

Online and Reinforcement Learning (2025–2026)

Home Assignment 2

Hamsa Mohamed (XXXXXXXXX)
your@email.com

Deadline: 18 February 2026, 20:59

Submission checklist

- PDF report with detailed answers (no full code dumps).
- Separate `.zip` with well-commented source code and a main file per question (or one main).

Contents

1	Short Questions (8 points)	2
2	Introduction of New Products (25 points)	2
2.1	Problem statement	2
2.2	Solution	2
3	Empirical Comparison of FTL and Hedge (25 points)	2
3.1	Experimental setup	2
3.2	Methods	3
3.3	Results	3
3.4	Discussion	3
4	Value Function Bounds (22 points)	3
4.1	(i) Proof of the inequality	3
4.2	(ii) Interpretation	4
5	Solving a Discounted Grid-World (20 points)	4
5.1	Environment summary	4
5.2	(i) Policy Iteration (PI)	4
5.3	(ii) Value Iteration (VI) with $\varepsilon = 10^{-6}$	4
5.4	(iii) VI with $\gamma = 0.998$ and convergence discussion	5
A	Reproducibility notes (optional)	5

1 Short Questions (8 points)

(1) In a finite discounted MDP, every possible policy induces a Markov Reward Process.

Answer: ☐ True ☒ False

Justification: Missing

(2) 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: ☐ True ☒ False

Justification: Missing

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

Answer: ☐ True ☒ False

Justification: Missing

(4) Policy Iteration (PI) may return a near-optimal policy.

Answer: ☐ True ☒ False

Justification: Missing

2 Introduction of New Products (25 points)

2.1 Problem statement

Briefly restate Exercise 7.6 in your own words (1–3 lines), then proceed.

2.2 Solution

Result

Final answer (clean statement):

Write the final bound / algorithm / conclusion clearly here.

3 Empirical Comparison of FTL and Hedge (25 points)

3.1 Experimental setup

- Dataset / synthetic setting:
- Losses / feedback model:
- Horizon T , number of experts N :
- Learning rate / tuning:
- Random seeds / repetitions:

3.2 Methods

FTL. Describe the update briefly.

Hedge. Describe the update briefly (weights, learning rate, normalization).

3.3 Results



Figure 1: Example: Regret curves for FTL vs Hedge (fill in).

3.4 Discussion

Explain what you observe and why (a few paragraphs).

4 Value Function Bounds (22 points)

4.1 (i) Proof of the inequality

Given. Let $M_1 = (\mathcal{S}, \mathcal{A}, P_1, R_1, \gamma)$ and $M_2 = (\mathcal{S}, \mathcal{A}, P_2, R_2, \gamma)$ be finite discounted MDPs, with rewards in $[0, 1]$ and for all (s, a) :

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

Goal: show that for any stationary deterministic policy π and any (s, a) ,

$$|Q_1^\pi(s, a) - Q_2^\pi(s, a)| \leq \frac{\alpha}{1 - \gamma} + \frac{\gamma\beta}{(1 - \gamma)^2}.$$

Proof.

4.2 (ii) Interpretation

Explain in words: how value differences scale with reward mismatch and transition mismatch, and why the factor blows up as $\gamma \rightarrow 1$.

5 Solving a Discounted Grid-World (20 points)

5.1 Environment summary

- Grid size: 7×7 , accessible states: 20 (after walls).
- Start: upper-left (red). Reward state: lower-right (yellow), absorbing with reward 1 forever.
- Actions: up/left/down/right. Slippery: intended 0.7, stay 0.1, perpendiculars 0.1 each.
- Discount: $\gamma = 0.97$ (then $\gamma = 0.998$ for part (iii)).

5.2 (i) Policy Iteration (PI)

Implementation notes. State what code you used (file name), and any parameters.

Optimal policy and V^* . Include:

- A visualization of the policy (arrows in a matrix).
- A visualization/table of V^* .

5.3 (ii) Value Iteration (VI) with $\varepsilon = 10^{-6}$

Algorithm 1 Value Iteration (VI)

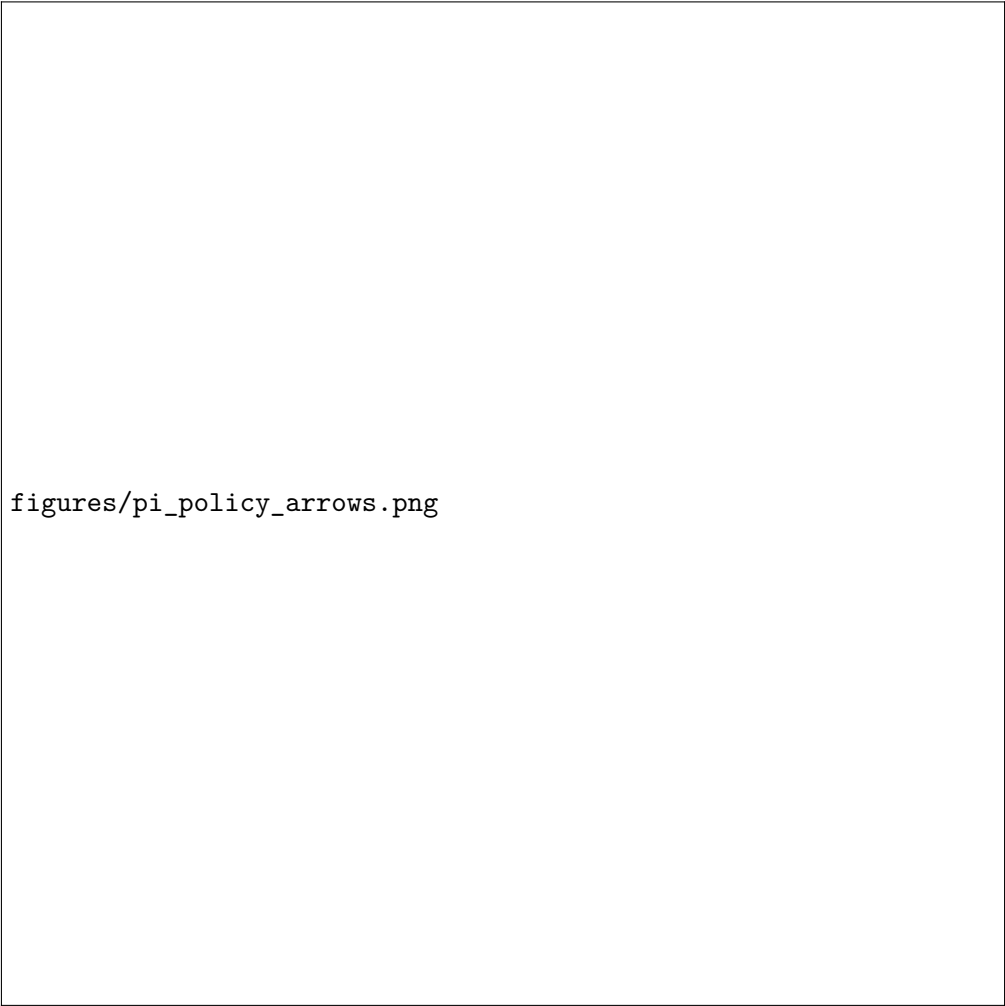
```

1: Input: MDP  $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ , accuracy  $\varepsilon$ 
2: Initialize  $V_0(s) \leftarrow 0$  for all  $s$ 
3: repeat
4:   for each state  $s$  do
5:      $V_{\text{new}}(s) \leftarrow \max_{a \in \mathcal{A}} (R(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s'))$ 
6:   end for
7:    $\Delta \leftarrow \max_s |V_{\text{new}}(s) - V(s)|$ 
8:    $V \leftarrow V_{\text{new}}$ 
9: until  $\Delta < \varepsilon$ 
10: Extract policy  $\pi(s) \in \arg \max_a (R(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s'))$ 
11: Output:  $\pi, V$ 

```

Algorithm.

Results. Show the VI policy and value function (like in part (i)) and confirm it matches PI.



figures/pi_policy_arrows.png

Figure 2: Optimal policy from PI (arrows).

5.4 (iii) VI with $\gamma = 0.998$ and convergence discussion

Report:

- Number of iterations to converge (or runtime) for $\gamma = 0.97$ vs $\gamma = 0.998$.
- A short explanation of why higher γ slows convergence (contraction factor).

Optional: Improved Parametrization of UCB1 (0 points)

If you solved it, place it here. Otherwise omit this section in your final PDF.

A Reproducibility notes (optional)

- How to run: `python main_q3.py` / `python main_q5.py`
- Dependencies: `requirements.txt`
- Random seeds used:



Figure 3: Optimal value function V^* from PI.