Reinforcement Learning: Assignment 2

Alexander Y. Shestopaloff

July, 2025

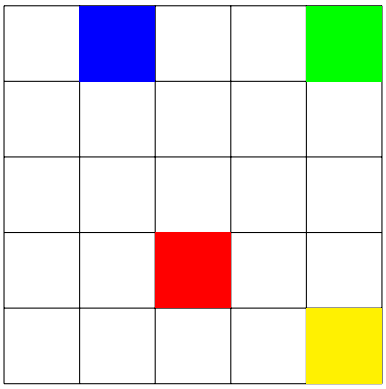
In this assignment, your goal will be to experiment with learning policies on simple reinforcement policies using different methods that we have covered in class.

You should provide a report including plots describing your results and a link to an online repository with commented and reproducible code. You should also provide a readme file that can make it easy for anyone to replicate the results in this assignment. Note that modern machine learning conferences e.g., NeurIPS, require submission of documented code for reproducibility of reported results.

Each part is worth 25 marks. You will be graded on correctness, clarity and reproducibility of your results. The assignment is due on July 21st. It may be done individually or in pairs. Both participants in a pair will receive the same grade and no preference will be given to people working individually or in pairs.

1 Part 1

Consider a simple 5×5 gridworld problem, described below. This is the simplest abstraction of a reinforcement learning problem that allows us to benchmark and compare various learning algorithms to one another and is known as the 'gridworld' environment.



Each of the 25 cells of the gridworld represent a possible state of the world. An agent in the gridworld environment can take a step up, down, left or right. If the agent attempts to step off the grid, the location of the agent remains unchanged.

The blue, green, red and yellow squares represent special states at which the behaviour of the system is as follows. At the blue square, **any** action yields a reward of 5 and causes the agent to jump to the red square. At the green square, **any** action yields a reward of 2.5 and causes the agent to jump to either the yellow square or the red square with probability 0.5.

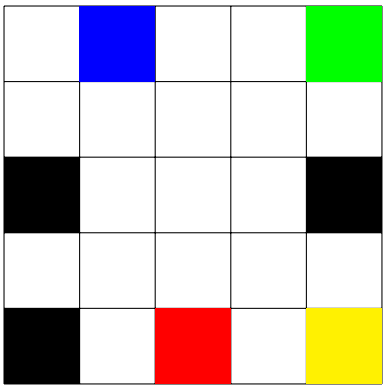
An attempt to step off the grid from a white / yellow square yields a reward of -0.5 and otherwise any move from a white / red / yellow square to any square yields a reward of 0. Intuitively, an agent with a good policy should try to find the states with a high value, and exploit the rewards available at those states.

1. Consider a reward discount of $\gamma = 0.95$ and a policy which simply moves to one of the four directions with equal probability of 0.25. Estimate the value function for each of the states using (1) solving the system of Bellman equations explicitly (2) iterative policy evaluation. Which states have the highest value? Does this surprise you?

2. Determine the optimal policy for the gridworld problem by (1) explicitly solving the Bellman optimality equation (2) using policy iteration with iterative policy evaluation (3) policy improvement with value iteration.

2 Part 2

Now let's change the environment a bit by adding some terminal states represented as the black squares. This gives rise to episodes where termination occurs once the agent hits one of the black squares. We will also assume, unlike in Part 1, that any move from a white / yellow / red square to any square yields a reward of -0.2 and an attempt to step off the grid from a white / red / yellow square yields a reward of -0.5, like in Part 1.



1. Use the Monte Carlo method with (1) exploring starts and (2) without exploring starts but the ϵ-soft approach to learn an optimal policy for this modified gridworld problem. Use the same discount factor of $\gamma = 0.95$ as you have in the Part 1 above. You can start with a policy with equiprobable moves.

2. Now use a behaviour policy with equiprobable moves to learn an optimal policy. Note here the dynamics of the world are known exactly, so you can actually compute the importance weights needed for this.