Advanced Topics in Reinforcement Learning

Lecture 7: Off-Policy Monte Carlo Methods

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework released. Due: October 21 at 9:30AM (minute class starts)
- Read chapter 6 for next week.
- Project proposals due: Thursday, October 2nd.

Learning Outcomes

After this week, you will be able to:

1. Differentiate between value function computation and learning.

2. Describe and implement approaches to estimating value functions from sampled experience in an MDP.

3. Learn optimal policies from sampled experience.

4. Differentiate between on- and off-policy learning.

Single-State First-Visit Monte Carlo

- Only estimate $q_\pi(s_0, a_0)$ for a fixed state, s_0, a_0.
- Assume we always start in state s_0 and all episodes eventually terminate.
- To evaluate policy π, set $\texttt{total} \leftarrow 0$, and repeat n times:
  - Start at s_0, take action a_0.
  - Until termination: $S_t, R_t \sim p(S', R \mid S_{t-1}, A_{t-1}), A_t \sim \pi(A \mid S_t)$.
  - $\texttt{total} \leftarrow \texttt{total} + \sum_{t=0} \gamma^t R_{t+1}$.
- Return $Q_n(s_0, a_0) \leftarrow \texttt{total}/n$
- As $n \to \infty$, $Q_n(s_0, a_0) \to q_\pi(s_0, a_0)$.

First-Visit Monte Carlo

- To evaluate policy π, set $\texttt{returns}(s, a) \leftarrow \{\}$, and repeat n times:
  - Sample $S_0, A_0 \sim d_{\texttt{explore}}$
  - Until termination: $S_t, R_t \sim p(S', R \mid S_{t-1}, A_{t-1}), A_t \sim \pi(A \mid S_t)$.
  - For first visit to (s, a) in episode (at timestep i):
    - $\texttt{returns}(s, a) \leftarrow \texttt{returns}(s, a) \cup \{ \sum_{t=i}^{T} \gamma^t R_{t+1} \}$.
- Return $Q_n(s_0, a_0) \leftarrow \texttt{mean}(\texttt{returns})$
- As $n \to \infty$, $Q_n(s_0, a_0) \to q_\pi(s_0, a_0)$.

Monte Carlo Policy Iteration

- To find $\pi^\star$, start with arbitrary π_0, and alternate:
  - Run Monte Carlo policy evaluation of π_k for n episodes.
  - Make π_{k+1} the greedy policy w.r.t. q_k.
- How large must n be?
- Exploring starts ensures convergence only if all returns averaged come from same policy.
  - Conjectured that there is no need to discard returns as policy changes but no formal proof.

Brian’s Presentation

High-Dimensional Continuous Control Using Generalized Advantage Estimation (GAE)

Schulman, Mortiz, Levine, Jordan, and Abbeel

Slides

Ensuring Exploration

- Exploring starts are restrictive. What else to do?
  - ϵ-greedy policies: select $a^\star = \arg\max_a q(s, a)$ with probability $1 - \epsilon$; else random action.
  - Hard policy $\equiv$ Deterministic policy, Soft policy $\equiv$ All actions have some probability.
- Do ϵ-greedy methods converge? If so, to what?
- Can we still reach $\pi^\star$?
  - What if we decay epsilon?

Off-Policy Motivation

- What is the difference between off-policy and on-policy learning?
  - Trajectories generated by behavior policy, used to evaluate target policy.
  - If behavior = target ($\forall s, a, \pi(a \mid s) = b(a \mid s)$), then on-policy. Otherwise, off-policy.
- Why do we need off-policy learning?
  - Behavior policy explores, target policy exploits.
  - Learn for many reward functions at the same time.
  - Behavior policy is a known and safe policy.
- What is the main challenge in off-policy learning?
  - Distribution shift! Behavior policy and target policy induce different trajectory distributions. Thus, $q_b(s, a) \neq q_\pi(s, a)$ in general.

Importance Sampling Methods

- Given distribution $d(X)$ and real-valued function $f(X)$, estimate:

$$\mathbf{E}_d[f(X)] = \sum_x d(x)f(x)$$

- The distribution d is unknown but we can sample $X \sim b$.

- Monte Carlo approximation:

If we set $b \leftarrow d$ then reduces to standard Monte Carlo.

$$\sum_x d(x)f(x) = \sum_x b(x)\frac{d(x)}{b(x)}f(x) \approx \frac{1}{n}\sum_{i=1}^{n}\frac{d(X_i)}{b(X_i)}f(X_i) \qquad X_i \sim b$$

- Law of large numbers tells us that as $n \to \infty$ that error in the approximation goes to zero.

- Error is order $1/\sqrt{n}$ (assuming $\frac{d(x)}{b(x)}$ is bounded).

Off-Policy Monte Carlo in RL

- Key idea: correct return distribution with importance sampling.
- Trajectory distribution that is induced from MDP and behavior policy.

$$\Pr(s_0, a_0, r_1, s_1, \ldots r_T, s_T) = \prod_{t=0}^{T-1} b(a_t \mid s_t) p(s_{t+1}, r_{t+1} \mid s_t, a_t)$$

- Desired trajectory distribution induced from MDP and target policy.

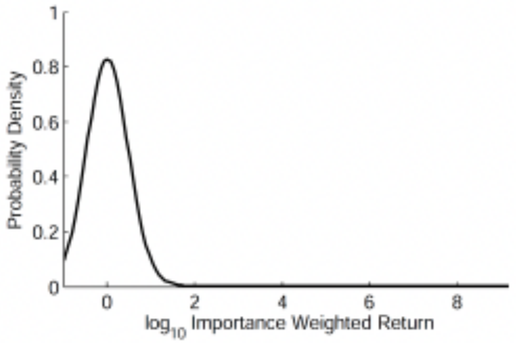
$$\Pr(s_0, a_0, r_1, s_1, \ldots r_T, s_T) = \prod_{t=0}^{T-1} \pi(a_t \mid s_t) p(s_{t+1}, r_{t+1} \mid s_t, a_t)$$

- Importance weighted returns:

$$\rho_{t:T} := \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)}{b(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)} = \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i)}{b(a_i \mid s_i)} \qquad v_\pi(s_t) \approx \rho_{t:T} G_t$$

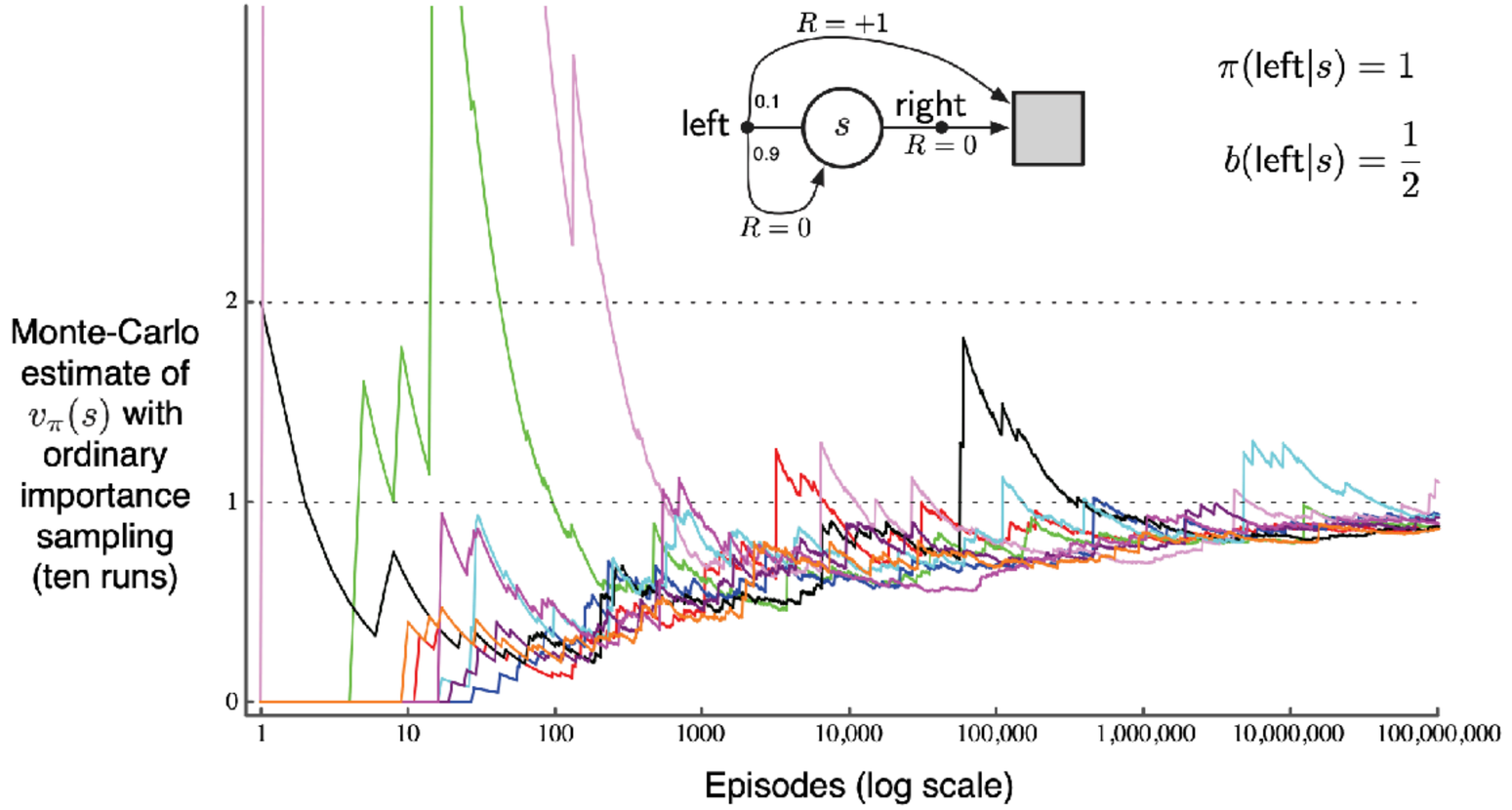
Importance Sampling Variance

- Importance sampling provides unbiased estimates of $v_\pi(s)$ using returns sampled by running the behavior policy.
  - Assuming that, if $\pi(a \mid s) > 0$, then $b(a \mid s) > 0$.
- In practice:
  - Can have infinite variance.
  - Most of the time, importance sampling severely under-estimates and then rarely, massively over-estimates.
  - Can return implausible estimates.
  - Ex: Suppose you *know* G_t is bounded and hence $v_\pi(s)$ is bounded. Importance sampling may estimate a value much greater than the bound.

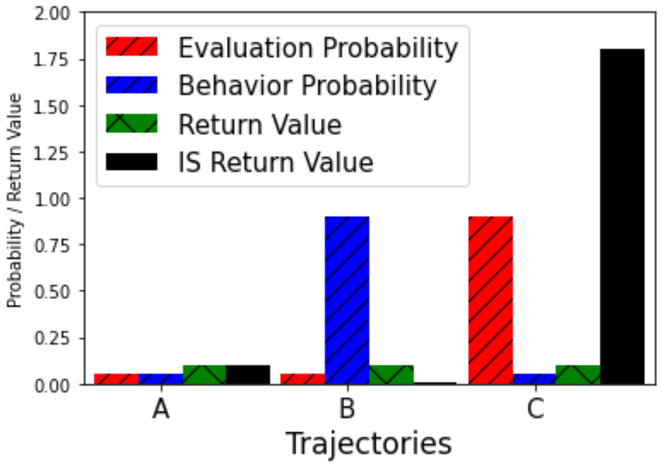


Thomas et al. 2015

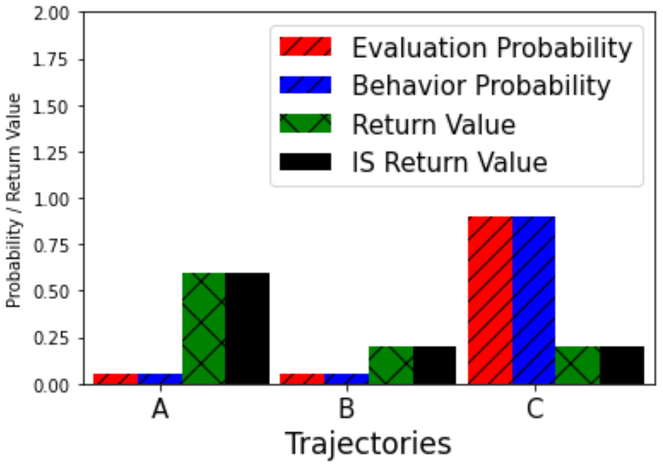
Importance Sampling Variance



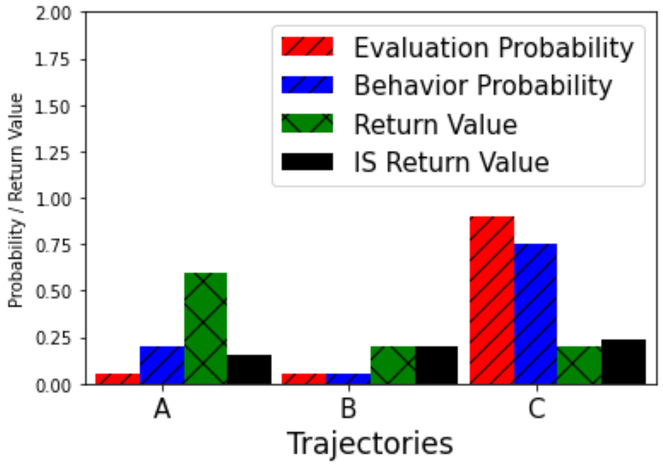
Variance of Importance Sampling



Extreme High Variance Off-Policy



On-Policy



Low Variance Off-Policy

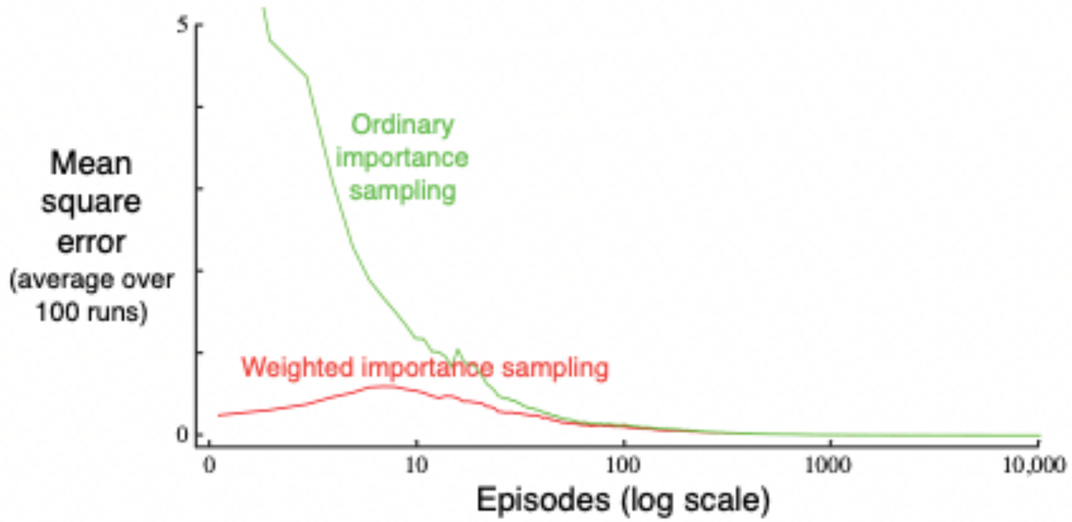
Data-efficient policy evaluation through behavior policy search: https://jmlr.org/papers/v25/21-0346.html

Weighted Importance Sampling

- Estimation error = Variance + Bias^2. Often a trade-off: can reduce variance by introducing bias.

- Weighted Importance Sampling introduces bias but can drastically lower variance.

$$V(s) := \frac{\sum_{t \in T(s)} \rho_{t:T} G_t}{\sum_{t \in T(s)} \rho_{t:T}}$$



Per-Decision Importance Sampling

- Ordinary importance sampling re-weights all rewards the same:

$$\rho_{t:T} G_t = \rho_{t:T}(R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-t-1} R_T)$$

- Actions that follow a reward do not affect the likelihood of that reward.

$$\rho_{t:T} \gamma^k R_{t+k+1} = \rho_{1:k} \cdot \rho_{k+1:T} \gamma^k R_{t+k+1}$$

- Per-decision importance sampling takes advantage of this by dropping factors in the importance ratios.

$$\mathbf{E}_b[\rho_{t:T} \gamma^k R_{t+k+1}] = \mathbf{E}_b[\rho_{1:k} \gamma^k R_{t+k+1}]$$

Off-Policy Control

- With off-policy prediction, we can run a *soft* behavior policy to provide exploration while improving the target policy greedily.
  - Behavior policy must ensure state-action coverage.
  - Ex: Behavior policy is ϵ-greedy and target policy is greedy.
- Still follow general policy iteration scheme:
  - Evaluate target policy (i.e., estimate q_π) with off-policy Monte Carlo.
  - Make target policy greedy w.r.t. q_π.
  - Converges to $\pi^\star$.
- Is this efficient?

$$\rho_{t:T} := \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)}{b(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)} = \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i)}{b(a_i \mid s_i)}$$

Off-Policy First-Visit Monte Carlo

- To evaluate policy π, set $\texttt{returns}(s, a) \leftarrow \{\}$, and repeat n times:
  - Sample $S_0 \sim d_{\texttt{initial}}$
  - Until termination: $S_t, R_t \sim p(S', R \mid S_{t-1}, A_{t-1})$ $A_t \sim b(A \mid S_t)$.
  - For first visit to (s, a) in episode (at timestep i):
    - $\texttt{returns}(s, a) \leftarrow \texttt{returns}(s, a) \cup \{\rho_{i:T} \sum_{t=i}^{T} \gamma^t R_{t+1}\}$.
- Return $Q_n(s_0, a_0) \leftarrow \texttt{mean}(\texttt{returns})$
- As $n \to \infty$, $Q_n(s_0, a_0) \to q_\pi(s_0, a_0)$.

$$\rho_{t:T} := \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)}{b(a_i \mid s_i) p(s_{i+1}, r_{i+1} \mid s_i, a_i)} = \prod_{i=t}^{T-1} \frac{\pi(a_i \mid s_i)}{b(a_i \mid s_i)}$$

How to use IS in practice

- Clip or bound weights, i.e., $\rho \leftarrow \min(\frac{\pi(a \mid s)}{b(a \mid s)}, 1)$.
- Restrict policy difference.
- Baselines and doubly robust estimators.
- Bootstrap (next week) — truncate the return after k steps and use $\gamma^{t+k-1} v_\pi(S_{t+k})$ in place of the sum of the remaining rewards.

Discounting Aware Importance Sampling

- Discounted return: $G_t := R_{t+1} + \gamma R^2_{t+2} + \ldots \gamma^{T-1} R_T$
- Alternatively, the discount represents the probability of not terminating. Episodes terminate with probability $1 - \gamma$.
- What is the expected *undiscounted* return under this formalism:
- $$(1-\gamma)R_{t+1} + (1-\gamma)\gamma(R_{t+1} + R_{t+2}) + \ldots + (1-\gamma)\gamma^{T-t-2}\sum_{k=1}^{T-1} R_{t+k} + \gamma^{T-t-1}\sum_{k=1}^{T} R_{t+k} = G_t$$
- Now a very similar idea to per-decision IS; no need to importance sample actions after all rewards in a partial return have been received.

Summary

- Off-Policy Monte Carlo policy evaluation methods enable learning q_π while taking actions according to a behavior policy b.
- Importance sampling re-weights returns so that — in expectation — they are equal to q_π.
- Off-Policy Monte Carlo policy iteration uses a behavior policy for exploration while learning an optimal target policy.

Action Items

- Start on homework
- Start reading chapter 6 for next week.
- Be thinking about final project — proposal due next week.
  - The more concrete your proposal is, the better guidance you will receive!