6.246 Reinforcement Learning: Foundations and Methods April 29, 2021

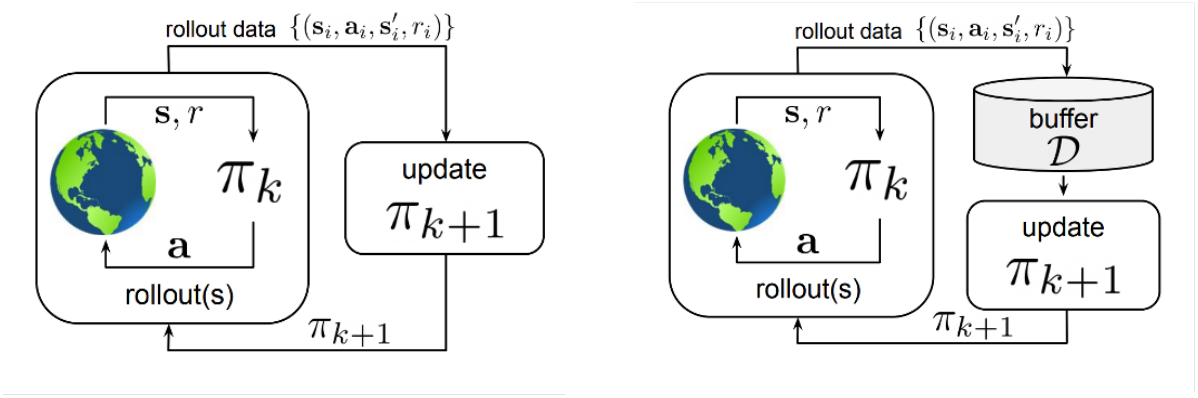
Lecture 19: Off-Policy Learning

Instructor: Cathy Wu *Scribe: Abdullah Alomar & Satvat Jagwani*

Note: the lecture notes have not been thoroughly checked for errors and are not at the level of publication.

1 Introduction

In off-Policy reinforcement learning (RL), algorithms evaluate and improve a target policy that is different from the observational policy used to generate the data. This is in contrast with on-policy RL, where a policy π_k is updated via data collected by π_k itself. In a broader sense, classical off-policy algorithms uses all past experience collected by an agent through its interaction with the environment using different policies. This experience is typically appended to a data buffer (also known as a replay buffer) $\mathcal{D}$. This collected dataset $\mathcal{D}$ is then used to update the policy, followed by further interaction with the environment using this new policy. This setting should be familiar to us, as we have seen algorithms that uses a replay buffer to update the policy. For example, DQN is an off-policy method as it updates the policy (or more precisely the Q network) by training on the replay buffer. In contrast, SARSA is an on-policy method, as it updates the Q-function using actions from the same policy. See Figure 1 for a depiction of the difference between off-policy and on-policy RL.



(a) On-policy RL

(b) Off-policy RL

Figure 1: depiction of off-policy and on-policy RL (Figure adopted from [2])

In this lecture, we will focus on a specific sub-problem of off-policy RL. Specifically, we investigate off-policy policy evaluation. Recall that evaluation of a policy π refer to estimating the expected future reward available from each state following that policy. More precisely, in the discounted finite horizon set up, the policy evaluation of π, denoted by $V^\pi(s)$, is given by

$$V^\pi(s) = \mathbb{E} \sum_{t=1}^{H} \gamma^t r_t$$

Where s is the initial state, γ is the discount factor, H is the horizon, and r_t is the reward at time t with s as the initial state. Note that the expectation is with respect to the randomness in the transition dynamics and the stochastic policy π.

The Estimation of the policy value function , $V^\pi(s)$ is a central question in RL and is a part of many policy-based algorithms, such as policy iteration. In the model-free setting, we have seen that we can estimate the policy evaluation using methods like Monte Carlo. However, that method naturally requires interacting with the environment using the policy we want to evaluate. In many situations, such interaction may not be possible, as it is costly and/or risky to try an unknown policy. For example, deciding on which medical treatment to suggest for a patient, which advertisement to show a user visiting a website, and which curriculum to recommend for a student are very risky decisions with a potentially costly consequences. Hence, we ask the question,

Can we estimate the value function of a specific target policy from data collected by another policy?

Setup. We consider the episodic MDP. Specifically, the agent interacts with its environment in a sequence of episodes, indexed by m. Each episode has a finite number of time steps $t \in \{0, \ldots, T\}$. The environment has a finite number of states and action, and the unknown state-transition probabilities are denoted by $\mathbb{P}(s' \mid s, a)$. Further, let $r(s, a)$ denotes the one-step reward at state s and action a. Under this MDP, we are interested in two (arbitrary) stationary Markov policies:

- π_b: A behavioral policy used to generate the data. (with $\pi_b(\cdot, \cdot) > 0 \forall s, a$.
- π_g: The target (goal) policy whose value function we seek to estimate.

Goal. As alluded to earlier, the goal is Using episodes generated via π_b, evaluate $V^{\pi_g}(\cdot)$, the value function of the policy π_g,

2 Importance Sampling Based Estimators

One way to view the task of off-policy learning is that it is a mismatch of distributions: we want trajectories sampled from the distribution of the target policy π_g; however, we have data drawn from the distribution of the behavior policy π_b. Importance sampling is a classical technique for handling just this kind of mismatch.

For example, consider the task of estimating the expectation of a random variable x distributed according to $\mathbb{P}_d$ through samples drawn from another distribution $\mathbb{P}_0$. See Figure 2 for an example of such a situation. It turns out, as long as $\mathbb{P}_0(x) > 0 \ \forall x$ then,

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_d}[x] &= \int_x x \mathbb{P}_d(x) dx = \int_x x \frac{\mathbb{P}_d(x)}{\mathbb{P}_0(x)} \mathbb{P}_0(x) dx \\ &= \mathbb{E}_{\mathbb{P}_0}\left[x \frac{\mathbb{P}_d(x)}{\mathbb{P}_0(x)}\right]\end{aligned}$$

And hence, the following is a consistent, unbiased estimator:

$$E[x] \approx \frac{1}{n} \sum_{i=1}^{n} x_i \frac{\mathbb{P}_d(x_i)}{\mathbb{P}_0(x_i)}$$

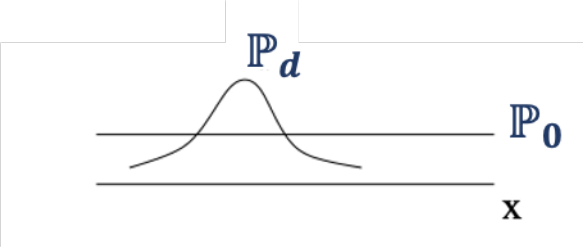


Figure 2: Can we estimate the expectation of X distributed according to $\mathbb{P}_d$ through samples drawn from another distribution $\mathbb{P}_0$?

This motivates a simple, unbiased, and consistent estimator of $V^{\pi_g}(\cdot)$, as we explain in the following procedure.

1. Using policy π_b, collect M episodes worth of data from the environment. That is, in each episode, we collect the data $\{(s_0, a_0, r_0), (s_1, a_1, r_1), \ldots, (s_{T-1}, a_{T-1}, r_{T-1}), s_T\}$.
2. We want to estimate the value function $V^{\pi_g}(s)$, for an arbitrary state s. To do so, let t_m be the first time when $s_t = s$ in the m-th episode.
3. Then we define the importance sampling estimator of $V^{\pi_g}(s)$ as:

$$V^{IS}(s) := \frac{1}{M} \sum_{m=1}^{M} R_m w_m,$$

where $\quad R_m := \sum_{t=t_m}^{T} \gamma^{t-tm} r_t \quad w_m := \prod_{t=t_m}^{T} \frac{\pi_g(s,a)}{\pi_b(s,a)}$

Note that in the on-policy case, when $\pi_b = \pi_g$, we get our Monte-Carlo estimator back.

While this is an unbiased and consistent estimator, it suffers from high variance. Specifically, if an unlikely trajectory occurs its weight w_m will be very large and cause large variation. That is, the reward of that trajectory would be given a very high weight and hence our estimate would be very sensitive to these low probability trajectories (according to π_b). Thus, one way to get around this is to use a weighted average of the samples. Specifically, the weighted importance sampling estimator is defined as

$$V^{WIS}(s) := \frac{\sum_{m=1}^{M} R_m w_m}{\sum_{m=1}^{M} w_m}.$$

While this is a consistent estimator, it is biased. However, it has a lower variance, and hence shown to be more stable in practice.

The two IS estimators defined above all consider complete trajectories, which make them unsuitable for an incremental implementation. This motivated the Per decision IS algorithm, which we describe next. Recall that,

$$R_m w_m = \sum_{t=t_m}^{T} \gamma^{t-tm} r_t \prod_{t'=t_m}^{T} \frac{\pi_g\left(s_{t'}, a_{t'}\right)}{\pi_b\left(s_{t'}, a_{t'}\right)} \tag{1}$$

$$R_m w_m = \sum_{t=t_m}^{T} \gamma^{t-tm} r_t \frac{\pi_g\left(s_{t_m}, a_{t_m}\right)}{\pi_b\left(s_{t_m}, a_{t_m}\right)} \cdots \frac{\pi_g\left(s_{t-1}, a_{t-1}\right)}{\pi_b\left(s_{t-1}, a_{t-1}\right)} \frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \cdots \frac{\pi_g\left(s_T, a_T\right)}{\pi_b\left(s_T, a_T\right)}. \tag{2}$$

Intuitively, the reward at time t should not depend on the future transitions, which motivate the estimator

$$V^{PD}(s) := \frac{1}{M} \sum_{m=1}^{M} \sum_{t=t_m}^{T} \gamma^{t-tm} r_t \frac{\pi_g\left(s_{t_m}, a_{t_m}\right)}{\pi_b\left(s_{t_m}, a_{t_m}\right)} \cdots \frac{\pi_g\left(s_{t-1}, a_{t-1}\right)}{\pi_b\left(s_{t-1}, a_{t-1}\right)}$$

This estimator, as shown in Theorem 1 in [3], The per-decision importance sampling estimator is a consistent unbiased estimator of $V^{\pi_g}(\cdot)$.

Proof. To prove this, we will show that $\mathbb{E}\left[V^{IS}\right] = \mathbb{E}\left[V^{PD}\right]$. This is sufficient as we know that V^{IS} is unbiased.

$$\mathbb{E}\left[V^{IS}(s)\right] = \mathbb{E}\left[\left(\sum_{t=t_m}^{T} \gamma^{t-t_m} r_t\right) \prod_{t'=t_m}^{T} \frac{\pi_g\left(s_{t'}, a_{t'}\right)}{\pi_b\left(s_{t'}, a_{t'}\right)} \mid s_{t_m} = s, \pi_b\right] \tag{3}$$

$$= \mathbb{E}\left[\left(\sum_{t=t_m}^{T} \gamma^{t-t_m} r_t \prod_{t'=t_m}^{T} \frac{\pi_g\left(s_{t'}, a_{t'}\right)}{\pi_b\left(s_{t'}, a_{t'}\right)}\right) \mid s_{t_m} = s, \pi_b\right] \tag{4}$$

Let's examine the expectation for one of the terms for some $t_m \leq t \leq T$:

$$\mathbb{E}\left[\gamma^{t-t_m} r_t \prod_{t=t_m}^{T} \frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \mid s_{t_m} = s, \pi_b\right] \tag{5}$$

$$= \mathbb{E}\left[\gamma^{t-t_m} r_t \frac{\pi_g\left(s_{t_m}, a_{t_m}\right)}{\pi_b\left(s_{t_m}, a_{t_m}\right)} \cdots \frac{\pi_g\left(s_{t-1}, a_{t-1}\right)}{\pi_b\left(s_{t-1}, a_{t-1}\right)} \mid s_{t_m}, a_{t_m}, \ldots, s_{t-1}, a_{t-1}\right] \cdot \mathbb{E}\left[\frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \cdots \frac{\pi_g\left(s_T, a_T\right)}{\pi_b\left(s_T, a_T\right)} \mid s_t, a_t, \pi_b\right] \tag{6}$$

Note that the second term evaluates to one as follow:

$$\mathbb{E}\left[\frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \cdots \frac{\pi_g\left(s_T, a_T\right)}{\pi_b\left(s_T, a_T\right)} \mid s_t, a_t, \pi_b\right] \tag{7}$$

$$= \sum_{s_t a_t, \ldots, s_T} \frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \cdots \frac{\pi_g\left(s_T, a_T\right)}{\pi_b\left(s_T, a_T\right)} \mathbb{P}_{\pi_b}\left[s_{t+1}, a_{t+1}, \ldots, s_T\right] \tag{8}$$

$$= \sum_{s_p a_p \ldots, s_T} \frac{\pi_g\left(s_t, a_t\right)}{\pi_b\left(s_t, a_t\right)} \cdots \frac{\pi_g\left(s_T, a_T\right)}{\pi_b\left(s_T, a_T\right)} \prod_{t'=t+1}^{T} \pi_b\left(s_t, a_t\right) \mathbb{P}\left(s_{t'} \mid s_t, a_t\right) \tag{9}$$

$$= \sum_{s_p a_r \ldots, s_T} \pi_g\left(s_{t+1}, a_{t+1}\right) \ldots \pi_g\left(s_T, a_T\right) \prod_{t'=t+1}^{T} \mathbb{P}\left(s_{t'} \mid s_t, a_t\right) \tag{10}$$

$$= \sum_{s_t a_t, \ldots, s_T} \mathbb{P}_{\pi_g}\left[s_{t+1}, a_{t+1}, \ldots, s_T\right] = 1 \tag{11}$$

Thus, by using the Markov property, we have:

$$\mathbb{E}\left[\left(\sum_{t=t_m}^{T}\gamma^{t-t_m r_t}\prod_{t'=t_m}^{T}\frac{\pi_g\left(s_{t'},a_{t'}\right)}{\pi_b\left(s_{t'},a_{t'}\right)}\right)\mid s_{t_m}=s,\pi_b\right]=\mathbb{E}\left[\left(\sum_{t=t_m}^{T}\gamma^{t-t_m r_t}\prod_{t'=t_m}^{t-1}\frac{\pi_g\left(s_{t'},a_{t'}\right)}{\pi_b\left(s_{t'},a_{t'}\right)}\right)\mid s_{t_m}=s,\pi_b\right],$$

That is, $\mathbb{E}\left[V^{IS}\right]=\mathbb{E}\left[V^{PD}\right]$.

□

Finally, similar to how the weighted IS estimator was introduced, a weighted version of the PD can be considered. Specifically,

$$V^{WPD}(s):=\frac{\sum_{m=1}^{M}\sum_{t=t_m}^{T}\gamma^{t-t_m}r_t\frac{\pi_g(s_{t_m},a_{tm})}{\pi_b(s_{tm}a_{t_m})}}{\sum_{m=1}^{M}\sum_{t=t_m}^{T}\frac{\pi_B\left(s_{t_t},a_{t_m}\right)}{\pi_b(s_{t_m},a_{t_m})}}\cdots\frac{\pi_g\left(s_{t-1},a_{t-1}\right)}{\pi_b\left(s_{t-1},a_{t-1}\right)}.$$

Similar to the weighted IS estimator, this estimator is biased but has a lower variance.

3 Doubly Robust Estimator - Guided Importance Sampling (adapted from [1])

In Section 2, we noticed that Importance Sampling methods suffer from high variance, and this variance can become especially large for longer horizon. One option to resolve this problem is to look at some model-based methods for off-policy policy evaluation. Specifically, we can try fitting a model $\widehat{M}$ to the given data and learn the transition function $\widehat{P}(s'|s,a)$ and the reward function $\widehat{R}(s,a)$, and then evaluating the policy to get $V^{\pi_g}(\cdot)$ through the estimated model. These model-based methods have very low variance and work well if the observed data is well-predictive of the state. However, they suffer from high bias if we use function approximators or if the observed data does not indicate the states very well. Worse, it is often not possible to quantify the bias in terms of the data.

To make a good use of both ideas, we can use guided importance sampling, i.e., use an approximate model to guide but not replace importance sampling estimates. This gives us the idea of Doubly Robust estimators. These are unbiased estimators and they have low variance if we use a good model to subtract baseline value estimates. We can see this first in a contextual bandit (i.e., horizon of one) setting where we estimate $V^{\pi_g}(\cdot)$ using the equation

$$V_{DR}(s)=\widehat{V}(s)+\rho(r-\widehat{R}(s,a)),$$

where $\rho=\frac{\pi_g(s,a)}{\pi_b(s,a)}$ and $\widehat{V}(s)=\sum_a\pi_g(s,a)\widehat{R}(s,a)$

We can extend this idea to the sequential setting. Firstly, we know that $V_{step-IS}^{H+1-t}=\rho_t(r_t+\gamma V_{step-IS}^{H-t})$ and $\mathbb{E}[r_t+\gamma V_{step-IS}^{H-t}]=Q(s_t,a_t)$. Using this, we get the estimation equation

$$V_{DR}^{H+1-t}=\widehat{V}(s_t)+\rho_t(r_t+\gamma V_{DR}^{H-t}-\widehat{Q}(s_t,a_t))$$

Analysis of the variance of DR estimator. Variance of DR estimator is given by

$$Var_t[V_{DR}^{H+1-t}]=Var_t[V(s_t)]+\mathbb{E}_{\approx}[Var_t[\rho_t\Delta(s_t,a_t)|s_t]]+\mathbb{E}_{\approx}[\rho_t^2Var_{t+1}[r_t]]+\mathbb{E}_{\approx}[\gamma^2\rho_t^2Var_{t+1}[V_{DR}^{H-t}]],$$

where $\Delta(s_t, a_t) = \widehat{Q}(s_t, a_t) - Q(s_t, a_t)$. This variance depends on the estimate $\widehat{Q}$ via the error term $\widehat{Q} - Q$. Note that Importance Sampling corresponds to the special case with $\widehat{Q} = 0$. Therefore, the variance of this estimator is generally lower than that of IS and performs better than IS if $\widehat{Q}$ is well chosen.

DR Cramer Rao bound. Cramer Rao Bound is a lower bound on the variance of an unbiased estimator of a parameter. It is used to find the minimum MSE or variance an unbiased estimator can possibly attain. Note that biased estimators can achieve even lower MSE or variance. The paper in [1] examines that DR achieves the lower bound at least in two cases - Discrete-tree MDP and MDP with DAG structure.

1. Discrete-tree MDP. Variance of any unbiased OPE (off-policy policy estimator) is lower bounded by
$$\sum_{t=1}^{H+1} \mathbb{E}[\rho_{1:(t-1)}^2 Var_t[V(s_t)]]$$
. We can see that DR achieves this bound if we unfold the recursion for variance of DR estimator.

2. MDP with Directed Acyclic Graph (DAG) structure. Variance of an unbiased OPE is lower bounded by
$$\sum_{t=1}^{H+1} \mathbb{E}[\frac{P_g(s_{t-1}, a_{t-1})^2}{P_b(s_{t-1}, a_{t-1})^2} Var_t[V(s_t)]]$$
where for trajectory τ, $P_b(\tau) = \mu(s_1)\pi_b(a_1|s_1)P(s_2|s_1, a_1)\dots P(s_{H+1}|s_H, a_H)$ and P_b is its marginal probability. $P_g(\cdot)$ is similarly defined for π_g. Again, DR achieves this bound.

DR Results. DR achieves lower MSE than IS and WIS for OPE in MountainCar Environment (see figure 3).

DR Safe Policy Improvement. Given the application of OPE in safe policy improvement, it makes sense to check whether OPE methods like DR perform well in this domain. The idea is to evaluate a target policy and get a reasonable value estimate before we deploy that policy.

The problem is formulated as follows. Dataset D is generated through a uniformly random policy π_b and is split into D_{train} and D_{test}. D_{train} is used to estimate the model and identify the optimal policy π_{train} given the estimated model. π_b and π_{train} are mixed in various amounts to get various candidate policies π_g, which are then evaluated in D_{test} using IS, DR and other methods.

The results are shown in figure 4. When the problem is formulated to select good policies incrementally (policy with highest Lower Confidence Bound $V - C\sigma$), DR performs much better than IS methods. And when the problem is formulated to select bad policies incrementally (policy with lowest LCB), DR performs at least as safe as IS methods. Therefore, DR can be used a replacement for IS in safe policy improvement domain.

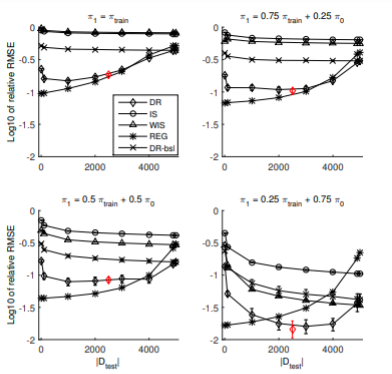


Figure 1. Comparison of the methods as point estimators on Mountain Car. 5000 trajectories are generated for off-policy evaluation, and all the results are from over 4000 runs. The subgraphs correspond to the target policies produced by mixing π_{train} and π_0 with different portions. X-axes show the size of D_{test}, the part of the data used for IS/WIS/DR/DR-bsl. The remaining data are used by the regression estimator (REG; DR uses it as $\widehat{Q}$). Y-axes show the RMSE of the estimates divided by the true value in logarithmic scale. We also show the error of 2-fold DR as an isolated point ($\diamond$).

Figure 3: DR comparison with Importance Sampling methods in Mountain Car environment

3.1 Weighted Doubly Robust Estimator (adapted from [4])

Moving further, we can try WDR (Weighted Doubly Robust) estimators. The key idea here is that we need not have 0 bias. It is sufficient to have low MSE for off-policy policy evaluation so that we can use evaluated policies for purposes like policy iteration. MSE of an estimator $\widehat{\theta}$ is given by $MSE(\widehat{\theta}, \theta) = \mathbb{E}[(\theta - \widehat{\theta})^2] = Var(\widehat{\theta}) + Bias(\widehat{\theta})^2$. To have low MSE, we need not have 0 bias but we need to balance this bias-variance tradeoff. This is done by WDR estimators. Similar to WIS, the weights are defined as $w_t^i = \frac{\rho_t^i}{\sum_{j=1}^n \rho_t^j}$. We can see the results of DR, WDR and other estimators in a few different environments in figure 5.

4 Hybrid Estimators: MAGIC Estimator (adapted from [4])

We can check from some experiments that WDR sometimes outperforms model-based methods and sometimes gives worse results than model-based methods. One key rationale is to identify what gives it the error. Essentially, if reward and state transitions are stochastic and importance weights have high variance, WDR can have high mean squared error, thereby making it perform worse than model-based method if model can be perfectly learnt.

After identifying this error, we can ask whether there is a way we can get an estimator which switches between WDR and model-based or combines them in such a way that it performs almost as good as WDR for the first kind of scenarios and almost as good as model-based for the second kind of scenarios. Here, we come up with the idea of partially importance sampling based estimators. These estimators use IS methods for first j steps of the horizon and model-based methods for the remaining steps of the horizon.

$$g^{(j)}(D) = IS^{[0:j]}(D) + AM^{[j+1:\infty]}(D)$$

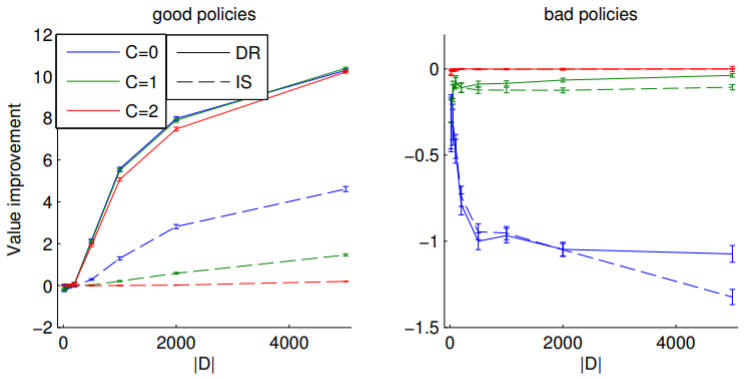
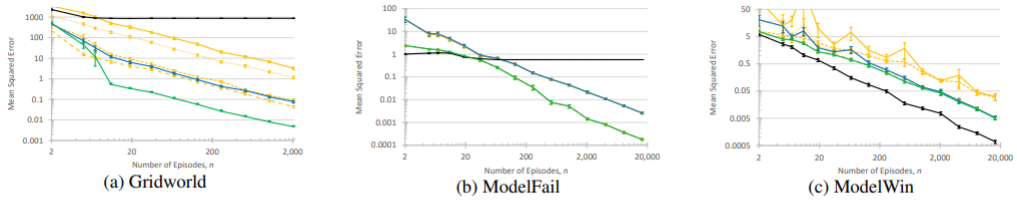


Figure 4. Safe policy improvement in Mountain Car. X-axis shows the size of data and y-axis shows the true value of the recommended policy subtracted by the value of the behavior policy.

Figure 4: DR Safe Policy Improvement Results



(a) Gridworld (b) ModelFail (c) ModelWin

Figure 1: Empirical results for three different experimental setups. All plots in this paper have the same format: they show the mean squared error of different estimators as n, the number of episodes in D, increases. Both axes always use a logarithmic scale and standard error bars are included from 128 trials. All plots use the following legend:

IS PDIS WIS CWPDIS DR AM WDR

Figure 5: DR, WDR and other IS-based methods compared

$$g^{(\infty)}(D) = \lim_{j \to \infty} g^{(j)}(D)$$

Note that $g^{(-1)}(D)$ corresponds to a purely model-based estimator and $g^{(\infty)}(D)$ corresponds to an importance sampling estimator. We can take a linear combination of this big class of estimators to get the best of both worlds. This is called the MAGIC estimator (model and guided importance sampling combined), which is essentially $\sum_j x_j g^{(j)}(D)$ where weights x_j are to be determined.

Here is a more elaborate explanation of how we create the MAGIC estimator. First, we define a vector $g(D) = (g^{(-1)}(D), g^{(0)}D, g^{(1)}(D), \dots)^T$ of all partially importance sampling based estimators. The new estimator is going to be $x^T g(D)$ where x is chosen such that $MSE(x^T g(D), v(\pi_g))$ is minimized. To make computation easier, we choose $g_J(D) \in R^{|J|}$ where J is a finite subset of $\{-1, 0, 1, \dots\} \cup \{+\infty\}$. This reduces the problem to finding x such that $x^T[\Omega_n + b_n b_n^T]x$ is minimized, where matrix Ω_n and column vector b_n are defined as follows.

$$\Omega_n(i,j) = Cov(g^{(J_i)}(D), g^{(J_j)}(D))$$

$$b_n(j) = \mathbb{E}[g^{(J_j)}(D)] - v(\pi_g)$$

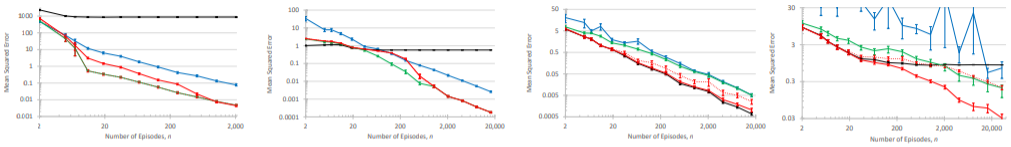
Now, Ω_n can be estimated using $\widehat{\Omega}_n(j,k) = \frac{n}{n-1}\sum_{i=1}^{n}(g_i^{(J_j)}(D) - \bar{g}_i^{(J_j)}(D)) \times (g_i^{(J_k)}(D) - \bar{g}_i^{(J_k)}(D))$. Estimating b_n is more tricky because using Model-based gives high bias and using IS gives high variance. So, it is done using distance from confidence intervals. See figure 6 for a description of the algorithm.

Algorithm 1 MAGIC(D)

1: **Input:** Historical data, $\mathcal{D}$, evaluation policy, π_e, an approximate model, and a set of return-lengths, $\mathcal{J}$.
2: Compute $|\mathcal{J}| \times |\mathcal{J}|$ matrix $\widehat{\Omega}_n$ according to (5).
3: Compute a 90% confidence interval, $[l, u]$, on WDR(D) using the percentile bootstrap method.
4: Compute $|\mathcal{J}| \times 1$ vector $\widehat{\mathbf{b}}_n$, where $\widehat{\mathbf{b}}_n(j) = \text{dist}(g^{(\mathcal{J}_j)}(D), [l, u])$.
5: $\mathbf{x} \leftarrow \arg\min_{\mathbf{x} \in \Delta^{|\mathcal{J}|}} \mathbf{x}^\intercal[\widehat{\Omega}_n + \widehat{\mathbf{b}}_n\widehat{\mathbf{b}}_n^\intercal]\mathbf{x}$
6: **return** $\mathbf{x}^\intercal \mathbf{g}_{\mathcal{J}}(D)$

Figure 6: MAGIC Algorithm description

If we look at the results (figure 7) in the same environments tried earlier, we can see that this estimator performs comparable to the best estimators for the given situations and outperforms both WDR and model-based when the first few states are partially observable whereas later states are fully observable.



(a) Gridworld (b) ModelFail (c) ModelWin (d) Hybrid

Figure 2: Empirical comparison of MAGIC to other estimators using the legend from Figure 1. All plots use the following legend (although only Figure 2d includes MAGIC-B):

DR AM WDR MAGIC MAGIC-B

Figure 7: MAGIC Estimator compared with model-based and IS-based methods

5 Contributions

Abdullah Alomar prepared Sections 1 and 2, while Satvat Jagwani provided the scribe for Sections 3 and 4. TA Tiancheng reviewed the draft.

References

[1] N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2016.

[2] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

[4] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.