_e and π_b.

5.2. Empirical Results

We now present our empirical results.

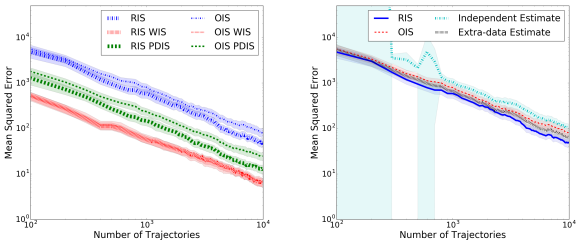
RIS with Linear Function Approximation Our next set of experiments consider continuous state and action spaces in the Linear Dynamical System domain. RIS represents $\pi_{\mathcal{D}}$ as a Gaussian policy with mean given as a linear function of the state features. We compare three variants of IS, each implemented with RIS and OIS weights: the ordinary IS estimator, weighted IS (WIS), and per-decison IS (PDIS) (Precup et al., 2000). Each method is averaged over 200 trials and results are shown in Figure 1(a).

We see that RIS weights improve both IS and PDIS, while both WIS variants have similar MSE. This result suggests that the MSE improvement from using RIS weights depends, at least partially, on the variant of IS being used.

We also evaluate alternative data sources for estimating $\pi_{\mathcal{D}}$ in order to establish the importance of using $\mathcal{D}$ to both estimate $\pi_{\mathcal{D}}$ and compute the value estimate. Specifically, we consider:

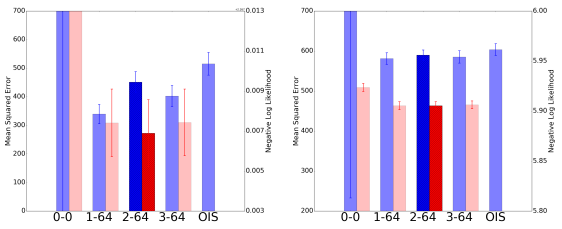
1. **Independent Estimate**: In addition to $\mathcal{D}$, this method has access to an additional set, $\mathcal{D}_{\texttt{train}}$. The behavior

[3] For these tasks we use the Roboschool versions: https://github.com/openai/roboschool



(a) LDS (b) LDS Alt. Weights

Figure 1: Linear dynamical system results. Figure 1(a) shows the mean squared error (MSE) for three IS variants with and without RIS weights. Figure 1(b) shows the MSE for different methods of estimating the behavior policy compared to RIS and OIS. Axes are log-scaled.



(a) Hopper (b) HalfCheetah

Figure 2: Figures 2(a) and 2(b) compare different neural network architectures (specified as #-layers-#-units) for regression importance sampling on the Hopper and HalfCheetah domain. The darker, blue bars give the MSE for each architecture and OIS. Lighter, red bars give the negative log likelihood of a hold-out data set. Our main point of comparison is the MSE of the architecture with the lowest hold-out negative log likelihood (given by the darker pair of bars) compared to the MSE of IS.

policy is estimated with $\mathcal{D}_{\texttt{train}}$ and the policy value estimate is computed with $\mathcal{D}$.

2. **Extra-data Estimate**: This baseline is the same as **Independent Estimate** except it uses both $\mathcal{D}_{\texttt{train}}$ and $\mathcal{D}$ to estimate π_b. Only $\mathcal{D}$ is used to compute the policy value estimate.

Independent Estimate gives high variance estimates for small sample sizes but then approaches OIS as the sample size grows. **Extra-Data Estimate** corrects for some sampling error and has lower MSE than OIS. RIS lowers MSE compared to all baselines.

RIS with Neural Networks Our remaining experiments use the Hopper and HalfCheetah domains. RIS represents $\pi_{\mathcal{D}}$ as a neural network that maps the state to the mean of a Gaussian distribution over actions. The standard deviation of the Gaussian is given by state-independent parameters. In these experiments, we sample a single batch of 400 trajectories and compare the MSE of RIS and IS on this batch. We repeat this experiment 200 times for each method.

Figure 2 compares the MSE of RIS for different neural network architectures. Our main point of comparison is RIS using the architecture that achieves the lowest validation error during training (the darker bars in Figure 2). Under this comparison, the MSE of RIS with a two hidden layer network is lower than that of OIS in both Hopper and HalfCheetah though, in HalfCheetah, the difference is statistically insignificant. We also observe that the policy class with the best validation error does *not* always give the lowest MSE (e.g., in Hopper, the two hidden layer network gives the lowest validation loss but the network with a single layer of hidden units has $\approx 25\%$ less MSE than the two hidden layer network). This last observation motivates our final experiment.

RIS Model Selection Our final experiment aims to better understand how hold-out validation error relates to the MSE of the RIS estimator when using gradient descent to estimate neural network approximations of $\pi_{\mathcal{D}}$. This experiment duplicates our previous experiment, except every 25 steps of gradient descent we stop optimizing $\pi_{\mathcal{D}}$ and compute the RIS estimate with the current $\pi_{\mathcal{D}}$ and its MSE. We also compute the training and hold-out validation negative log-likelihood. Plotting these values gives a picture of how the MSE of RIS changes as our estimate of $\pi_{\mathcal{D}}$ changes. Figure 3 shows this plot for the Hopper domain.

We see that the policy with minimal MSE and the policy that minimizes validation loss are misaligned. If training is stopped when the validation loss is minimized, the MSE of RIS is lower than that of OIS (the intersection of the RIS curve and the vertical dashed line in Figure 3. However, the $\pi_{\mathcal{D}}$ that minimizes the validation loss curve is *not* identical to the $\pi_{\mathcal{D}}$ that minimizes MSE.

To understand this result, we also plot the average RIS estimate throughout behavior policy learning (bottom of Figure 3). We can see that at the beginning of training, RIS tends to *over-estimate* $v(\pi_e)$ because the probabilities given by $\pi_{\mathcal{D}}$ to the observed data will be small (and thus the RIS weights are large). As the likelihood of $\mathcal{D}$ under $\pi_{\mathcal{D}}$ increases (negative log likelihood decreases), the RIS weights become smaller and the estimates tend to *under-estimate* $v(\pi_e)$. The implication of these observations, for RIS, is that during behavior policy estimation the RIS estimate will likely have zero MSE at some point. Thus, there may be an early stopping criterion – besides minimal validation loss – that would lead to lower MSE with RIS, however, to date we have not found one. Note that OIS also tends to under-estimate policy value in MDPs as has been previously analyzed by Doroudi et al. (2017). We make the same

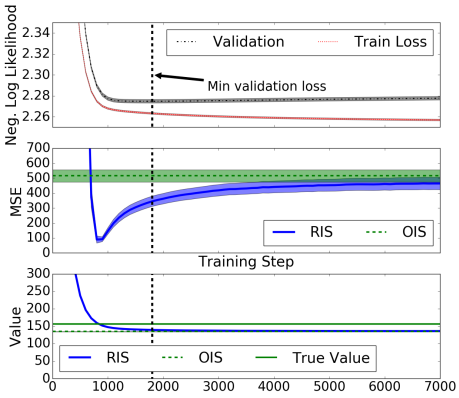


Figure 3: Mean squared error and estimate of the importance sampling estimator during training of $\pi_{\mathcal{D}}$. The x-axis is the number of gradient descent steps. The top plot shows the training and validation loss curves. The y-axis of the top plot is the average negative log-likelihood. The y-axis of the middle plot is mean squared error (MSE). The y-axis of the bottom plot is the value of the estimate. MSE is minimized close to, but slightly before, the point where the validation and training loss curves indicate that overfitting is beginning. This point corresponds to where the RIS estimate transitions from over-estimating to under-estimating.

observations in the HalfCheetah domain though we do not include the resulting figures.

6. Related Work

In this section we survey work related to behavior policy estimation for importance sampling. Methods related to RIS have been studied for Monte Carlo integration (Henmi et al., 2007; Delyon & Portier, 2016) and causal inference (Hirano et al., 2003; Rosenbaum, 1987). The REG method (discussed below) can be seen as the direct extension of these methods to MDPs. In contrast to these works, we study policy evaluation in Markov decision processes which introduces sequential structure into the samples and unknown stochasticity in the state transitions. These methods have also, to the best of our knowledge, *not* been studied in Markov decision processes or for sequential data.

Li et al. (2015) study the *regression* (REG) estimator for off-policy evaluation and show that its minimax MSE is asymptotically optimal though it might perform poorly for small sample sizes. Though REG and RIS are equivalent for multi-armed bandit problems, for MDPs, the definition of REG and RIS diverge. Intuitively, REG corrects for sampling error in both the action selection and state transitions through knowledge of the true state-transition function. However, such knowledge is usually unavailable and, in these cases, REG is inapplicable.

Narita et al. (2019) study behavior policy estimation for policy evaluation and improvement in multi-armed bandit problems. Their results are only for the bandit setting.

In the contextual bandit literature, Dudik et al. (2011) present finite sample bias and variance results for importance sampling that is applicable when the behavior policy probabilities are different than the true behavior policy. Farajtabar et al. (2018) extended these results to full MDPs. These works make the assumption that $\pi_{\mathcal{D}}$ is estimated independently from the data used in the final IS evaluation. In contrast, RIS uses the same set of data to both estimate π_b and compute the IS evaluation. This choice allows RIS to correct for sampling error and improve upon the OIS estimate (as shown in Figure 1(b)).

7. Discussion and Future Work

Our experiments demonstrate that regression importance sampling can obtain lower mean squared error than ordinary importance sampling for off-policy evaluation in Markov decision process environments. The main practical conclusion of our paper is the importance of estimating $\pi_{\mathcal{D}}$ with the same data used to compute the importance sampling estimate.

In this paper we focused on *batch* policy evaluation where $\mathcal{D}$ is given and fixed. Studying RIS for *online* policy evaluation setting is an interesting direction for future work. Finally, incorporating RIS into policy improvement methods is an interesting direction for future work. In work parallel to our own, two of the authors (Hanna & Stone, 2019) explored using an estimated behavior policy to lower sampling error in on-policy policy gradient learning. However, our approach in that paper only focuses on reducing variance in the one-step action selection while RIS could lower variance in the full return estimation.

8. Conclusion

We have studied a class of off-policy evaluation importance sampling methods, called regression importance sampling methods, that apply importance sampling after first estimating the behavior policy that generated the data. Notably, RIS estimates the behavior policy from the same set of data that is also used for the IS estimate. Computing the behavior policy estimate and IS estimate from the same set of data allows RIS to correct for the sampling error inherent to importance sampling with the true behavior policy. We evaluated RIS across several policy evaluation tasks and show that it improves over ordinary importance sampling – that uses the true behavior policy – in several off-policy policy evaluation tasks. Finally, we showed that, as the sample size grows, it can be beneficial to ignore knowledge that the true behavior policy is Markovian.

References

Delyon, B. and Portier, F. Integral approximation by kernel smoothing. *Bernoulli*, 22(4):2177–2208, 2016.

Doroudi, S., Thomas, P. S., and Brunskill, E. Importance sampling for fair policy selection. In *Uncertainty in Artificial Intelligence (UAI)*, 2017.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 1097–1104. Omnipress, 2011.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Ghahramani, Z. and Rasmussen, C. E. Bayesian monte carlo. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 505–512, 2003.

Hanna, J. and Stone, P. Reducing sampling error in the monte carlo policy gradient estimator. In *Proceedings of the 19th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2019.

Hanna, J., Niekum, S., Stone, P., and Niekum, S. Data-efficient policy evaluation through behavior policy search. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Henmi, M., Yoshida, R., and Eguchi, S. Importance sampling via the estimated sampler. *Biometrika*, 94(4):985–991, 2007.

Hirano, K., Imbens, G. W., and Ridder, G. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

Jiang, N. and Li, L. Doubly robust off-policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

Li, L., Munos, R., and Szepesvári, C. Toward minimax off-policy value estimation. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Narita, Y., Yasui, S., and Yata, K. Efficient counterfactual learning from bandit feedback. In *The 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

O'Hagan, A. Monte carlo is fundamentally unsound. *The Statistician*, pp. 247–249, 1987.

Precup, D., Sutton, R. S., and Singh, S. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pp. 759–766, 2000.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Rosenbaum, P. R. Model-based direct adjustment. *Journal of the American Statistical Association*, 82(398):387–394, 1987.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Thomas, P. S. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.