Empirical Analysis of Off-Policy Policy Evaluation for Reinforcement Learning

Cameron Voloshin [1] **Hoang M. Le** [1] **Yisong Yue** [1]

Abstract

Off-policy policy evaluation (OPE) is the task of predicting the online performance of a policy using only pre-collected historical data (collected from an existing deployed policy or set of policies). For many real-world applications, accurate OPE is crucial since deploying bad policies can be prohibitively costly or dangerous. With the increasing interest in deploying learning-based methods for safety-critical applications, the study of OPE has also become correspondingly more important. In this paper, we present the first comprehensive empirical analysis of most of the recently proposed OPE methods. Based on thousands of experiments and detailed empirical analyses, we offer a summarized set of guidelines for effectively using OPE in practice, as well as suggest directions for future research to address current limitations.

1. Introduction

Off-policy policy evaluation (OPE) aims to estimate the value of a new policy from a pre-collected dataset (Dann et al., 2014). In many real-world applications of reinforcment learning (RL), deploying a new policy to estimate its value can be prohibitively expensive (Jiang & Li, 2016). Testing a new policy, without analyzing how it will perform, in systems such as robotics, autonomous vehicles, trading, advertising, drug trials, traffic control put people, capital and equiptment at risk (Li et al., 2011; Wiering, 2000; Bottou et al., 2013; Bang & Robins, 2005). It is critically important to generate accurate counter-factual predictions of how well a new policy performs without running the policy.

Contemporary methods can be partitioned into three classes of algorithms (Farajtabar et al., 2018). The first class is the direct methods (DM). DM aim to fit the value of a policy directly through model-free function approximation or by fitting a model of the environment and averaging over rollouts within such a model. Alternatively, the second class is the inverse propensity score (IPS) methods, also known as importance sampling (IS) methods. IPS methods estimate the value of a policy by rescaling rewards by importance sample weights, a measure of how likely a reward is under a new policy compared to the data-generating policy. The third class is the Doubly-Robust methods (DRM), which blends between the IPS methods and an estimate of the action-value function, typically supplied by a DM.

Many of these algorithms were initially designed for OPE in the bandit scenario (Precup et al., 2000; Dudík et al., 2011; Swaminathan et al., 2017; Wang et al., 2017). Later these methods were generalized and new methods were created to tackle the RL problems (Precup et al., 2000; Mahmood et al., 2014; Jiang & Li, 2016; Thomas et al., 2015; Harutyunyan et al., 2016; Thomas & Brunskill, 2016; Farajtabar et al., 2018; Liu et al., 2018; Le et al., 2019). Yet, existing benchmarking of the algorithms is lacking, not comprehensive, and fails to provide insight into areas where methods particularly shine and where current research should focus. We aim to present a benchmark which highlights challenges one expects to face when tackling the RL setting. We implement the majority of contemporary OPE methods and report consice findings based on a systematic evaluation.

2. Notation

We will use a notational standard that is common across modern OPE literature (Thomas & Brunskill, 2016). We define a markov decision process by $\langle X, A, P, R, \gamma \rangle$, where X is the state space, A is the action space, $P : X \times A \to X$ is the transition function with $P(x'|x, a)$ giving the probability entering state x' by taking action a in state x, $R : X \times A \to \mathbb{R}$ is the reward function with $R(x, a)$ giving the reward of taking action a in state x, and $\gamma \in (0, 1]$ is the discount factor. Let d_0 be the initial state distribution. Let a policy $\pi : X \times A \to [0, 1]$ be a distribution of actions for each state where an action $a \in A$ in state $x \in X$ has probability $\pi(a|x)$.

We assume we are given a historical dataset, D, of the form $D = \{\tau^i = (x_0^i, a_0^i, r_0^i, s_1^i, \ldots)\}_{i=1}^{|D|}$ where i denotes the ith trajectory in D. Each trajectory i is generated by a, possibly unique, policy π_i. In the event that there is only one data-generating policy, we call this policy π_b, base

[1]California Institute of Technology, Pasadena, CA. Correspondence to: Cameron Voloshin <cvoloshi@caltech.edu>.

Real-world Sequential Decision Making workshop at ICML 2019.

policy. Hence $D = \{(\tau^i, \pi_i)\}_{i=1}^{|D|}$, where $\tau^i \sim \pi_i$. Let $\rho^i_{j:j'} = \rho_{j:j'}(\tau^i, \pi_e, \pi_i) = \prod_{t=j}^{\min(j', |\tau^i|-1)} \frac{\pi_e(a^i_t|x^i_t)}{\pi_i(a^i_t|x^i_t)}$ be the importance weight between an evaluation policy, π_e, and a base policy π_i (Thomas & Brunskill, 2016). Let $w_{j:j'} = \frac{1}{|D|}\sum_{i=1}^{|D|} \rho^i_{j:j'}$ be a normalization factor of the importance weights. For convenience, when $t' < t$, let $\rho^i_{t:t'} = 1$ for any trajectory i.

Finally, given a desired evaluation policy π_e, we define the value of π_e as

$$V(\pi_e) = \mathbb{E}_{x\sim d_0}\left[\sum_{t=0}^{|\tau|-1} \gamma^t r_t | x_0 = x\right],$$

with $a_t \sim \pi_e(\cdot|x_t)$, $x_{t+1} \sim P(\cdot|x_t, a_t)$, $r_t \sim R(x_t, a_t)$, $\tau = (x_0, a_0, r_0, \ldots)$. Unless otherwise specified, we omit dependence on π_e and abbreviate the value as V.

3. Methods

3.1. Inverse Propensity Scoring (IPS) Methods

There are several variations of the IPS method. Let $H_i = |\tau_i| - 1$,

Table 1. IPS methods. (Dudík et al., 2011; Jiang & Li, 2016)

POLICY	STANDARD	STEP-WISE
IS	$\sum_{i=1}^{\lvert D\rvert} \frac{\rho^i_{0:H_i}}{\lvert D\rvert}(\sum_{t=0}^{H_i}\gamma^t r_t)$	$\sum_{i=1}^{\lvert D\rvert}\sum_{t=0}^{H_i}\gamma^t \frac{\rho^i_{0:t}}{\lvert D\rvert} r_t$
WIS	$\sum_{i=1}^{\lvert D\rvert} \frac{\rho^i_{0:H_i}}{w_{0:H_i}}(\sum_{t=0}^{H_i}\gamma^t r_t)$	$\sum_{i=1}^{\lvert D\rvert}\sum_{t=0}^{H_i}\gamma^t \frac{\rho^i_{0:t}}{w_{0:t}} r_t$

Table 1 shows the calculation for the four traditional IPS estimators: $V_{IS}, V_{Step-IS}, V_{WIS}, V_{Step-WIS}$. In addition, we include the following method as well since it is a Rao-Blackwellization (Liu et al., 2018) of the IPS estimators:

State Density Ratio Estimation (IH): (Liu et al., 2018)

$$V_{IH} = \sum_{i=1}^{|D|}\sum_{t=0}^{H_i} \frac{\gamma^t \omega(s^i_t)\rho_{t:t} r^i_t}{\sum_{i'=0}^{|D|}\sum_{t'=1}^{H_i}\gamma^{t'}\omega(s^{i'}_{t'})\rho_{t':t'}}$$

$$\omega(s^i_t) = \lim_{t\to\infty} \frac{\sum_{t=0}^{T}\gamma^t d_{\pi_e}(s^i_t)}{\sum_{t=0}^{T}\gamma^t d_{\pi_b}(s^i_t)}$$

where π_b is assumed to be a fixed data-generating policy, and d_π is the distribution of states when executing π from $s_0 \sim d_0$. The details for how to find ω can be found in Algorithm 1 and 2 of (Liu et al., 2018).

3.2. Doubly-Robust Methods (DRM)

DRM rely on being supplied an action-value function $\widehat{Q}$, an estimate of Q, from which one can also yield $\widehat{V}(x) = \sum_{a\in A}\pi(a|x)\widehat{Q}(x,a)$. Let $H_i = |\tau_i| - 1$.

Sequential Doubly-Robust (SDR): (Jiang & Li, 2016)

$$V_{SDR} = \frac{1}{|D|}\sum_{i=1}^{|D|} V^{H_i}_{Seq-DR}, \quad V^0_{Seq-DR} \equiv 0$$

$$V^{H_i+2-t}_{SDR} = \widehat{V}(x_t) + \rho_{0:t}(r_t + \gamma V^{H_i+1-t}_{Seq-DR} - \widehat{Q}(s_t, a_t))$$

Doubly-Robust (DR): (Thomas & Brunskill, 2016)

$$V_{DR} = \frac{1}{|D|}\sum_{i=1}^{|D|}\widehat{V}(x^i_0) + \frac{1}{|D|}\sum_{i=1}^{|D|}\sum_{t=0}^{\infty}\gamma^t\rho^i_{0:t}[r^i_t - \widehat{Q}(x^i_t, a^i_t) + \gamma\widehat{V}(x^i_{t+1})]$$

Weighted Doubly-Robust (WDR): (Thomas & Brunskill, 2016)

$$V_{WDR} = \frac{1}{|D|}\sum_{i=1}^{|D|}\widehat{V}(x^i_0) + \sum_{i=1}^{|D|}\sum_{t=0}^{\infty}\gamma^t\frac{\rho^i_{0:t}}{w_{0:t}}[r^i_t - \widehat{Q}(x^i_t, a^i_t) + \gamma\widehat{V}(x^i_{t+1})]$$

MAGIC: (Thomas & Brunskill, 2016) Given $g_J = \{g^i | i \in J \subseteq \mathbb{N}\cup\{-1\}\}$ where

$$g^j(D) = \sum_{i=1}^{|D|}\sum_{t=0}^{j}\gamma^t\frac{\rho^i_{0:t}}{w_{0:t}}r^i_t + \sum_{i=1}^{|D|}\gamma^{j+1}\frac{\rho^i_{0:t}}{w_{0:t}}\widehat{V}(x^i_{j+1}) - \sum_{i=1}^{|D|}\sum_{t=0}^{j}\gamma^t(\frac{\rho^i_{0:t}}{w_{0:t}}\widehat{Q}(x^i_t, a^i_t) - \frac{\rho^i_{0:t-1}}{w_{0:t-1}}\widehat{V}(x^i_t)),$$

then define $dist(y, Z) = \min_{z\in Z}|y - z|$ and

$$\widehat{b}_n(j) = dist(g^J_j(D), CI(g^\infty(D), 0.5))$$

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

then, for a $|J|-$simplex $\Delta^{|J|}$ we can calculate

$$\widehat{x}^* \in \arg\min_{x\in\Delta^{|J|}} x^T[\widehat{\Omega}_n + \widehat{b}\widehat{b}^T]x$$

which, finally, yields

$$V_{MAGIC} = (\widehat{x}^*)^T g_J.$$

MAGIC can be thought of as a weighted average of different blends of the DM and DRM. In particular, for some $i \in J$, g^i represents estimating the first i steps of $V(\pi_e)$ according to DR (or WDR) and then estimating the remaining steps via $\widehat{Q}$. Hence, V_{MAGIC} finds the most appropriate set of weights which trades off between using a direct method and a DRM.

3.3. Direct Methods (DM)

3.3.1. MODEL-BASED

Fitting the MDP (MDP): (Jiang & Li, 2016) An approach to model-based value estimation is to directly fit the transition

dynamics $P(x_{t+1}|x_t, a_t)$, reward $R(x_t, a_t)$, and terminal condition $P(x_{t+1} \in X_{terminal}|x_t, a_t)$ of the MDP using some for of maximum likelihood or function approximation. This yields a simulation environment from which one can extract the value of a policy using an average over rollouts. Thus, $V(\pi) = \mathbb{E}[\sum_{t=1}^{T} \gamma^t r(x_t, a_t)|x_0 = x, a_0 = \pi(x_0)]$ where the expectation is over initial conditions $x \sim d_0$ and the transition dynamics of the simulator.

3.3.2. Model-Free

Every estimator in this section will approximate Q with $\widehat{Q}(\cdot;\theta)$, parametrized by some θ. From $\widehat{Q}$ the OPE estimate we seek is

$$V = \frac{1}{|D|} \sum_{i=1}^{|D|} \sum_{a \in A} \pi_e(a|s) \widehat{Q}(s_0^i, a; \theta)$$

Let $H_i = |\tau_i| - 1$, the length of trajectory i. Note that $\mathbb{E}_{\pi_e} Q(x_{t+1}, \cdot) = \sum_{a \in A} \pi_e(a|x_{t+1}) Q(x_{t+1}, a)$.

Direct Model Regression (Reg): (Farajtabar et al., 2018)

$$\widehat{Q}(\cdot, \theta) = \min_{\theta} \frac{1}{|D|} \sum_{i=1}^{|D|} \sum_{t=0}^{H_i} \gamma^t \rho_{0:t}^i \left(R_{t:H_i}^i - \widehat{Q}(x_t^i, a_t^i; \theta) \right)^2$$

$$R_{t:H_i}^i = \sum_{t'=t}^{H_i} \gamma^{t'-t} \rho_{(t+1):t'}^i r_{t'}^i$$

Fitted Q Evaluation (FQE): (Le et al., 2019) $\widehat{Q}(\cdot, \theta) = \lim_{k \to \infty} \widehat{Q}_k$ where

$$\widehat{Q}_k = \min_{\theta} \frac{1}{|D|} \sum_{i=1}^{|D|} \sum_{t=0}^{H_i} (\widehat{Q}_{k-1}(x_t^i, a_t^i; \theta) - y_t^i)^2$$

$$y_t^i \equiv r_t^i + \gamma \mathbb{E}_{\pi_e} \widehat{Q}_{k-1}(x_{t+1}^i, \cdot; \theta)$$

Retrace(λ) (R(λ)), Tree-Backup (Tree), $Q^\pi(\lambda)$: (Munos et al., 2016; Precup et al., 2000; Harutyunyan et al., 2016) $\widehat{Q}(\cdot, \theta) = \lim_{k \to \infty} \widehat{Q}_k$ where

$$\widehat{Q}_k(x, a; \theta) = \widehat{Q}_{k-1}(x, a; \theta) + \mathbb{E}_{\pi_b}[\sum_{t \geq 0} \gamma^t \prod_{s=1}^{t} c_s y_t | x_0 = x, a_0 = a]$$

and

$$y_t = r^t + \gamma \mathbb{E}_{\pi_e} \widehat{Q}_{k-1}(x_{t+1}, \cdot; \theta) - \widehat{Q}_{k-1}(x_t, a_t; \theta)$$

$$c_s = \begin{cases} \lambda \min(1, \frac{\pi_e(a_s|x_s)}{\pi_b(a_s|x_s)}) & R(\lambda) \\ \lambda \pi_e(a_s|x_s) & Tree \\ \lambda & Q^\pi(\lambda) \end{cases}$$

More Robust Doubly-Robust (MRDR): (Farajtabar et al., 2018) Given

$$\Omega_{\pi_b}(x) = diag[1/\pi_b(a|x)]_{a \in A} - ee^T$$

$$e = [1, \ldots, 1]^T$$

$$R_{t:H_i}^i = \sum_{j=t}^{H_i} \gamma^{j-t} \rho_{(t+1):j}^i r(x_j^i, a_j^i)$$

and

$$q_\theta(x, a, r) = diag[\pi_e(a'|x)]_{a' \in A} [\widehat{Q}(x, a'; \theta)]_{a' \in A} - r[\mathbf{1}\{a' = a\}]_{a' \in A}$$

where $\mathbf{1}$ is the indicator function, then

$$\widehat{Q}(\cdot, \theta) = \min_{\theta} \frac{1}{|D|} \sum_{i=1}^{|D|} \sum_{t=0}^{H_i} \gamma^{2t} (\rho_{0:t-1}^i)^2 \times \rho_t^i q_\theta(x_t^i, a_t^i, R_{t:H_i}^i)^T \Omega_{\pi_b}(x_t^i) q_\theta(x_t^i, a_t^i, R_{t:H_i}^i)$$

4. Environments

For every environment, we initialize the environment with a fixed horizon length T. If the agent reaches a goal before T or if the episode is not over by step T, it will transition to an environment-dependent absorbing state where it will stay until time T.

4.1. Toy-Graph

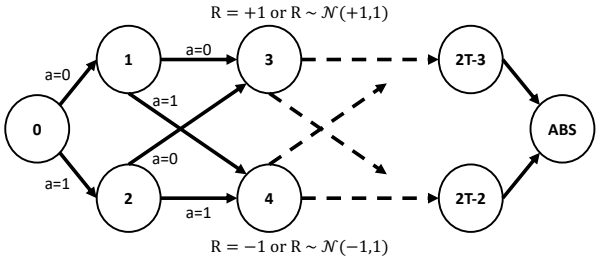

Figure 1. Toy-Graph Environment

Figure 1 shows a visualization of the Toy-Graph environment. The graph is initialized with horizon T and with absorbing state $x_{abs} = x_0 = 0$. In each episode, the agent starts at a single starting state $x_0 = 0$ and has two actions, $a = 0$ and $a = 1$. At each time step $t < T$, the agent can enter state $x_{t+1} = 2t + 1$ by taking action $a = 0$, or $x_{t+1} = 2t + 2$ by taking action $a = 1$. If the environment is stochastic, we simulate noisy transitions by allowing the agent to slip into $x_{t+1} = 2t+2$ instead of $x_{t+1} = 2t+1$ and vice-versa with probability .25. At the final time $t = T$, the agent always enters the terminal state $x_{abs} = 0$. The reward is $+1$ if the agent transitions to an odd state, otherwise is -1. If the environment provides sparse rewards, then $r = +1$ if x_{T-1} is odd, $r = -1$ if x_{T-1} is even, otherwise $r = 0$. Similarly to deterministic rewards, if the environment's rewards are stochastic, then the reward is $r \sim N(1, 1)$ if the

agent transitions to an odd state, otherwise $r \sim N(-1, 1)$. If the rewards are sparse and stochastic then $r \sim N(1, 1)$ if x_{T-1} is odd, otherwise $r \sim N(-1, 1)$ and $r = 0$ otherwise.

Table 2. Toy-Graph Policies

POLICY	$\pi(a=0)$	$\pi(a=1)$
$\pi_{TG,0}$	.2	.8
$\pi_{TG,1}$	.6	.4
$\pi_{TG,2}$	.8	.2

4.2. Toy Mountain Car (Toy-MC)

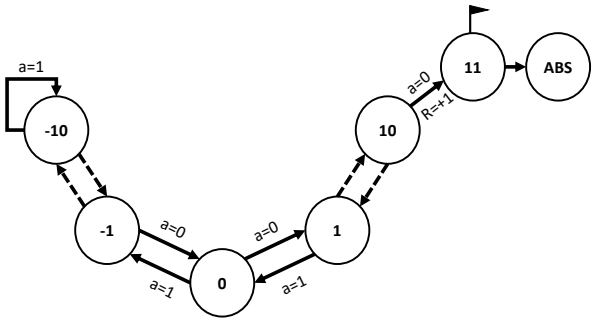

Figure 2. Toy-MC Environment

Figure 2 shows a visualization of the Toy-MC environment. This environment is a 1-D graph-based simplification of Mountain Car (see following description for the standard version). The agent starts at $x_0 = 0$, the center of the valley and can go left or right. There are 21 total states, 10 to the left of the starting position and 11 to the right of the starting position, and a terminal absorbing state $x_{abs} = x_0 = 0$. The agent recieves a reward of $r = 0$ at every timestep unless reaching $x = +11$, where the agent transitions to x_{abs}, the episode terminates, and the agent receives a reward of $+1$. If the agent reaches $x = -10$ and continues left then the agent remains in $x = -10$. If the agent does not reach state $x = +11$ by step T then the episode terminates and the agent transitions to the absorbing state.

Table 3. Toy-MC Policies

POLICY	$\pi(a=0)$	$\pi(a=1)$
$\pi_{TMC,0}$	.55	.45
$\pi_{TMC,1}$	.4	.6

4.3. Mountain Car

We use the OpenAI version of Mountain Car with a few simplifying modifications (Brockman et al., 2016; Sutton & Barto, 2018). The car starts in a valley and has to go back and forth to gain enough momentum to scale the mountain and reach the end goal. The state space is given by the position and velocity of the car. At each time step, the car has the following options: accelerate backwards, forwards or do nothing. The reward is $r = -1$ for every time step until the car reaches the goal. While the original trajectory length is capped at 200, we decrease the effective length by applying every action a_t five times before observing x_{t+1}. Furthermore, we modify the random initial position from being uniformly between $[-.6, -.4]$ to being one of $\{-.6, -.5, -.4\}$, with no velocity. The environment is initialized with a horizon T and absorbing state $x_{abs} = [.5, 0]$, position at .5 and no velocity.

Table 4. MC Policies. These policies are epsilon greedy. With probability $1 - \epsilon$ select $\max_a Q(x, a)$ otherwise select randomly. Q is trained with DQN (Mnih et al., 2013; Van Hasselt et al., 2016)

POLICY	ϵ	DESCRIPTION
$\pi_{MC,0}$	1.	RANDOM POLICY
$\pi_{MC,1}$	0.	ALWAYS FOLLOW Q

5. Experimental Analysis

5.1. Preliminary Details

An experiment generally consists of selecting an environment, the data-collecting policy (π_b), the evaluation policy (π_e), and the number of trajectories to collect ($|D|$). We rollout π_b a total of $|D|$ times to generate a dataset onto which we apply different methods to calculate $\widehat{V}(\pi_e)$, an estimate of the value. We rollout π_e a total of $|D|$ times from which we calculate (a monte-carlo estimate of) the true on-policy value $V(\pi_e)$. Each experiment is repeated 10 times with different random seeds. For each method we compute the mean squared error (MSE) over the seeds, MSE$= \frac{1}{10}\sum_{i=1}^{10}(\widehat{V}(\pi_e)_i - V(\pi_e)_i)^2$, a standard metric for OPE (Farajtabar et al., 2018).

We have run thousands of experiments and have highlighted the details in tables. Methods that are within $2\times$ the lowest MSE among the set of experiments are highlighted in bold.

5.2. Horizon and Environment

What is the effect of horizon on OPE methods? What effect do sparse/dense rewards and determinstic/stochastic transitions or rewards have on how well methods can be expected to perform?

Table 5 and Table 6 illustrate the effects on the methods from increasing the horizon on the Toy-Graph environment from $T = 4$ to $T = 16$ under complete determinism and dense rewards. While every method experiences

an increase in error due to increased horizon but fixed number of trajectories, methods that rely on terms with ρ (REG,MRDR,$R(\lambda)$,$Q^\pi(\lambda)$, both DM and DRM counterparts, and IPS) suffer substantially more. The repeated multiplication of importance weights causes the variance of the ρ term to blows up exponentially which can cripple IPS and DRM-based performance (Liu et al., 2018).

Secondly, for finite state and actions spaces, the inherent bellman error is zero:

$$\sup_{g \in F} \inf_{f \in F} \|f - \mathbb{T}^{\pi_e} g\|_{\pi_e} = 0$$

for F, an $|S| \times |A|$-dimensional table, and $\|\cdot\|_{\pi_e}$, the ℓ_2 norm weighted by the π_e-induced state-action distribution, and $\mathbb{T}^\pi Q = r + \gamma \mathbb{E}_{x' \sim X}[Q(x', \pi(x')]$. Direct methods such as FQE and $Q^\pi(\lambda)$, without ρ terms, are all capable of finding an unbiased Q^{π_e} which is the fixed point to $T^{\pi_e} Q^{\pi_e} = Q^{\pi_e}$ when using F in a finite state and action space environment. Thus, these methods have very small error regardless of horizon. We examine the extent of the bias when the inherent bellman error is not zero in the later sections.

Table 5. MSE Toy-Graph. $T = 4$. $|D| = 1024$. $\pi_b = \pi_{TG,0}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Deterministic transitions and rewards. Reward is received at every timestep.

	DM	DRM: Combining $\widehat{Q}$ with			
$\widehat{Q}$ (Tab.)	Direct	SDR	DR	WDR	MAGIC
MDP	3E-03	4E-02	4E-02	3E-02	3E-02
Reg	3E-01	5E-03	5E-03	4E-03	2E-02
MRDR	1E-01	3E-02	3E-02	2E-01	2E-01
FQE	**6E-04**	**6E-04**	**6E-04**	**6E-04**	**6E-04**
R(λ)	**6E-04**	**6E-04**	**6E-04**	**6E-04**	**6E-04**
Q$^\pi(\lambda)$	**6E-04**	**6E-04**	**6E-04**	**6E-04**	**6E-04**
Tree	**6E-04**	**6E-04**	**6E-04**	**6E-04**	**6E-04**

IPS		
Method	Standard	Step-wise
IH	8E-03	
IS	2E0	2E-01
WIS	3E-01	6E-02
NAIVE	2E1	

Under stochastic transitions as seen in Table 7 and Table 8, the spread in the performance of the methods becomes tighter: the best and worst case performances are within 2 orders of magnitude rather than 4 orders of magnitude (Table 5 and Table 6). The environment is homogenizing the policies and making them closer together (we will describe this in further detail in the Policy Mismatch section).

Unlike stochasticity in the environment which makes DRM help most methods (other than FQE and $Q^\pi(\lambda)$), stochasticity of rewards has the opposite effect as seen in Table 9.

Table 6. MSE Toy-Graph. $T = 16$. $|D| = 1024$. $\pi_b = \pi_{TG,0}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Deterministic transitions and rewards. Reward is received at every timestep.

	DM	DRM: Combining $\widehat{Q}$ with			
$\widehat{Q}$ (Tab.)	Direct	SDR	DR	WDR	MAGIC
MDP	6E-03	8E0	8E0	4E0	2E-02
Reg	2E1	1E2	1E2	2E1	2E1
MRDR	3E1	4E1	4E1	3E2	3E2
FQE	**1E-03**	**1E-03**	**1E-03**	**1E-03**	**1E-03**
R(λ)	6E0	5E0	5E0	4E0	6E0
Q$^\pi(\lambda)$	**1E-03**	**1E-03**	**1E-03**	**1E-03**	**1E-03**
Tree	1E1	8E0	8E0	5E0	1E1

IPS		
Method	Standard	Step-wise
IH	6E-02	
IS	6E1	2E1
WIS	5E1	1E1
NAIVE	3E2	

Table 7. MSE Toy-Graph. $T = 4$. $|D| = 1024$. $\pi_b = \pi_{TG,0}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Stochastic transitions and deterministic rewards. Reward is received at every timestep.

	DM	DRM: Combining $\widehat{Q}$ with			
$\widehat{Q}$ (Tab.)	Direct	SDR	DR	WDR	MAGIC
MDP	**2E-02**	1E-01	1E-01	1E-01	3E-02
Reg	9E-02	1E-01	1E-01	9E-02	5E-02
MRDR	8E-02	1E-01	1E-01	1E-01	1E-01
FQE	**1E-02**	9E-02	9E-02	9E-02	**2E-02**
R(λ)	3E-02	9E-02	9E-02	9E-02	**2E-02**
Q$^\pi(\lambda)$	5E-02	1E-01	1E-01	9E-02	3E-02
Tree	3E-02	9E-02	9E-02	9E-02	**2E-02**

IPS		
Method	Standard	Step-wise
IH	3E-02	
IS	4E-01	1E-01
WIS	3E-01	6E-02
NAIVE	5E0	

Direct methods relying on regressing on rewards, such as MRDR and Reg, will perform worse and require more data. We found sparsity of reward to have no notable effect on how the OPE methods perform.

Most notably the top performer is direct FQE. MDP and IH, particularly when T is large, are close competitors. Among the DRM, only MAGIC remains somewhat competitive. Since MAGIC is a blend between a direct method and WDR of that direct method, the choice of the set J (see description of the method) is critical to its performance. The authors offer some heuristics for selecting J but a principled approach remains an interesting open problem.

Table 8. MSE Toy-Graph. $T = 16$. $|D| = 1024$. $\pi_b = \pi_{TG,0}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Stochastic transitions and deterministic rewards. Reward is received at every timestep.

	DM	DRM: COMBINING $\widehat{Q}$ WITH			
$\widehat{Q}$ (TAB.)	DIRECT	SDR	DR	WDR	MAGIC
MDP	**3E-02**	2E1	2E1	1E1	6E-02
REG	7E0	3E0	3E0	7E0	9E0
MRDR	7E0	8E0	8E0	3E1	3E1
FQE	**4E-02**	3E0	3E0	1E0	4E-01
R(λ)	3E0	3E0	3E0	8E-01	3E0
Q$^\pi$(λ)	2E-01	3E0	3E0	1E0	3E-01
TREE	4E0	4E0	4E0	8E-01	4E0

	IPS	
METHOD	STANDARD	STEP-WISE
IH	**3E-02**	
IS	2E1	7E0
WIS	1E1	9E-01
NAIVE	7E1	

Table 9. MSE Toy-Graph. $T = 16$. $|D| = 1024$. $\pi_b = \pi_{TG,0}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Deterministic transitions and stochastic rewards. Reward is received at every timestep.

	DM	DRM: COMBINING $\widehat{Q}$ WITH			
$\widehat{Q}$ (TAB.)	DIRECT	SDR	DR	WDR	MAGIC
MDP	**9E-02**	1E3	1E3	1E1	2E0
REG	1E2	7E2	7E2	1E3	1E2
MRDR	7E1	3E3	3E3	2E3	5E2
FQE	**6E-02**	2E2	2E2	2E0	**7E-02**
R(λ)	7E0	4E1	4E1	6E0	6E0
Q$^\pi$(λ)	5E-01	2E2	2E2	1E0	5E-01
TREE	1E1	1E2	1E2	8E0	1E1

	IPS	
METHOD	STANDARD	STEP-WISE
IH	**1E-01**	
IS	1E4	1E2
WIS	5E1	1E1
NAIVE	3E2	

5.3. Policy Mismatch

To what extent does the difference between the evaluation and base policy influence OPE methods?

When the base policy is equivalent to the evaluation policy, $\rho_{t:t'} \equiv 1$ for any t, t'. Importance weight-based methods reduce to on-policy value evaluation and the estimators take a sample average of the rewards in the dataset. However, as the policies diverge, the variance of $\rho = \prod \frac{\pi_e(x,a)}{\pi_b(x,a)}$ grows exponentially. While we have illustrated in the last section that horizon and stochasticity in the environment can lead to trouble for methods relying on importance weights ρ, we have found that it does not always. We quantify the amount of policy mismatch by calculating the conservative estimate

$$M = \left(\sup_{(x,a) \in X \times A} \frac{\pi_e(x,a)}{\pi_b(x,a)} \right)^T. \quad (1)$$

Clearly $M > \rho_{0:T}$ for any T.

Table 10. MSE Toy-Graph. $T = 4$. $|D| = 1024$. $\pi_b = \pi_{TG,1}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Stochastic transitions and deterministic rewards. Reward is recieved at every timestep.

	DM	DRM: COMBINING $\widehat{Q}$ WITH			
$\widehat{Q}$ (TAB.)	DIRECT	SDR	DR	WDR	MAGIC
MDP	7E-03	8E-03	8E-03	8E-03	**4E-03**
REG	5E-03	**3E-03**	**3E-03**	**3E-03**	5E-03
MRDR	3E-02	**3E-03**	**3E-03**	**2E-03**	1E-02
FQE	**2E-03**	**3E-03**	**3E-03**	**3E-03**	**2E-03**
R(λ)	**3E-03**	**3E-03**	**3E-03**	**3E-03**	**3E-03**
Q$^\pi$(λ)	**4E-03**	**3E-03**	**3E-03**	**3E-03**	**4E-03**
TREE	**3E-03**	**3E-03**	**3E-03**	**3E-03**	**3E-03**

	IPS	
METHOD	STANDARD	STEP-WISE
IH	**3E-03**	
IS	8E-03	5E-03
WIS	6E-03	**4E-03**
NAIVE	6E-01	

Table 11. MSE Toy-Graph. $T = 16$. $|D| = 1024$. $\pi_b = \pi_{TG,1}$. $\pi_e = \pi_{TG,2}$. Tabular (Tab) function class. Stochastic transitions and deterministic rewards. Reward is recieved at every timestep.

	DM	DRM: COMBINING $\widehat{Q}$ WITH			
$\widehat{Q}$ (TAB.)	DIRECT	SDR	DR	WDR	MAGIC
MDP	**2E-02**	1E-01	1E-01	2E-01	4E-02
REG	7E-02	3E-02	3E-02	3E-02	3E-02
MRDR	5E-02	3E-02	3E-02	3E-02	5E-02
FQE	**1E-02**	3E-02	3E-02	3E-02	**1E-02**
R(λ)	3E-02	3E-02	3E-02	3E-02	3E-02
Q$^\pi$(λ)	**1E-02**	3E-02	3E-02	3E-02	**2E-02**
TREE	5E-02	3E-02	3E-02	4E-02	**2E-02**

	IPS	
METHOD	STANDARD	STEP-WISE
IH	**2E-02**	
IS	3E-01	8E-02
WIS	9E-02	3E-02
NAIVE	7E0	

Table 7 and Table 10 illustrate the difference between $M = (\frac{.8}{.2})^4 = 256$ and $M = (\frac{.8}{.6})^4 \approx 3.16$, respectively. Since M is a measure of policy similarity, then we can expect the error to be smaller for smaller M across the board. In these experiments the error is lower by about an order of magnitude.

Similarly, 8 and Table 11 illustrate the difference between $M = (\frac{.8}{.2})^{16} \approx 4.29 \times 10^9$ and $M = (\frac{.8}{.6})^{16} \approx 100$. The crucial distinction between M small and M large is the reaction of the DRM estimators to importance weights. When M is small, DRM help some DM (like Reg and MRDR) and only mildly hurt other direct methods' performance, while when M is large they generally hurt DM across the board. We find it difficult to predict when DRM will help with this experiment, but have found that there is normally a DM which performs better than any DRM.

5.4. State Distribution Mismatch

Under what conditions do IPS methods perform well but IH does not?

IH (Liu et al., 2018) greatly circumvents the issue of policy mismatch and horizon by learning a ratio of state visitation frequency. We have already seen an instance where learning this state ratio distribution is challenging. In particular notice that in Table 5 and Table 6, IH struggles slightly and we wish to understand why.

Table 12. MSE Toy-MC. $T = 250$. $|D| = 1024$. $\pi_b = \pi_{TMC,0}$. $\pi_e = \pi_{TMC,1}$. Tabular (Tab) function class.

	DM	DRM: Combining $\widehat{Q}$ with			
$\widehat{Q}$ (Tab.)	Direct	SDR	DR	WDR	MAGIC
MDP	8e-02	1e0	2e-02	2e-02	2e-02
Reg	7e-03	2e3	4e1	4e1	3e1
MRDR	8e-03	6e0	2e-01	2e-01	2e-01
FQE	**5e-05**	2e3	2e1	2e1	2e1
R(λ)	2e-01	4e0	2e-04	2e-04	7e-04
Q$^\pi(\lambda)$	8e-04	4e1	5e-01	6e-01	5e-01
Tree	4e-01	2e1	4e-03	9e-04	2e-03

	IPS	
Method	Standard	Step-wise
IH	1e-01	
IS	4e-03	4e-03
WIS	3e-04	9e-04
NAIVE	3e-01	

For the Toy-MC environment, even though T is very large, because the environment terminates when the agent reaches the goal, the horizon over which ρ compounds is significantly shorter. Hence IPS (and DRM) actually should stand a chance here. Table 12 is generated with a policy that goes right with probability .55 while the evaluation goes right with probability .4. In this environment, a base policy which preferentially goes right causes the environment to terminate (since the goal is reached) and would take a lot of data to explore all the states to the left. IH has a large error compared to many DM, DR methods, and even standard IS. On the other hand, when reversing the probabilities with which the agent goes right , the IH estimator struggles less (achieving 3e-4 error, several orders of magnitude less than in Table 12) since the evaluation policy induces a set of states that is well within the base policy's support. This issue illustrates IH's sample inefficiency and serves as an example of how standard IS can beat IH when the the policies induce a large enough support mismatch.

5.5. Function Class

How does the choice of function class and/or representation influence affect performance of OPE methods?

Since direct methods rely on reductions to supervised regression in order to get an OPE estimate, their performance could potentially be influenced by their ability to generalize to unseen states. Once the state space becomes continuous then we cannot be sure that the inherent bellman error is zero for our choice of function class. Measuring the amount of inherent bellman error is intractable (Sutton & Barto, 2018).

Table 13. MSE MC. Linear Function class. $T = 250$. $|D| = 128$. $\pi_b = \pi_{MC,0}$. $\pi_e = \pi_{MC,1}$.

	DM	DRM: Combining $\widehat{Q}$ with			
$\widehat{Q}$ (Lin.)	Direct	SDR	DR	WDR	MAGIC
MDP	5e1	3e2	3e2	2e3	6e2
Reg	6e2	4e2	4e2	8e3	2e3
MRDR	7e2	4e2	4e2	5e2	4e2
FQE	2e2	2e2	2e2	2e4	1e3
R(λ)	3e1	**1e1**	**1e1**	7e2	7e2
Q$^\pi(\lambda)$	4e1	2e2	2e2	1e3	1e3
Tree	3e1	**1e1**	**1e1**	7e2	7e2

	IPS	
Method	Standard	Step-wise
IH	2e3	
IS	1e3	5e2
WIS	**2e1**	2e2
NAIVE	3e3	

Table 13 and Table 14 illustrate the results of using a linear vs neural network function class on the MC environment. Increasing the complexity of the function class does not necessarily lead to an improvement in performance across the board. In fact, among direct methods, only FQE saw improvement. Reg and MRDR saw little change. R(λ), $Q^\pi(\lambda)$, Tree, and MDP saw dramatic declines in performance.

5.6. Technical Challenges

How long does it take to run these methods? What are the challenges of each method?

For this qualitative analysis, we consider a single training

Table 14. MSE MC. Neural Network function class. $T = 250$. $|D| = 128$. $\pi_b = \pi_{MC,0}$. $\pi_e = \pi_{MC,1}$.

	DM	DRM: COMBINING $\widehat{Q}$ WITH			
$\widehat{Q}$ (NN.)	DIRECT	SDR	DR	WDR	MAGIC
MDP	8E2	4E2	4E2	2E3	1E3
REG	7E2	4E2	4E2	5E2	2E2
MRDR	9E2	5E2	5E2	2E2	4E2
FQE	**5E1**	2E2	2E2	1E4	3E3
R(λ)	9E2	1E3	1E3	2E3	2E3
Q$^\pi$(λ)	3E3	4E3	4E3	5E4	8E3
TREE	1E3	1E3	1E3	3E3	2E3

	IPS	
METHOD	STANDARD	STEP-WISE
IH	1E3	
IS	1E3	5E2
WIS	**8E1**	2E2
NAIVE	3E3	

batch $B \subset \{(x_t^i, a_t^i, r_t^i, x_{t+1}^i)\} = D$ and the neural network function class.

Closed form methods such as the IPS methods (other than IH) will run the fastest, since there is no reliance on training or inference. IH runs fairly quickly as well up to the choice of batch size $|B|$. In fact, part of the algorithm for IH requires creating a kernel matrix, a metric of state similarity, of size $|B| \times |B|$. As $|B|$ grows, the algorithm will slow. Yet, the larger the kernel matrix, the better the algorithm performs because it is able to properly assign similar visitation frequency weight to neighboring states. If $|B|$ is too small, all the states seem independent. For IH to function properly, $|B|$ has to be chosen to be as large as possible while weighing the computational cost.

To calculate the direct methods, model free methods require $\mathcal{O}(|D|)$ calls to the neural network. Model based methods (such as MDP) require $\mathcal{O}(T|D|)$ calls. Doubly-Robust methods demand an estimate $\widehat{Q}$ for every trajectory i and time t, which requires $\mathcal{O}(T|D|)$ and $\mathcal{O}(T^2|D|)$ calls for model free and model based, respectively. In general inference is fast since it can be done in batch. However, because MDP requires rollouts (depending on calls to π_e) it must be done serially and this has a huge impact on the run-time of DR-based MDP.

We found R(λ), $Q^\pi(\lambda)$, Tree can be quite expensive to run in practice, especially when using neural networks. For every item $(x_t^i, a_t^i) \in B \subset D$, they all require calculating $\widehat{Q}(x_{t'}^i, a_{t'}^i)$ for every $t' > t$. On average, this requires $\mathcal{O}(T|B|)$ calls to the neural network to do a single step of gradient descent. On the other hand, FQE is generally cheap, only requiring $|B|$ calls per gradient step.

FQE, R(λ), $Q^\pi(\lambda)$, Tree suffer from similar issues that have plagued standard DQN (Mnih et al., 2013). In particular, because they are iterative approaches, they rely on moving targets and the quality of their performance relies on how well the loss stays under control as the errors compound between fitting Q_k on Q_{k-1}. In addition, these methods tend to oscillate and their convergence is finnicky. Tricks and tips that make DQN more stable can be developed to help these methods fit.

While MRDR was designed so that DR MRDR beats DR Reg, we find MRDR to be ill conditioned in discrete settings and the choice of regularization impacts its performance.

6. Discussion

We have found that direct FQE tends to perform consistently well when the model class is chosen appropriately (tabular in discrete settings and neural network in continuous settings). This would be our choice, in practice. DR methods rarely outperform the best direct method, but do help struggling direct methods when the effective horizon is short enough. Yet we have not found a consistent way to predict when a DR approach would help a method. DR methods are particularly prone to error when there is modest to long horizons and sufficient policy mismatch. IH, designed to fix the pitfalls of DR methods, can be robust to lengthy horizon and policy mismatch. However it is sensitive to hyperparameter tuning and base vs evaluation policy support mismatch in the absense of large quantities of data. IPS methods generally only perform well when the effective horizon is short and there is sufficient policy match. As expected, DR MRDR performs better than DR Reg in general, but neither of these methods is consistently among the best.

A few avenues for research that we have found include

- Creating a blend between DM and IH (like DR creates a blend between DM and IPS).
- Embedding actions into IH to learn state-action density, which may make it robust to its current issues
- Finding a principled way of choosing the set J in MAGIC and the batch size in IH
- Finding ways to stabilize iterative methods like FQE

We leave the study of how high dimensional state spaces, such as pixel-based Atari, and POMPDs affect the methods to future work.

References

Bang, H. and Robins, J. M. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005. doi: 10.1111/j.1541-0420.2005.00377.x.

Bottou, L., Peters, J., nonero Candela, J. Q., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260, 2013.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Dann, C., Neumann, G., and Peters, J. Policy evaluation with temporal differences: A survey and comparison. *The Journal of Machine Learning Research*, 15(1):809–883, 2014.

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

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. In *ICML*, 2018.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. Q(lambda) with off-policy corrections. In *ALT*, 2016.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661, 2016.

Le, H. M., Voloshin, C., and Yue, Y. Batch policy learning under constraints. *arXiv preprint arXiv:1903.08738*, 2019.

Li, L., Chu, W., Langford, J., and Wang, X. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. pp. 297–306, 01 2011. doi: 10.1145/1935826.1935878.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5361–5371, 2018.

Mahmood, A. R., van Hasselt, H. P., and Sutton, R. S. Weighted importance sampling for off-policy learning with linear function approximation. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3014–3022. Curran Associates, Inc., 2014.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and efficient off-policy reinforcement learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1054–1062. Curran Associates, Inc., 2016.

Precup, D., Sutton, R. S., and Singh, S. P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 759–766, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3632–3642. Curran Associates, Inc., 2017.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Thomas, P., Theocharous, G., and Ghavamzadeh, M. High-confidence off-policy evaluation, 2015.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, pp. 5. Phoenix, AZ, 2016.

Wang, Y.-X., Agarwal, A., and Dudík, M. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pp. 3589–3597, 2017.

Wiering, M. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158, 2000.