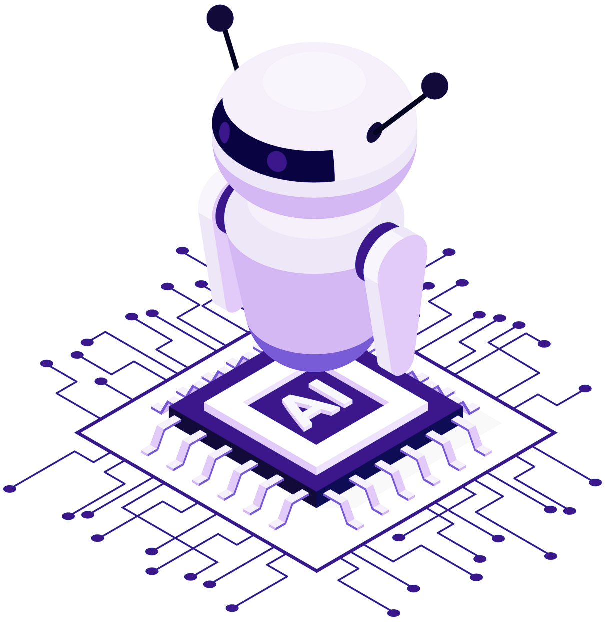

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 7:

Value-Based Theory

Designed By:

Ghazal Hosseini Mighan
ghazaldesu@gmail.com

Soroush Vafaie Tabar
svafaiet@gmail.com

Spring 2025

Preface

Welcome to the homework!

Reinforcement Learning (RL) is a powerful framework for training agents to make optimal decisions through trial and error. At its core lies the fundamental concept of value functions, which estimate the long-term utility of states and actions. This assignment delves into the theoretical foundations of value-based methods, focusing on their convergence properties and practical trade-offs.

In previous assignments, you explored practical implementations of RL algorithms. Now, we shift our focus to the mathematical underpinnings of two cornerstone algorithms:
1. Value Iteration: An iterative approach that directly approximates the optimal value function
2. Policy Iteration: An alternative method that alternates between policy evaluation and improvement

The insights you gain from these theoretical foundations will prove invaluable as you progress to more advanced RL concepts. The mathematical understanding you develop here forms the essential basis for modern value-based deep reinforcement learning systems.

As you work through these problems, we emphasize the importance of crafting clear, rigorous mathematical proofs.

Good luck, and enjoy exploring the beautiful theory behind value-based reinforcement learning!

Grading

The grading will be based on the following criteria, with a total of 100 points:

Section	**Points**
Positive Rewards	15
General Rewards	10
Policy Turn	25
Bellman Operators	15
Bellman Residuals	35
Bonus 1: Writing your report in Latex	5
Bonus 2: Question 2.2.11	5

Submission

The deadline for this homework is 1404/2/21 (May 11th 2025) at 11:59 PM.

Please submit your work by following the instructions below:

- Your written solution must be a single PDF file named `HW7_Solution.pdf`.
- Zip all the files together with the following naming format:

  `DRL_HW7_[StudentNumber]_[FullName].zip`

  - Replace `[FullName]` and `[StudentNumber]` with your full name and student number, respectively. Your `[FullName]` must be in CamelCase with no spaces.
- Submit the zip file through Quera in the appropriate section.
- We provided this LaTeX template for writing your homework solution. There is a 5-point bonus for writing your solution in LaTeX using this template and including your LaTeX source code in your submission, named `HW7_Solution.zip`.
- If you have any questions about this homework, please ask them in the Homework section of our Telegram Group.
- If you are using any references to write your answers, consulting anyone, or using AI, please mention them in the appropriate section. In general, you must adhere to all the rules mentioned here and here by registering for this course.

Keep up the great work and best of luck with your submission!

Contents

1 Iteration Family

Let $M = (S, A, R, P, \gamma)$ be a finite MDP with $|S| < \infty$, $|A| < \infty$, bounded rewards $|R(s, a)| \leq R_{\text{max}}$ $\forall(s, a)$, and discount factor $\gamma \in [0, 1)$. In this section, we will first explore an alternative proof approach for the value iteration algorithm, then we cover policy iteration which is discussed in the class more precisely.

1.1 Positive Rewards

Assume $R(s, a) \geq 0$ for all s, a.

1. Derive an upper bound for the optimal k-step value function V_k^*.
2. Prove V_k^* is non-decreasing in k. Giving a policy π such that:

$$V_{k+1}^{\pi} \geq V_k^*.$$

   Use this to show convergence of Value Iteration to a solution satisfying the Bellman equation.

3. By taking the limit in the Bellman equation, prove that the V^* is optimal.

1.2 General Rewards

Remove the non-negativity constraint on $R(s, a)$. Assume no terminating states exist. Consider a new MDP defined by adding a constant reward r_0 to all rewards of the current MDP. That is, for all (s, a), the new reward is:

$$\hat{R}(s, a) = R(s, a) + r_0$$

4. By deriving the optimal action and V_k^* in terms of the original MDP's values and r_0, show that Value Iteration still converges to the optimal value function V^* (and optimal policy) of the original MDP even if rewards are negative. Also compute the new value V^*.
5. Why is it necessary to assume the absence of a terminating state? Try to explain with a counterexample.

1.3 Policy Turn

In this part we want to dive into the mathematical proof of policy iteration.

6. Let π_k be the policy at iteration k. Prove the following:

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s) \quad \forall s \in S,$$

   with strict inequality for at least one state unless π_k is already optimal. Use the definition of the greedy policy and explain why policy improvement leads to a better or equal value function.

7. Prove that Policy Iteration always converges to the optimal policy in a finite MDP. Specifically, show that after a finite number of policy evaluations and improvements, the algorithm reaches a policy π^* that satisfies the Bellman optimality equation. You may use theorems discussed in class, but if a result was not proven, please provide a full justification.

8. Prove that Value Iteration and Policy Iteration both converge to the same optimal value function V^*, even if the policies may differ. How the policies are still optimal despite possible differences?

9. Compare and contrast the computational cost of one step of Policy Iteration (i.e., full Policy Evaluation + Policy Improvement) versus one iteration of Value Iteration.

10. In the context of a (MDP) with an infinite horizon, when the discount factor $\gamma = 1$, analyze how both Value Iteration and Policy Iteration behave.

2 Bellman or Bellwoman

[1] Recall that a value function is a $|S|$-dimensional vector where $|S|$ is the number of states of the MDP. When we use the term V in these expressions as an "arbitrary value function", we mean that V is an arbitrary $|S|$-dimensional vector which need not be aligned with the definition of the MDP at all. On the other hand, V^π is a value function that is achieved by some policy π in the MDP. For example, say the MDP has 2 states and only negative immediate rewards. $V = [1, 1]$ would be a valid choice for V even though this value function can never be achieved by any policy π, but we can never have a $V^\pi = [1, 1]$. This distinction between V and V^π is important for this question and more broadly in reinforcement learning.

2.1 Bellman Operators

In the first part of this problem, we will explore some general and useful properties of the Bellman backup operator. We know that the Bellman backup operator B, defined below, is a contraction with the fixed point as V^*, the optimal value function of the MDP. The symbols have their usual meanings. γ is the discount factor and $0 \leq \gamma < 1$. In all parts, $\|v\| = \max_s |v(s)|$ is the infinity norm of the vector.

$$(BV)(s) = \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s') \right]$$

We also saw the contraction operator B^π with the fixed point V^π, which is the Bellman backup operator for a particular policy given below:

$$(B^\pi V)(s) = r(s,\pi(s)) + \gamma \sum_{s' \in S} p(s'|s,\pi(s))V(s')$$

In this case, we'll assume π is deterministic, but it doesn't have to be in general. You have seen that $\|BV - BV'\| \leq \gamma \|V - V'\|$ for two arbitrary value functions V and V'.

1. Show that the analogous inequality, $\|B^\pi V - B^\pi V'\| \leq \gamma \|V - V'\|$, holds.
2. Prove that the fixed point for B^π is unique. Recall that the fixed point is defined as V satisfying $V = B^\pi V$. You may assume that a fixed point exists.
3. Suppose that V and V' are vectors satisfying $V(s) \leq V'(s)$ for all s. Show that $B^\pi V(s) \leq B^\pi V'(s)$ for all s. *Note: all of these inequalities are elementwise.*

2.2 Bellman Residuals

We can extract a greedy policy π from an arbitrary value function V using the equation below:

$$\pi(s) = \arg\max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s') \right]$$

It is often helpful to know what the performance will be if we extract a greedy policy from an arbitrary value function. To see this, we introduce the notion of a Bellman residual.

Define the Bellman residual to be $(BV - V)$ and the Bellman error magnitude to be $\|BV - V\|$.

4. For what value function V does the Bellman error magnitude $\|BV - V\|$ equal 0? Why?
5. Prove the following statements for an arbitrary value function V and any policy π.

$$\|V - V^\pi\| \leq \frac{\|V - B^\pi V\|}{1-\gamma}$$

$$\|V - V^*\| \leq \frac{\|V - BV\|}{1-\gamma}$$

6. Let V be an arbitrary value function and π be the greedy policy extracted from V. Let $\varepsilon = \|BV - V\|$ be the Bellman error magnitude for V. Prove the following for any state s.

$$V^\pi(s) \geq V^*(s) - \frac{2\varepsilon}{1-\gamma}$$

7. Give an example real-world application or domain where having a lower bound on $V^\pi(s)$ would be useful.
8. Suppose we have another value function V' and extract its greedy policy π'. $\|BV' - V'\| = \varepsilon = \|BV - V\|$. Does the above lower bound imply that $V^\pi(s) = V^{\pi'}(s)$ at any s?

Say $V \leq V'$ if $\forall s$, $V(s) \leq V'(s)$.
What if our algorithm returns a V that satisfies $V^* \leq V$? I.e., it returns a value function that is better than the optimal value function of the MDP. Once again, remember that V can be any vector, not necessarily achievable in the MDP, but we would still like to bound the performance of V^π where π is extracted from said V. We will show that if this condition is met, then we can achieve an even tighter bound on policy performance.

9. Using the same notation and setup as part 5, if $V^* \leq V$, show the following holds for any state s. *Recall that for all π, $V^\pi \leq V^*$ (why?)*

$$V^\pi(s) \geq V^*(s) - \frac{\varepsilon}{1-\gamma}$$

Intuition: A useful way to interpret the results from parts (8) and (9) is based on the observation that a constant immediate reward of r at every time-step leads to an overall discounted reward of

$$r + \gamma r + \gamma^2 r + \cdots = \frac{r}{1-\gamma}$$

Thus, the above results say that a state value function V with Bellman error magnitude ε yields a greedy policy whose reward per step (on average), differs from optimal by at most 2ε. So, if we develop an algorithm that reduces the Bellman residual, we're also able to bound the performance of the policy extracted from the value function outputted by that algorithm, which is very useful!

10. It's not easy to show that the condition $V^* \leq V$ holds because we often don't know V^* of the MDP. Show that if $BV \leq V$ then $V^* \leq V$. Note that this sufficient condition is much easier to check and does not require knowledge of V^*.

    Hint: Try to apply induction. What is $\lim_{n\to\infty} B^n V$?

11. (Bonus) It is possible to make the bounds from parts (9) and (10) tighter. Let V be an arbitrary value function and π be the greedy policy extracted from V. Let $\varepsilon = \|BV - V\|$ be the Bellman error magnitude for V. Prove the following for any state s:

$$V^{\pi}(s) \geq V^{*}(s) - \frac{2\gamma\varepsilon}{1-\gamma}$$

Further, if $V^{*} \leq V$, prove for any state s

$$V^{\pi}(s) \geq V^{*}(s) - \frac{\gamma\varepsilon}{1-\gamma}$$

References

[1] Baesed on CS 234: Reinforcement Learning, Stanford University. Spring 2024.

[2] Cover image designed by freepik