Cairo University

Faculty of Engineering

Computer Engineering Department

CMPS458 Reinforcement Learning Report
Assignment 1 Policy Iteration

Team Name/Number:
Mohamed Ahmed Ibrahim Sobh 1210288

Omar Ahmed Ibrahim 1210020

Supervisor: Ayman AboElhassan

October 16, 2025

Deliverables

Repo link: Github Repository

Video record link: add link here

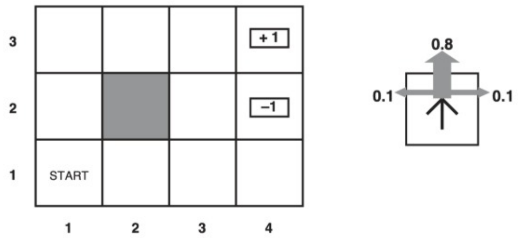

Figure 1: Example grid-world environment used for testing.

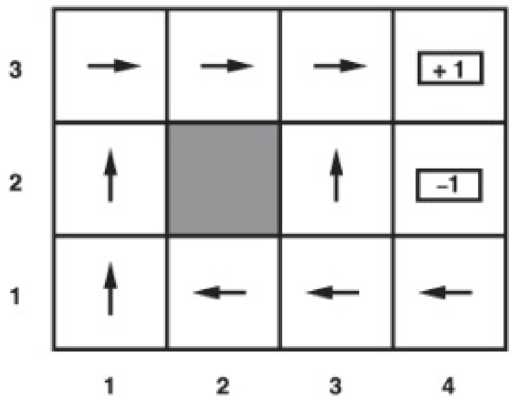

Figure 2: Example of a learned policy after convergence.

Table 1: Core parameters used in the policy iteration experiments.

Parameter	Symbol	Default Value	Description
Discount factor	γ	0.9	Future reward weighting
Convergence threshold	θ	1×10^{-6}	Stopping criterion for evaluation
Maximum iterations	N_{iter}^{max}	1000	Upper limit on policy updates
Step reward	R_{step}	-1	Penalty for each move
Goal reward	R_{goal}	10	Reward for reaching the goal state
Bad cell penalty	R_{bad}	-10	Penalty for entering a bad cell

Discussion

0.1 Experiments

Table 2: Effect of varying the discount factor γ on policy iteration results (seed=40501, maze=5, bad=2, correct hit=0.7 BUT the simulation steps are based on policy only).

γ	**Observation**	**Outcome / Behavior**
0.3	Agent focuses on immediate rewards	5 Policy Iterations AND 7 Simulation Steps
0.9	Balances short and long-term rewards	4 Policy Iterations AND 5 Simulation Steps

Table 3: Effect of convergence threshold θ on stability and convergence (maze=100, bad=40).

θ	**Observation**	**Outcome / Behavior**
10^{-1}	Fast convergence but less accurate	116 Iters to reach Stable Policy, Time taken = 15.75s
10^{-6}	Slower convergence with high precision	83 Iters to reach Stable Policy, Time taken = 19.57s

Table 4: Effect of changing reward values on agent behavior and learning outcome (maze=40, bads=20).

Reward Configuration	**Observation**	**Expected Outcome**
Goal=5, Bad=-1, Step=-0.1	Small penalty difference causes uncertain exploration	25 Policy Steps AND 55 Simulation Steps
Goal=10, Bad=-10, Step=-1	Balanced reward setup encourages stable learning	15 Policy Steps AND 55 Simulation Steps
Goal=20, Bad=-50, Step=-1	Strong contrast between rewards and penalties	20 Policy Steps AND 57 Simulation Steps

0.2 Question Answers

1. **What is the state-space size of the 5x5 Grid Maze problem?**
The state-space is defined by all possible agent positions on the grid. For a 5×5 maze, there are 25 possible states. The agent's state space is simply $|S| = 25$. With 4 possible actions (up, down, left, right), the total state-action space is $25 \times 4 = 100$.

2. **How to optimize the policy iteration for the Grid Maze problem?**
Policy iteration can be optimized by Using a higher convergence threshold θ to stop evaluation earlier, trading precision for speed.

3. **How many iterations did it take to converge on a stable policy for 5x5 maze?**
Using seed 40501, 10, -1, -10 reward system, 0.9 discount factor, and convergence factor 1e-6. Convergence was achieved in **4 iterations** of policy evaluation and improvement cycles.

4. **Explain, with an example, how policy iteration behaves with multiple goal cells.**
When multiple goals exist, each goal state acts as a separate terminal attractor. The value function forms *multiple basins of attraction*, where the agent's optimal policy directs it toward the nearest high-value region. For example, in a 5×5 grid with two goals at opposite corners, cells near each corner will point toward their closest goal, producing two distinct optimal sub-policies.

5. **Can policy iteration work on a 10x10 maze? Explain why.**
Yes, policy iteration can work on a 10×10 maze, but the computational cost grows quadratically with the number of states (100 states $\rightarrow$ 10,000 state-action pairs). It remains feasible, but slower.

6. **Can policy iteration work on a continuous-space maze? Explain why.**
No, standard policy iteration assumes a discrete and finite state space. In a continuous environment, the number of possible states is infinite, making it impossible to represent the value function as a lookup table.

7. **Can policy iteration work with moving bad cells (like Pac-Man moving ghosts)? Explain why.**
Not directly. Policy iteration assumes a stationary environment (fixed transition probabilities). If bad cells move, the environment becomes *non-stationary*.