Online and Reinforcement Learning (2025)
Home Assignment 2

Davide Marchi 777881

Contents

1 Short Questions

Determine whether each statement below is True or False and provide a very brief justification.

1. **Statement:** *"In a finite discounted MDP, every possible policy induces a Markov Reward Process."*

   Answer: False. This statement assumes that the policy depends only on the current state. If we allow policies to depend on the *entire* past history (*history-dependent* policies), then the resulting transitions in the state space may no longer satisfy the Markov property, since the chosen action at each step might be a function of all previous states and actions. Hence not *every* (fully history-dependent) policy necessarily induces a Markov Reward Process in the *original* state space.

2. **Statement:** *"Consider a finite discounted MDP, and assume that π is an optimal policy. Then, the action(s) output by π does not depend on history other than the current state (i.e., π is necessarily stationary)."*

   Answer: False. While it is true that there *exists* an optimal policy which is stationary deterministic, it does not follow that *all* optimal policies must be so. In fact, multiple distinct policies (some stationary, others possibly history-dependent or randomized) can achieve exactly the same optimal value. Hence it is incorrect to say that *any* optimal policy π must be purely state-dependent (stationary).

3. **Statement:** *"In a finite discounted MDP, a greedy policy with respect to optimal action-value function, Q^*, corresponds to an optimal policy."*

   Answer: True. From the Bellman optimality equations for Q^*, a policy that selects

   $$\arg\max_a \; Q^*(s,a)$$

   at each state s is indeed an optimal policy. This policy attains the same value as Q^* itself, thus achieving the optimal value.

4. **Statement:** *"Under the coverage assumption, the Weighted Importance Sampling Estimator $\widehat{V}_{\mathrm{wIS}}$ converges to V^π with probability 1."*

   Answer: True. The coverage assumption ensures that the target policy's state-action probabilities are absolutely continuous w.r.t. the behavior policy. Under this assumption, Weighted Importance Sampling (though slightly biased) is a *consistent* estimator of V^π, meaning it converges to V^π as the sample size grows unbounded.

2 MDPs with Similar Parameters Have Similar Values

Setup: We have two discounted MDPs

$$M_1 \;=\; (S, A, P_1, R_1, \gamma) \quad \text{and} \quad M_2 \;=\; (S, A, P_2, R_2, \gamma),$$

sharing the same discount factor $\gamma \in (0,1)$, the same finite state–action space, and rewards bounded in $[0, R_{\max}]$. For all state–action pairs (s, a):

$$\big|\, R_1(s,a) \;-\; R_2(s,a) \big| \;\le\; \alpha, \quad \big\|\, P_1(\cdot \mid s,a) \;-\; P_2(\cdot \mid s,a) \big\|_1 \;\le\; \beta.$$

Consider a fixed stationary policy π, and let V_1^π and V_2^π be its value functions in M_1 and M_2, respectively. The goal is to show that

$$\big|\, V_1^\pi(s) \;-\; V_2^\pi(s) \big| \;\le\; \frac{\alpha \;+\; \gamma\, R_{\max}\, \beta}{(1-\gamma)^2} \quad \text{for every state } s \,\in\, S.$$

Step 1: Write down the Bellman equations for each MDP.
By definition of π, the Bellman fixed-point form is:

$$V_1^\pi \;=\; r_1^\pi + \gamma\, P_1^\pi\, V_1^\pi, \quad V_2^\pi \;=\; r_2^\pi + \gamma\, P_2^\pi\, V_2^\pi,$$

where

$$r_m^\pi(s) \;=\; R_m\big(s, \pi(s)\big), \quad (P_m^\pi f)(s) \;=\; \sum_{s'} P_m\big(s' \mid s, \pi(s)\big)\, f(s'), \quad m = 1, 2.$$

Define $\delta \;=\; V_1^\pi - V_2^\pi$. Then

$$\delta \;=\; \big(r_1^\pi - r_2^\pi\big) \;+\; \gamma\, \big(P_1^\pi V_1^\pi - P_2^\pi V_2^\pi\big).$$

To facilitate the separation of terms, we introduce and subtract $\gamma\, P_1^\pi V_2^\pi$, which allows us to rewrite the second term as:

$$P_1^\pi V_1^\pi - P_2^\pi V_2^\pi \;=\; (P_1^\pi V_1^\pi - P_1^\pi V_2^\pi) \;+\; (P_1^\pi V_2^\pi - P_2^\pi V_2^\pi).$$

Substituting this back, we obtain:

$$\delta \;=\; \big(r_1^\pi - r_2^\pi\big) \;+\; \gamma\, P_1^\pi \big(V_1^\pi - V_2^\pi\big) \;+\; \gamma\, \big(P_1^\pi - P_2^\pi\big)\, V_2^\pi.$$

Step 3: Take norms and use triangle/inequality bounds.
Taking the supremum norm ($\|\cdot\|_\infty$) on both sides we obtain

$$\|\delta\|_\infty = \big\|(r_1^\pi - r_2^\pi) + \gamma P_1^\pi \delta + \gamma (P_1^\pi - P_2^\pi) V_2^\pi \big\|_\infty.$$

By the *triangle inequality*, the norm of a sum is at most the sum of the norms, so we can split the right-hand side as:

$$\|\delta\|_\infty \;\le\; \|r_1^\pi - r_2^\pi\|_\infty + \gamma \|P_1^\pi \delta\|_\infty + \gamma \|(P_1^\pi - P_2^\pi) V_2^\pi\|_\infty.$$

Now we can proceed with:

- *Reward difference:*
  Since $\big|R_1(s,a) - R_2(s,a)\big| \le \alpha$, it follows that $\|r_1^\pi - r_2^\pi\|_\infty \le \alpha$.

- *Term with $P_1^\pi\,\delta$:*
  We have
  $$\big\|\, P_1^\pi\,\delta \,\big\|_\infty \;\le\; \|\delta\|_\infty,$$
  since P_1^π is a probability kernel and thus a contraction in sup norm.

- *Term with $(P_1^\pi - P_2^\pi)\,V_2^\pi$:*
  For each s,
  $$\big|\,(P_1^\pi - P_2^\pi)\,V_2^\pi(s)\big| \;\le\; \sum_{s'}\big|\,P_1(s' \mid s,\pi(s)) - P_2(s' \mid s,\pi(s))\big|\,\big|\,V_2^\pi(s')\big|.$$
  By assumption, $\|P_1(\cdot \mid s,a) - P_2(\cdot \mid s,a)\|_1 \le \beta$, and $\|V_2^\pi\|_\infty \le \frac{R_{\max}}{1-\gamma}$. Hence,
  $$\big\|\,(P_1^\pi - P_2^\pi)\,V_2^\pi\big\|_\infty \;\le\; \beta\,\frac{R_{\max}}{1-\gamma}.$$

Putting these bounds together,

$$\|\delta\|_\infty \;\le\; \alpha \;+\; \gamma\,\|\delta\|_\infty \;+\; \gamma\,\beta\,\frac{R_{\max}}{1-\gamma}.$$

Step 4: Solve for $\|\delta\|_\infty$.
We isolate $\|\delta\|_\infty$ on one side:

$$(1-\gamma)\,\|\delta\|_\infty \;\le\; \alpha \;+\; \gamma\,\beta\,\frac{R_{\max}}{1-\gamma}.$$

Thus

$$\|\delta\|_\infty \;\le\; \frac{\alpha}{1-\gamma} \;+\; \frac{\gamma\,\beta\,R_{\max}}{(1-\gamma)^2}.$$

Since $\alpha/(1-\gamma) \;\le\; \alpha/(1-\gamma)^2$ whenever $0 < \gamma < 1$, we can write

$$\|\delta\|_\infty \;\le\; \frac{\alpha \;+\; \gamma\,\beta\,R_{\max}}{(1-\gamma)^2}.$$

Hence, for every state $s \in S$,

$$\big|\,V_1^\pi(s) - V_2^\pi(s)\big| \;\;\le\;\; \|\delta\|_\infty \;\le\; \frac{\alpha \;+\; \gamma\,R_{\max}\,\beta}{(1-\gamma)^2}.$$

This is the desired result.

3 Policy Evaluation in RiverSwim

Here are provided the results of the Monte Carlo simulation and exact computation of V^π for the RiverSwim MDP, where the policy π takes *right* action with probability 0.65 in states $\{1, 2, 3\}$ and always takes *right* in states $\{4, 5\}$.

The code used to run the experiments is contained in `HA2_RiverSwim.py`.

(i) Monte Carlo Estimation of V^π

I generated $n = 50$ trajectories of length $T = 300$ each, for each possible start state, accumulating returns

$$G = \sum_{t=0}^{T-1} \gamma^t r_t,$$

and then averaged over the simulated trajectories to obtain an approximate value $V^\pi(s)$. I used the discount factor $\gamma = 0.96$. Below are the resulting Monte Carlo estimates (in a few seconds of execution time):

State	**MC Estimate**	**Exact Value**
1	4.01793	4.12090
2	4.61536	4.71119
3	5.96556	6.33460
4	9.47802	9.73803
5	11.21253	11.17784

Table 1: Comparison of Monte Carlo estimates vs. exact values.

(ii) Exact Computation using the Bellman Equation

I also computed the exact value function

$$V^\pi = \left(I - \gamma P^\pi\right)^{-1} r^\pi$$

by constructing the transition matrix P^π and reward vector r^π under policy π, and then numerically solving the linear system $(I - \gamma P^\pi) v = r^\pi$ in Python with `numpy.linalg.solve`. The table above (right column) presents the resulting exact values of $V^\pi(s)$ for $s = 1, \ldots, 5$.

Although the Monte Carlo estimates are not the finest approximation and slightly deviate from the exact values (particularly in states 3 and 4) due to limited sample size, they were obtained in just a few seconds of execution. Increasing the number of trajectories (or their length) would make the approximation even closer in practice.

4 Solving a Discounted Grid-World

We have a 7×7 grid with walls, yielding 20 accessible states labeled 0 to 19. State 19 (bottom-right corner) has reward 1 and is effectively absorbing once reached. The agent chooses from 4 compass actions (`0` = Up, `1` = Right, `2` = Down, `3` = Left), but transitions are *slippery*: probability 0.7 for the chosen direction, 0.1 each for perpendicular directions, and 0.1 for staying in place. We set $\gamma = 0.97$ by default, except in part (iii) where $\gamma = 0.998$.

The Python code used to run these experiments (i.e. PI, VI, and Anchored VI, plus the 4-room environment and the visualizations) can be found in the file `HA2_gridworld.py`.

(i) Solve the grid-world task using PI (Policy Iteration).

I implemented *Policy Iteration*, which alternates:

$$\text{Policy Iteration:} \quad \begin{cases} \text{(a) Evaluate current policy } \pi: & V^\pi = (I - \gamma P^\pi)^{-1} r^\pi, \\ \text{(b) Improve } \pi: & \pi \leftarrow \arg\max_a \Big[r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^\pi(s') \Big]. \end{cases}$$

It converged in 5 iterations. The resulting *optimal policy* (array of size 20) is

$$\pi_{\text{PI}} = [1, 1, 1, 2, 2, 2, 0, 2, 3, 2, 2, 1, 1, 1, 1, 2, 1, 0, 1, 0].$$

The *optimal value function* $V^*(s)$, rounded to 2 decimals, is

$$V^* = [23.07, 23.97, 25.15, 26.26, 25.40, 23.97, 23.07, 27.71, 26.40, 25.15, 29.12, 26.26, 27.71, 29.12, 30.40, 31.74, 25.40, 26.40, 31.74, 33.33].$$

We can visualize π on the 7×7 grid as:

```
[ [Wall, Wall, Wall, Wall, Wall, Wall, Wall],
  [Wall, Right, Right, Right, Down, Down, Wall],
  [Wall, Down, Up, Wall, Down, Left, Wall],
  [Wall, Down, Wall, Wall, Down, Wall, Wall],
  [Wall, Right, Right, Right, Right, Down, Wall],
  [Wall, Right, Up, Wall, Right, Up, Wall],
  [Wall, Wall, Wall, Wall, Wall, Wall, Wall] ].
```

(ii) Implement VI and use it to solve the grid-world task.

I then implemented *Value Iteration*, repeatedly applying

$$V_{n+1}(s) = \max_a \Big[r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V_n(s') \Big].$$

With $\gamma = 0.97$, VI converged in 48 iterations and recovered *the same* optimal policy and value function as in part (i). The same map display applies.

(iii) Repeat (ii) with $\gamma = 0.998$.

Now we let $\gamma = 0.998$. VI converged more slowly (56 iterations). The *value function* is larger (about 487–500 in some states), and the policy is slightly modified in some early states. We still see a path leading to state 19. Overall, it remains optimal, just at a bigger scale of values.

Regarding the increasing number of iterations required for convergence, this occurs because when γ is larger (closer to 1), the updates contract the value function differences more slowly, so it typically takes *more* iterations to reduce $\|V_{n+1} - V_n\|_\infty$ below a given threshold. Intuitively, a higher γ means future rewards are weighted more heavily, causing bigger swings or slower settling of the estimated values across states, and thus lengthening the convergence process.

(iv) Anc-VI with different initial points.

I added an anchoring approach with:

$$V_{n+1} \;=\; \beta_{n+1}\, V_0 \;+\; (1 - \beta_{n+1})\, \max_a \{\dots\}.$$

Using three different anchors:

(a) $V_0 = 0$ (**614 iterations to converge**),

(b) $V_0 = \mathbf{1}$ (**613 iterations to converge**),

(c) random V_0 in $[0,\, 1/(1-\gamma)]^{nS}$ (**606 iterations to converge**),

they all converged to an *optimal policy* as well, matching the same final V^* observed before.

(v) Compare the convergence speed of VI vs. Anc-VI.

Finally, we can compare standard VI (`anc=False`) and the anchored variant (`anc=True`) for each available initialization of the values of V at time 0:

	$V_0 = 0$	$V_0 = 1$	$V_0 = \text{Random}$
Anchored VI	614 iters	613 iters	606 iters

Table 2: Number of iterations required by Anchored VI with different initial value functions.

	Iterations to converge
Standard VI	48 iters

Table 3: Number of iterations required by standard VI to converge.

In this experiment, the *anchored* method needed *more* iterations than the standard VI. However, all methods still converged to optimal policies. Depending on the scenario, anchored VI may either accelerate or delay convergence based on different discount factors or parameters.

5 Off-Policy Evaluation in Episode-Based River-Swim

In this exercise, we study off-policy evaluation in a 6-state "episode-based" RiverSwim environment with discount factor $\gamma = 0.96$. Our target policy π always chooses the "right" action (action 1). The behavior policy π_b from which the data was collected chooses action 0 with probability 0.35 and action 1 with probability 0.65. We wish to estimate $V^\pi(1)$, the discounted value of this target policy starting from state 1. The dataset `dataset0.csv` contains 200 episodes, each beginning in state 0 and ending in state 5 (terminal).

The code used to run the experiments is contained in `HA2_PolicyEval.py`.

(i) Estimating $V^\pi(s_{\text{init}})$ Using Four Methods

We employ four different estimators for $V^\pi(s_{\text{init}})$:

- **MB-OPE**: A model-based approach that estimates $P(s' \mid s, a)$ and $r(s, a)$ from data and solves
$$\left(I - \gamma P^\pi\right) V \ = \ r^\pi,$$
where $r^\pi(s) = \sum_a \pi(a \mid s)\, \hat{r}(s, a)$ and $P^\pi(s, s') = \sum_a \pi(a \mid s)\, \hat{P}(s' \mid s, a)$.

- **IS**: Plain trajectory-level Importance Sampling,
$$\widehat{V}^\pi_{\text{IS}}(s_{\text{init}}) \ = \ \frac{1}{n} \sum_{i=1}^{n} \Big(\rho^{(i)}_{1:T_i}\Big) \ \sum_{t=1}^{T_i} \gamma^{\,t-1}\, r^{(i)}_t,$$
where $\rho^{(i)}_{1:T_i} = \prod_{t=1}^{T_i} \frac{\pi(a^{(i)}_t | s^{(i)}_t)}{\pi_b(a^{(i)}_t | s^{(i)}_t)}$.

- **wIS**: Weighted Importance Sampling,
$$\widehat{V}^\pi_{\text{wIS}}(s_{\text{init}}) \ = \ \frac{\sum_{i=1}^{n} \rho^{(i)}_{1:T_i} \ \sum_{t=1}^{T_i} \gamma^{\,t-1}\, r^{(i)}_t}{\sum_{i=1}^{n} \rho^{(i)}_{1:T_i}}.$$

- **PDIS**: Per-Decision Importance Sampling,
$$\widehat{V}^\pi_{\text{PDIS}}(s_{\text{init}}) \ = \ \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \Big(\prod_{k=1}^{t} \frac{\pi(a^{(i)}_k | s^{(i)}_k)}{\pi_b(a^{(i)}_k | s^{(i)}_k)}\Big)\, \gamma^{\,t-1}\, r^{(i)}_t.$$

When we apply these four estimators to the data in `dataset0.csv`, we obtain the following estimates for $V^\pi(s_{\text{init}})$ (here, $s_{\text{init}} = 0$):

Method	**Estimate**
MB-OPE	1.3380
IS	0.4040
wIS	1.0000
PDIS	1.0180

Table 4: Estimates of $V^\pi(s_{\text{init}})$ on `dataset0.csv`.

In addition, I solved the linear system

$$\big(I - \gamma P^\pi\big)\, V \;=\; r^\pi$$

directly from the known RiverSwim model, which yields the "exact" solution $V^\pi(0) \approx 1.3186$. These numbers show that MB-OPE, wIS, and PDIS are reasonably close, while the plain IS estimate is further away (in this particular dataset).

(ii) Error Plots for Each Method

To visualize how these estimators improve with more data, we compute each estimator on the first k episodes for $k = 1, \ldots, 200$ and compare the result to the exact $V^\pi(s_{\text{init}})$. Figure 1 shows the absolute error as a function of k. As expected, the IS curve fluctuates heavily, whereas wIS and PDIS are more stable, and MB-OPE gradually converges to a value close to the true $V^\pi(s_{\text{init}})$.

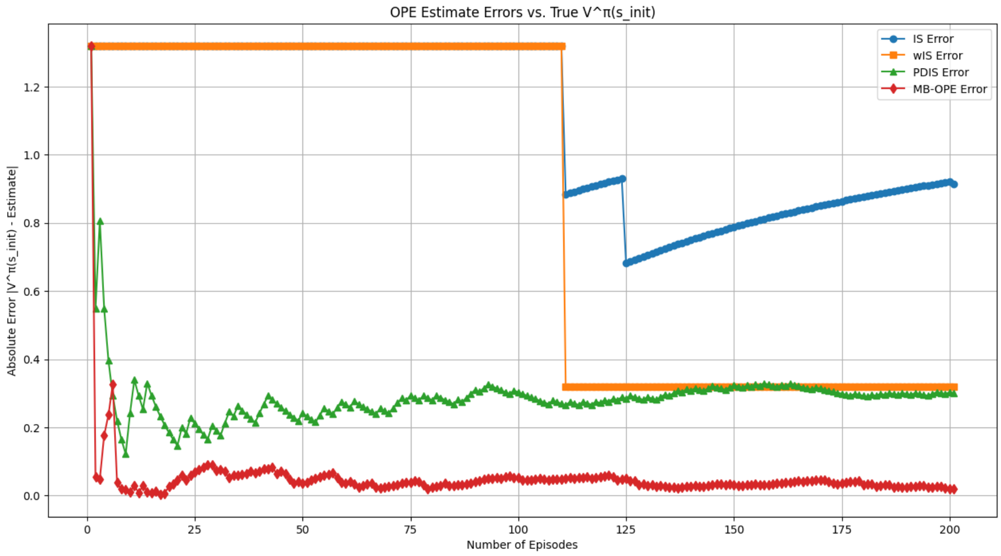

Figure 1: Absolute errors $\left|V^\pi(s_{\text{init}}) - \widehat{V}_k^\pi(s_{\text{init}})\right|$ versus the number of episodes k, for IS, wIS, PDIS, and MB-OPE.

(iii) Variance of Estimates Across 10 Datasets

We next consider 9 additional datasets (`dataset1.csv` through `dataset9.csv`), all generated in the same manner. For each dataset and each OPE method, we compute an estimate of $V^\pi(s_{\text{init}})$. In Table 5 we show the estimates for each method (left) and the resulting sample variance across the 10 datasets (right).

Method	Estimates across 10 Datasets	Variance
MB-OPE	[1.3380, 1.3306, 1.3544, 1.3080, 1.3625, 1.2955, 1.3351, 1.3210, 1.3079, 1.2866]	0.0006
IS	[0.4040, 0.9483, 0.0077, 7.9424, 0.0077, 0.0000, 0.4804, 0.0077, 0.2479, 0.3062]	5.3813
wIS	[1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.0000, 1.0000, 1.0000, 1.0000, 1.0000]	0.0900
PDIS	[1.0180, 1.0409, 0.9568, 0.9950, 1.0027, 0.9950, 0.9644, 0.9414, 1.0486, 0.9568]	0.0012

Table 5: Estimates of $V^\pi(s_{\text{init}})$ across 10 datasets (left) and sample variance (right).

These results corroborate the theoretical expectations:

- Plain IS exhibits the highest variance by far, and occasionally gives extreme estimates.
- Weighted IS is more stable but can still be off by a moderate amount.
- PDIS achieves lower variance and relatively small bias compared to IS and wIS.
- MB-OPE, though it depends on estimating a model from data, remains close to the true value and shows small variance in this setup.

(iv) Comparison in Terms of Error and Variance

Finally, Table 6 compares the methods by their mean absolute error (relative to the exact $V^\pi(s_{\text{init}})$) and the variance of those errors across the 10 datasets.

Method	**Mean Absolute Error**	**Variance of Errors**
MB-OPE	0.0206	0.0002
IS	1.6081	2.8755
wIS	0.4186	0.0900
PDIS	0.3266	0.0012

Table 6: Mean absolute error (relative to exact $V^\pi(s_{\text{init}})$) and variance of errors across 10 datasets.

Overall, the empirical results match the theoretical expectations described in the lecture slides: IS tends to have the highest variance, while wIS, PDIS, and MB-OPE produce estimates that are more stable and generally closer to the exact value.