Memorial University

REINFORCEMENT LEARNING:

ASSIGNMENT 2

(Spring 2025)

by

Hoang Phi Yen Duong - 202396817

Tinh Thanh Bui - 202398612

(PhD students)

St. John's, July 25, 2025

Outline

Code is available on GitHub: Click here

PART 1:

In this part, we consider a 5 x 5 gridworld problem shown in Fig. 1. The gridworld environment consists of 25 cells (possible states). An agent can take a step up, down, left, or right. The agent gets a reward of 5 and jumps to the red square if any action is taken at the blue square. Meanwhile, the agent get a reward of 2.5 and jumps to either the red square or the yellow square with probability of 0.5.

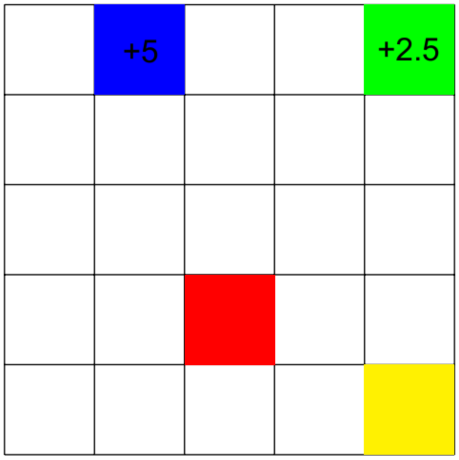

Fig. 1. *A simple 5 x 5 gridworld problem*

1.1. Estimate the value function for each of the states using solving the system of Bellman equations explicitly

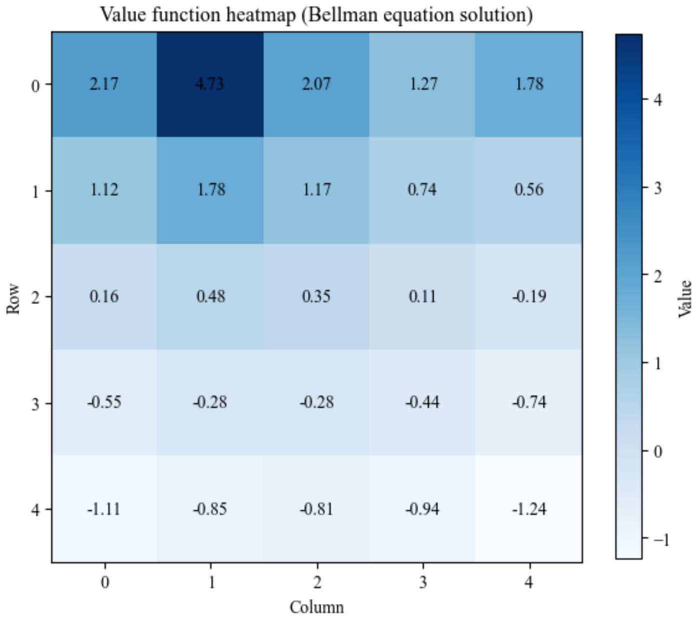

Fig. 1. *Heatmap of value function using Bellman equations solution*

The heatmap in Fig. 1 shows the value function of the given problem by solving the system of Bellman equations. The agent chooses among the four directions (i.e. up, down, left, and right) with equal probability of 0.25. Each grid cell represents the expected cumulative reward starting from that state. Notably, the state at position (0, 1) corresponding to the **blue square** holds the highest value of approximately 4.73. The reason is that any action is performed at the blue square obtains an immediate reward of 5 and transports the agent to the red square at (3, 2). The value at the green square (0, 4) equals to 1.78 slightly higher than those of its adjancent squares due to the immediate reward of 2.5. This value is 0.72 lower than 2.5 due to there is 50% chance to jump to the yellow square which has lowest value of -1.24. The bottom-left and bottom right corners show negative values, reflecting the off-grid penalties. Overall, solving the Bellman equations exactly provides a reliable estimate of long-term returns under current policy and captures the reward structure and transition dynamics of the grid world environment.

1.2. Estimate the value function for each of the states using iterative policy evaluation

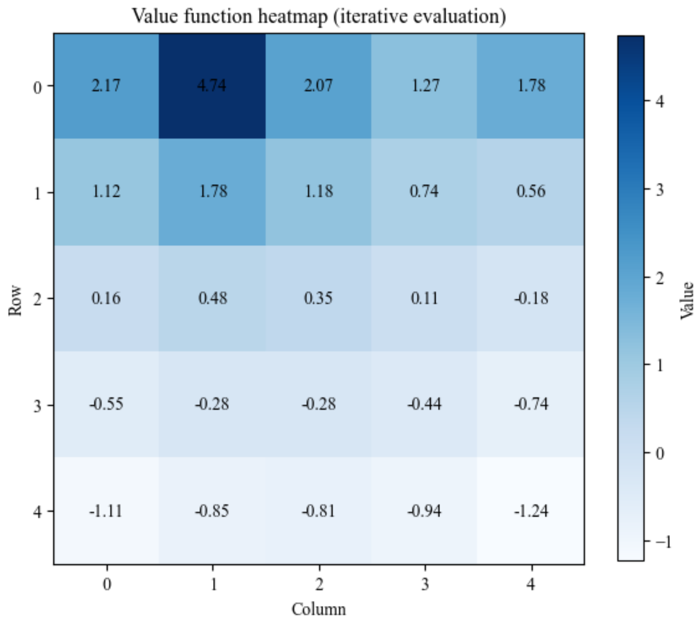

Fig. 2. *Heatmap of value function (iterative policy evaluation)*

The heatmap in Fig. 2 visualizes the value function obtained through iterative policy evaluation. As shown in the figure, the state at position (0, 1) corresponding to the **blue square** achieves the highest value of approximately 4.74. This outcome is nearly equal to the previous results from the Bellman equation method and is expected due to the immediate reward of 5 at the blue square and its deterministic transition to the red square at position (3, 2). Other states have lower values reflecting reduced or negative rewards. The value at the green square (0, 4) equals to 1.78 slightly higher than those of its adjancent squares due to the immediate reward of 2.5. The iterative policy evaluation techniques works efficiently in capturing the value function by gradually refining the estimates through repeated updates. This method yields closely the same value function compared to the method of solving Bellman equations explicitly.

- **Does this surprise?** The value of the state at the blue square (0, 1) is highest in both two cases of solving Bellman equations explicitly and iterative policy evaluation. This result is expected due to the given reward structure of the grid world problem. The blue square offers an immediate reward of 5 and jumps to the red square (3, 2) which has low chance of penalty by stepping off the grid. Meanwhile, the green square yields a reward of 2.5 but has 50% chance of jumping to the yellow square which has high chance of stepping off the grid due to on the corner.

2.1. The optimal policy for the gridworld problem by explicitly solving the Bellman optimality equation

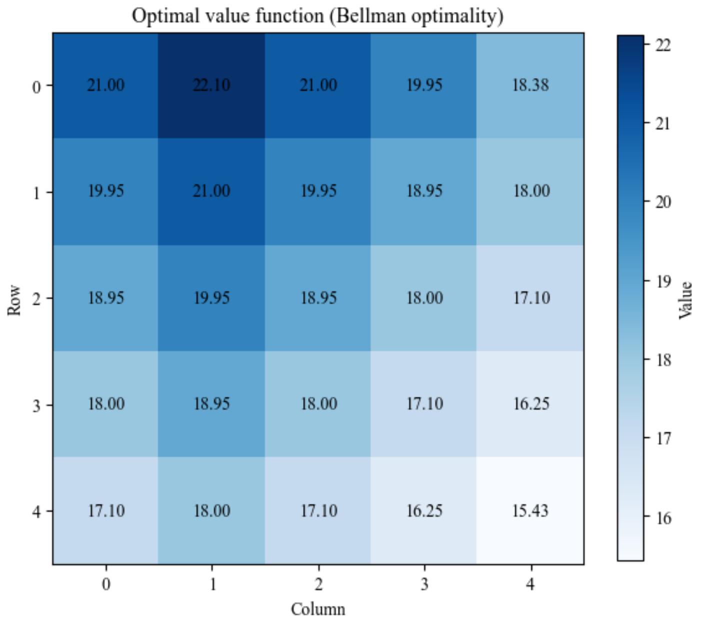

***Fig. 3.** Heatmap of value function (Bellman optimality equation)*

The optimal policy is expressed as

→	↑	←	←	↑
↑	↑	↑	↑	←
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑

The heatmap in Fig. 3 represents the optimal value function obtained by solving the Bellman optimality equation, revealing the highest values concentrated in the upper row of the grid, particularly around (0, 1) and (0, 4) i.e. the blue and green squares. These states provide immediate rewards of 5 and 2.5, respectively. As we move downward and to the right, the values gradually decrease due to increasing distance from the rewarding squares, the influence of the discount factor and the penalty of stepping off the grid. This creates a gradient of blue colors across the grid, with darker colors indicating high-value squares and lighter tones marking lower-value squares.

The associated policy arrows reveal the optimal policy. From state (0, 0), the policy directs the agent rightward toward the blue square, exploiting its high reward. On the right hand side of the blue square, the policy suggests the agent take a step left to get more reward instead of choosing the green square. Throughout the remaining rows, the policy consistently points upward, encouraging the agent to return to the upper states. With this optimal policy, the green square is not recommended because of left-action recommendation in its both adjacent squares (0, 3) and (1, 4). This behavior highlights how the optimal policy emerges from the value function, effectively steering the agent toward actions that maximize long-term cumulative rewards.

2.2. The optimal policy for the gridworld problem using policy iteration with iterative policy evaluation

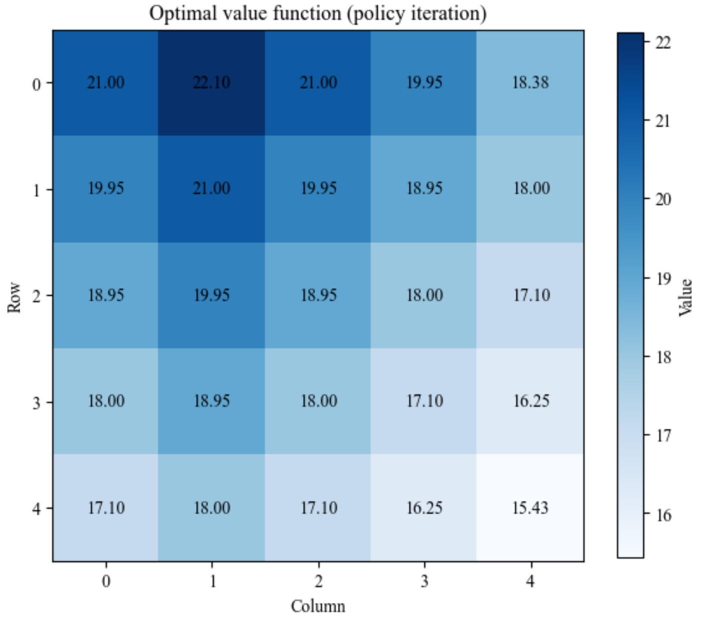

Fig. 4. *Heatmap of value function (iterative policy evaluation)*

The optimal policy is expressed as

→	↑	←	←	↑
↑	↑	↑	↑	←
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑

Using policy iteration with iterative policy evaluation, we observe that the highest values are again concentrated in the top-left region, particularly in the first two rows of the grid. This pattern is influenced with the immediate rewards from the blue square at (0, 1) and the green square at (0, 4), which effects propagate across nearby states through value updates. To the bottom-right, the values gradually decrease due to the increasing distance from high-reward states and the cumulative effect of the discount factor. The heatmap in Fig. 4 clearly visualizes this gradient, with darker blue colors indicating high-value squares and brighter colors marking squares of lower expected returns.

The optimal policy reflects a strategic and structured decision-making process. Starting from (0, 0), the agent is directed rightward toward the blue square at (0, 1) to collect its immediate reward. From there, the policy leads the agent to the red square at (3, 2). The policy then guides the agent through a sequence of leftward movements across the top row from (0, 2) to (0, 4) followed by an upward movement at the end. In the second row, the final move is rightward at point (1, 4), while in the remaining rows, the policy consistently recommends upward movements. This behavior reflects an optimal strategy where the agent seeks to return to higher-value states above, minimizing time spent in low-reward states. Overall, the policy and value function learned through this method demonstrate the agent's ability to leverage high-reward states and structured transitions, guiding it along an efficient path that maximizes long-term cumulative rewards.

2.3. The optimal policy for the gridworld problem by policy improvement with value iteration

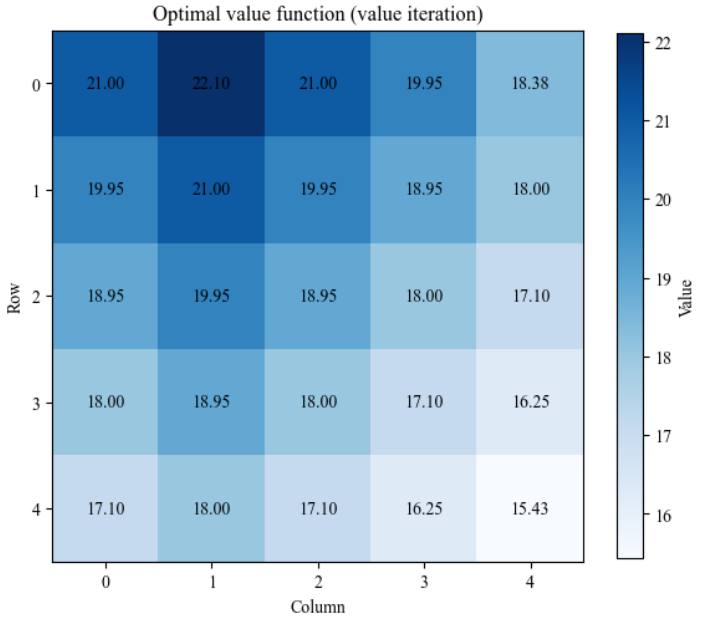

Fig. 5. *Heatmap of value function (policy improvement with value iteration)*

The optimal policy is expressed as

→	↑	←	←	↑
↑	↑	↑	↑	←
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑
↑	↑	↑	↑	↑

Using value iteration, we observe that the highest values are again concentrated in the top-left region, particularly in the first two rows of the grid. This pattern is influenced with the immediate rewards from the blue square at (0, 1) and the green square at (0, 4), which effects propagate across nearby states through value updates. To the bottom-right, the values gradually decrease due to the increasing distance from high-reward states and the cumulative effect of the discount factor. The heatmap in Fig. 5 clearly visualizes this gradient, with darker blue colors indicating high-value squares and brighter blue colors marking squares of lower expected returns.

The optimal policy reflects a strategic and structured decision-making process and is the same with the optimal policy received when using policy iteration with iterative policy evaluation. Starting from (0, 0), the agent is directed rightward toward the blue square at (0, 1) to collect its immediate reward. From there, the policy leads the agent to the red square at (3, 2). The policy then guides the agent through a sequence of leftward movements across the top row from (0, 2) to (0, 4) followed by an upward movement at the end. In the second row, the final move is rightward at point (1, 4), while in the remaining rows, the policy consistently recommends upward movements. The green square is not recommended if the agent uses this optimal policy. This behavior reflects an optimal strategy where the agent seeks to return to higher-value states above, minimizing time spent in low-reward states. Overall, the policy and value function learned through this method demonstrate the agent's ability to leverage high-reward states and structured transitions, guiding it along an efficient path that maximizes long-term cumulative rewards.

- All three methods yield value functions with the highest values near the top left, particularly around the blue square (0,1) and the green square (0, 4). The values gradually decrease towards the bottom of the grid due to lower expected cumulative rewards for far states from the high reward squares and the discount factor. The derived policies generally suggest the agent to take actions towards the blue square which yields immediate reward of 5, while the tendency toward the green square is not recommended due to 50% chance of jumping to the yellow square on the corner of the grid (high chance of stepping off grid and far from high reward squares).

PART 2:

In this part, we consider a 5 x 5 gridworld problem shown in Fig. 6. The gridworld environment consists of 25 cells (possible states). An agent can take a step up, down, left, or right. The agent gets a reward of 5 and jumps to the red square if any action is taken at the blue square. Meanwhile, the agent get a reward of 2.5 and jumps to either the red square or the yellow square with probability of 0.5. Three terminal states represented as the black squares are added to the grid at (2, 0), (4, 0), and (2, 4). Any move from a white/yellow/red squares yields a reward of -0.2, if the step-off move is performed, the agent gets a reward of -0.5.

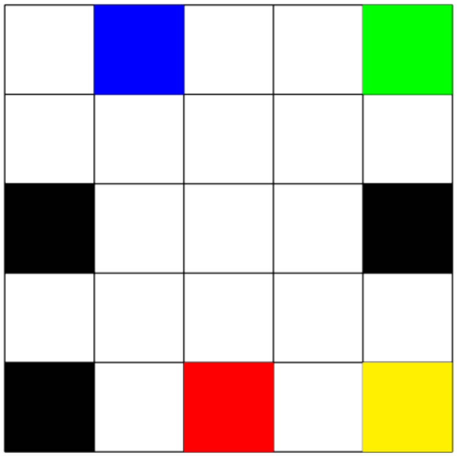

Fig. 6. *A simple gridworld problem with three terminal states as black squares*

1.1. Use the Monte Carlo method with exploring starts

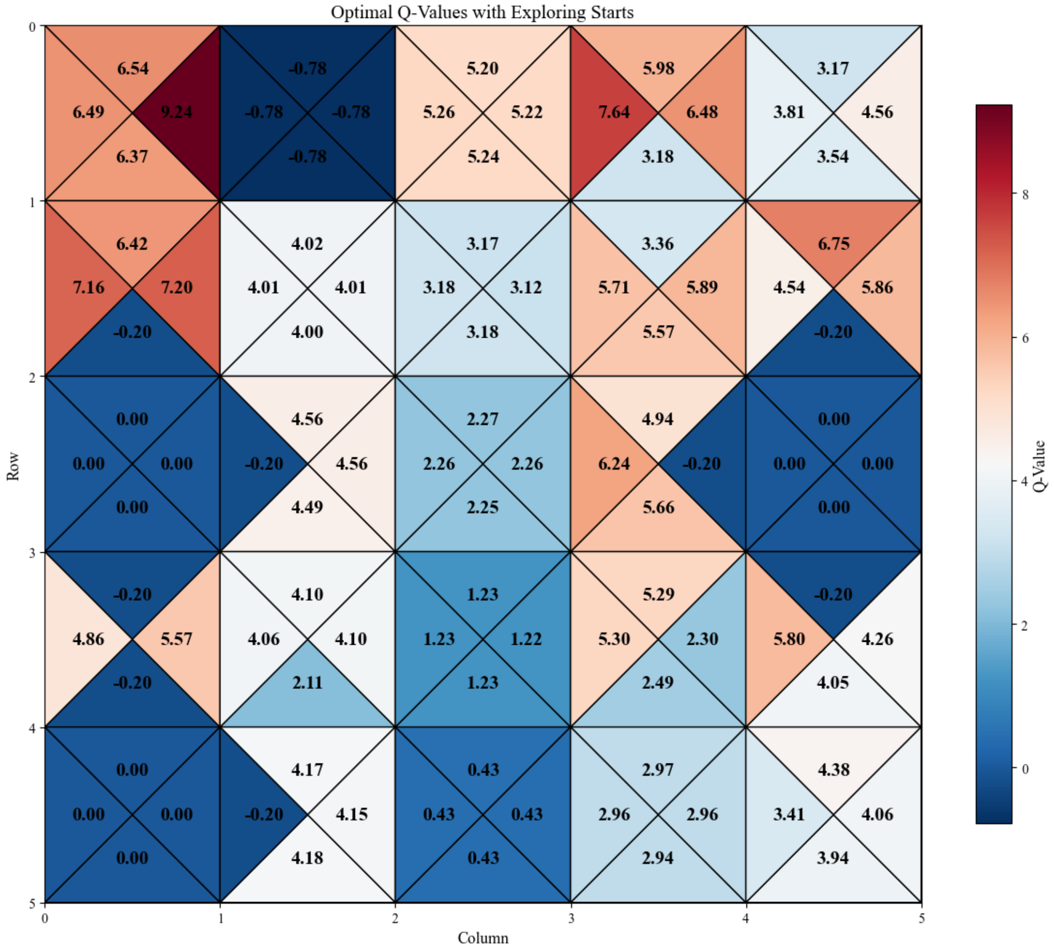

Fig. 7. *Optimal Q-Values with Exploring Starts*

Using the Monte Carlo method with exploring starts, the optimal policy is expressed as

→	↑	←	←	→
→	↑	←	→	↑
T	↑	↑	←	T
→	↑	↑	←	←
T	↓	↑	↑	↑

The optimal policy learned through the Monte Carlo method with exploring starts demonstrates a clear and strategic effort to direct the agent toward high-reward states, most notably the blue square at (0,1) and the green square at (0,4). These squares provide immediate rewards of 5 and 2.5, respectively, along with transitions to (4,2) and (4,4). The policy accurately reflects this by guiding movements toward these states, for instance, early rightward movements in the first column steer the agent directly to the blue and green squares to exploit their high returns. Differ from the optimal policy in Part 1 (sections 2.1, 2.2, 2.3), the green square is now recommended since a penalty of -0.2 is applied in this environment, requiring the agent to find near high reward square instead of spending many intermediate steps.

A notable pattern in the policy is the frequent upward movement across rows 2 to 4. This suggests a strategy aimed at re-engaging with high-value areas in the top rows or efficiently transitioning to stay away from terminal states. For example, in row 4, upward actions dominate, allowing the agent to quickly return to the top of the grid where immediate rewards are more accessible. Around all terminal squares, there is no action directing the agent toward terminal states. Additionally, rows 1 show a mix of upward and lateral movements, balancing reward-seeking, which helps limit unnecessary or penalty-incurring steps. The policy also demonstrates a level of efficiency and adaptability in navigating the grid. Rather than wandering through low-value white squares, the agent follows a calculated path that maximizes long-term returns while reducing exposure to negative rewards. This is especially evident in the strategic choices to reach terminal states like (2,0), (2,4), and (4,0) when no better options remain.

Overall, the learned policy aligns well with theoretical expectations for Monte Carlo method in gridworld environments. It reflects a strong understanding of the reward structure and dynamics of the environment. The Monte Carlo exploring starts approach has proven effective in discovering this optimal strategy, enabling the agent to consistently select actions that maximize expected returns and minimize penalties over time.

1.2. Use the Monte Carlo method without exploring starts but the ε-soft approach

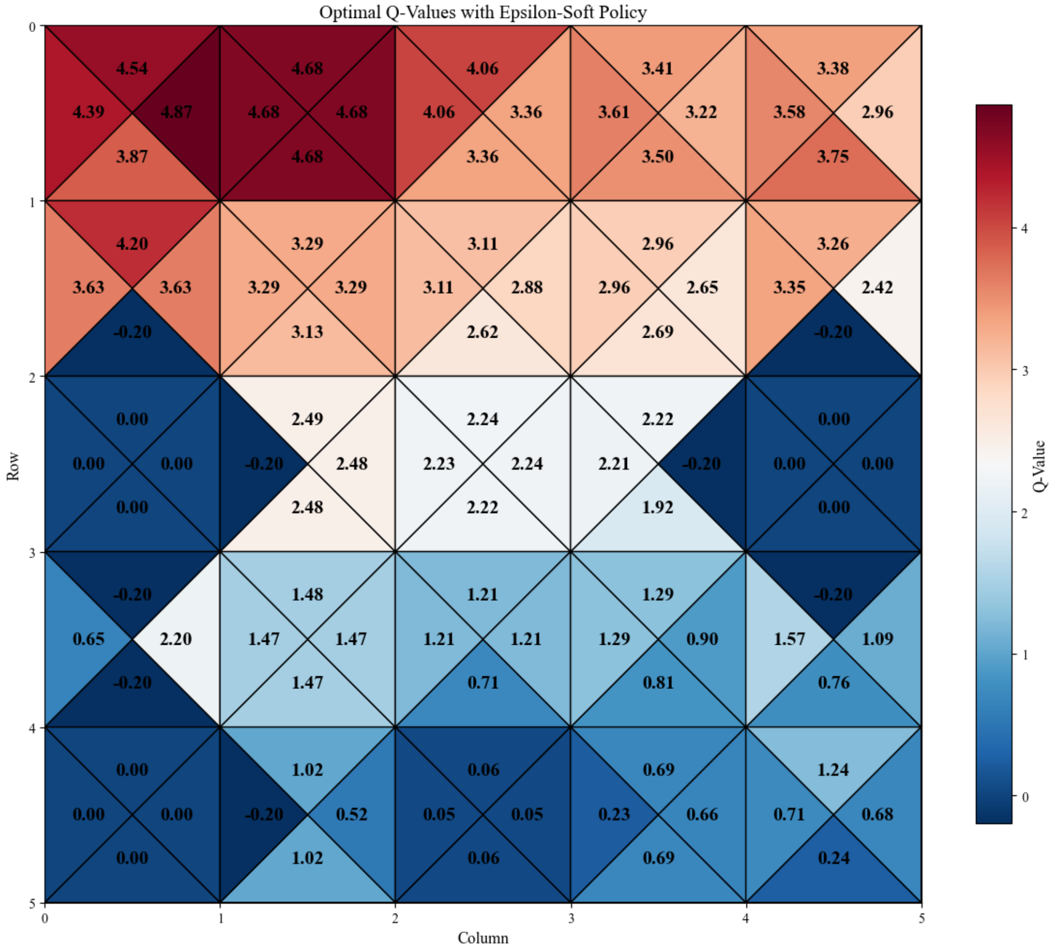

__Fig. 8.__ Optimal Q-Values with Epsilon-Soft Policy

Using the Monte Carlo method without exploring starts but the ϵ-soft approach, the optimal policy is expressed as

→	↓	←	←	↓
↑	→	←	↑	←
T	↑	↑	↑	T
→	↑	↑	↑	←
T	↑	↓	↑	↑

The optimal policy derived using the Monte Carlo method with an ε-soft approach reflects a well-balanced approach to navigating the gridworld, combining reward maximization with risk mitigation. This method allows the agent to follow actions with high estimated returns (exploitation) while still occasionally exploring less optimal choices, preventing it from getting stuck in suboptimal local policies.

The policy effectively guides the agent toward high-reward states such as the blue square at (0, 1) and the green square at (0, 4), which yield immediate rewards and favorable transitions to (4, 2) and (4, 4). The direction choices near these states confirm that the agent correctly leverages these transitions, e.g., actions from (0, 1) and (0, 4) lead to high-value outcomes. Furthermore, the policy features many upward movements, especially in rows 2 to 4, indicating a strong intent to return to top-row reward states or approach terminal states efficiently to limit cumulative penalties from intermediate moves. The policy also demonstrates logical structure and consistency. For example, rows 2 and 4 prioritize upward movement, efficiently aligning the agent with paths to high-value targets. Importantly, the ε-soft policy's stochastic nature still allows occasional deviation from greedy behavior, fostering a broader exploration of the environment, which is critical in Monte Carlo control without exploring starts. However, a few action choices could be refined for further optimization. For instance, the downward move from (4, 2) might be less efficient than moving upward toward (0,1) for quicker access to rewards. The reason for that is the equal Q value of upward and downward moves shown in Fig. 7 at point (4, 2).

Overall, this policy showcases a thoughtful balance between exploration and exploitation, made possible by the ε-soft approach. It ensures comprehensive learning of the environment while steadily guiding the agent toward high-return behaviors. The results highlight the ε-soft Monte Carlo method's effectiveness in deriving near-optimal policies in complex environments, especially when exploring starts are not feasible.

2. Use a behaviour policy with equiprobable moves

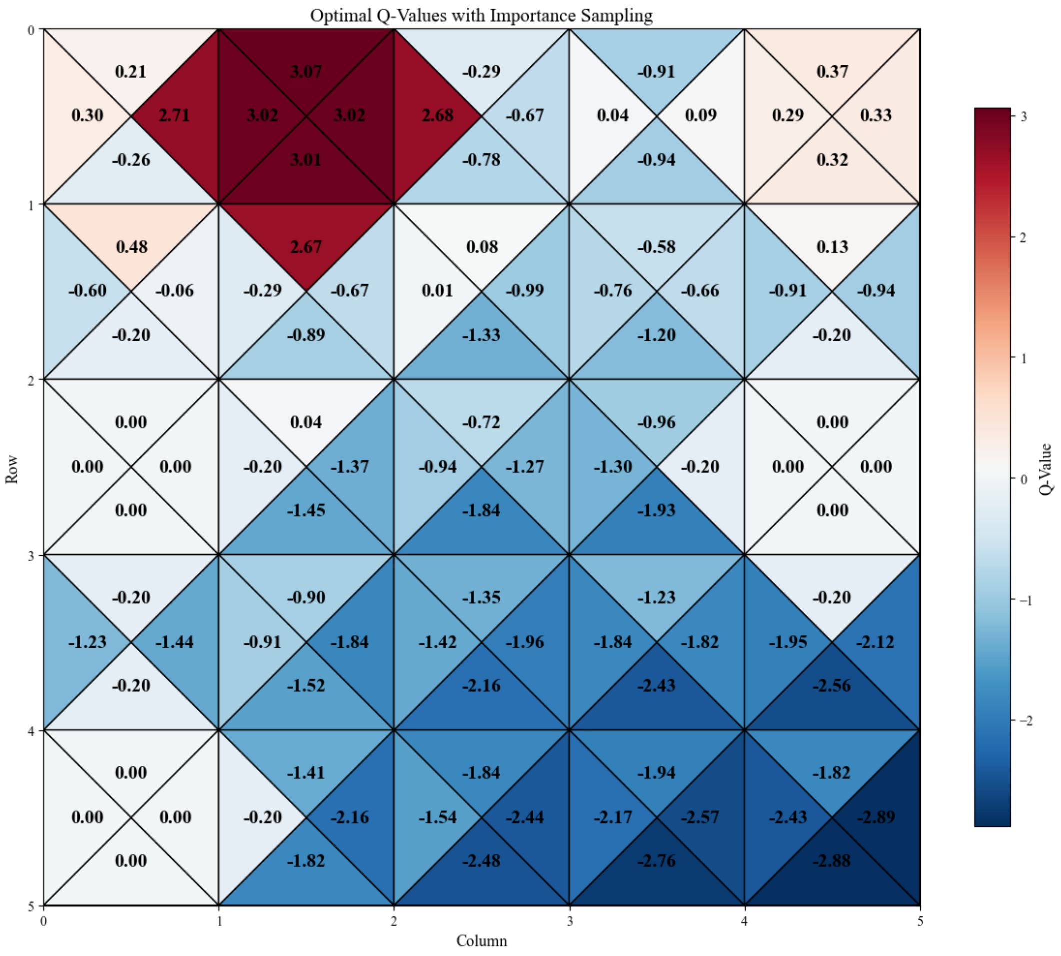

Fig. 8. *Optimal Q-Values with Epsilon-Soft Policy*

Using a behaviour policy with equiprobable moves, the optimal policy is expressed as

→	↑	←	→	↑
↑	↑	↑	↑	↑
T	↑	↑	→	T
↑	↑	↑	↑	↑
T	←	←	↑	↑

The optimal policy derived through importance sampling using an equiprobable behavior policy exhibits a consistent and reward-focused strategy across the gridworld. This approach adjusts for the mismatch between the behavior policy (used to generate episodes) and the target policy (being learned) by weighting the observed returns using importance weights. Given the uniform behavior policy ($\pi_b(a_t|s_t) = 1 / |A(s_t)|$), the importance weight simplifies to $W_t = \pi_t(a|s) / \pi_b(a_t|s_t)$. This effectively scales each return G_t based on how likely the target policy would have chosen that action.

The Q-values are updated with the weighted return:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot W_t \cdot (G_t - Q(s_t, a_t))$$

This mechanism ensures that updates are more heavily influenced by actions favored under the target policy, allowing the agent to learn the optimal behavior even when episodes are generated from a different, exploratory behavior policy. The resulting policy shows clear strategic alignment with high-reward squares, particularly the blue square at (0, 1) and the green square at (0, 4). For example, the policy begins with a rightward move from (0, 0) to reach (0, 1), which provides a reward of 5 and transitions to (4, 2). From (0, 1), an upward move reflects an intent to quickly capitalize on high-value states. Actions such as left from (0, 2) and right from (0, 3) indicate navigation toward known rewarding states.

In the middle rows, the dominant upward direction further supports this goal, facilitating transitions to higher-value top-row states. The third row, for instance, includes upward movements from majority of positions, maintaining alignment with value-driven navigation. However, a few action choices as right at (2, 3), left at (4, 1) could be refined for further optimization.

Overall, the policy reflects a robust learning outcome, with importance sampling effectively enabling the agent to estimate the optimal strategy despite following a random behavior policy. The method's ability to adjust for policy discrepancies ensures accurate learning, with the resulting policy closely aligned with the environment's reward dynamics. This confirms the effectiveness of importance sampling in off-policy reinforcement learning for deriving optimal policies in structured environments like gridworld.

References

[1] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction

[2] https://github.com/BY571/Medium_Code_Examples. Accessed on 15th of July 2024.