LEARNING TO MOVE IN THE GRIDWORLD WITH AND WITHOUT TRAPS

REINFORCEMENT LEARNING 2023/2024

INSTRUCTOR: F. DE PELLEGRINI

In this TP we consider a version of the gridworld MDP and we perform reinforcement learning in a setting 1) without traps, 2) in a setting with traps with an additional task for the valley gridworld variant.

1. System description.

The gridworld is a standard MDP used for Reinforcement Learning. The grid is a set of coordinate points $(x, y) \in \mathcal{S} = \{1, \dots, K\}^2$, where K is the gridworld side length K. There exist a start position $S = (1, 1)$ and a goal position $T = (K, K)$ (however, your code should work for any start and goal positions. The state s of an agent is its position (x, y), and the agent's action set at state $s \in \mathcal{S}$ is $A(s) \subset \{N, E, S, W\}$, where letters mean moving North, East, South and West on the grid with respect to the current position. Each action has a reward: the reward to move from state s to any state $s' \neq T$ is -1, wheras a transition to $s = T$ has reward $2(K - 1)$. T is a *terminal state*: the agent remains there forever.

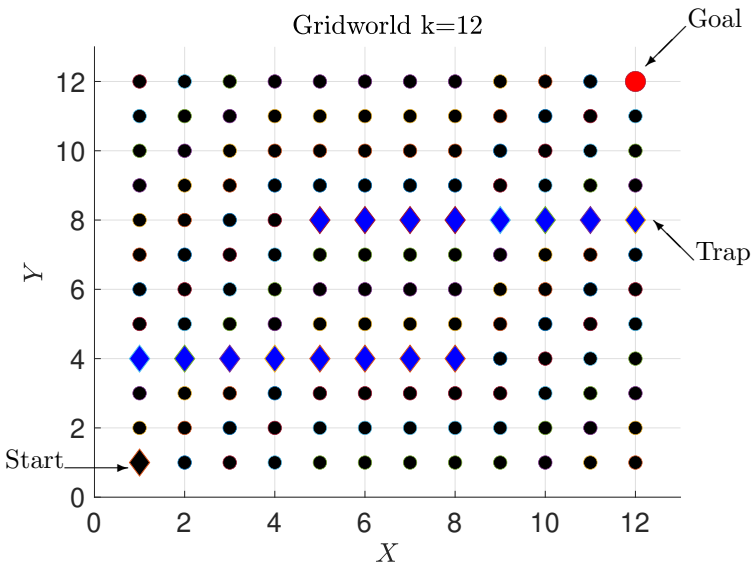

Figure 1. The gridworld for $K = 12$ with traps.

2. No Trap Case: preliminar setup

Task 1: Exploring the MDP. Write a program able to determine an optimal policy for the underlying MDP using value iteration *and* policy iteration. Plot the value function at each state (values on a grid) and draw the optimal policy starting at S.
Task 2: SARSA. Implement the SARSA algorithm for gridworld. Draw the optimal policy starting at S.
Task 3: Q-learning. Implement the Q-learning algorithm for gridworld. Draw the optimal policy starting at S.

3. Case with traps

Now fix $K = 12$ and consider the set of traps, i.e., terminal states placed at locations $U = \{(x, y)|1 \leq x \leq 8, y = 4\}$ and $\{(x, y)|5 \leq x \leq 12, y = 8\}$. However, moving from state s to $s' \in U$ has a reward $-2(K - 1)$.
Task 4: Learning with Traps. Repeat the previous tasks for the gridworld with traps. What is the difference you can notice among the algorithms? Can you explain the difference?

4. Montecarlo

Task 5: Monte Carlo (MC). Implement the Monte-Carlo policy iteration algorithm for gridworld 1) without traps and 2) with traps using *first visit* MC policy improvement. Draw the optimal policy starting at S. What is the difference you can notice among the case with traps and without? Can you explain the difference of behavior of the algorithm?