AI 533 **Intelligent Agents and Decision Making** Winter 2025

OREGON STATE UNIVERSITY

School of Electrical Engineering and Computer Science

Total points: 100 Mini Project 2 Due date: March 10, 2025

Instructions: This project consists of two parts. Collaboration is not allowed on any part of this project. You are welcome to brainstorm coding practices (including data structures to use) or clarify your understanding of the question with your peers but you cannot code together or copy/re-use solutions. Document who you worked with for this project and cite websites from which you utilized any code (e.g., Stack Overflow, Stack Exchange) in your code implementation. **You are not allowed to use AI tools to write the code for you**. You must submit a *project report (.pdf file) and the code*.

Problem Setup We will build on the gridworld used in Mini-project 1. Consider the gridworld in Figure 1, with two states covered in water and two states with wildfire. Each state is denoted as $\langle x, y, \text{water}, \text{fire}\rangle$. "Water" and "fire" are binary values, with 0 denoting the absence of water/fire and 1 denoting the presence of water/fire at location (x, y). The start state is denoted as $\langle 0, 0, 0, 0\rangle$ and the goal state is denoted by $\langle 3, 3, 0, 0\rangle$. The agent can move in all four directions. The agent succeeds with probability 0.8 and may slide to the neighboring cells with probability 0.1. Illustration of the transition probability for actions 'up' and 'right' are shown in Figure 1. If a move is invalid (such as moving into a wall), the agent will remain in that state with the corresponding probability. For example, when trying to move up in top-left cell (start state) of the grid, the agent will remain in that cell with probability 0.9 or move right with probability 0.1.

The agent receives a reward of $+100$ when it reaches the goal state. The agent receives a reward of -5 in the water states, -10 in widlfire states, and a reward of -1 in all other states. The process **terminates** when the agent reaches the goal state. The agent's objective is to maximize the expected reward it can obtain.

Part A: TD Methods (60 points)

1. **(20 points)** Implement tabular SARSA for this problem with $\gamma = 0.95$, using ϵ-greedy action selection. You may optimize the hyperparameters (α, ϵ) either manually or using automated search (e.g. grid search or random search in the space of hyperparameters). Report the hyperparameters you use. Include learning curve over 100 episodes in the .pdf file. Average results over 100 trials, and include standard deviation error bars. You may use an initial policy of your choice but clearly state the initial policy. in the document.

2. **(20 points)** Implement tabular Q-learning for this problem with $\gamma = 0.95$, using ϵ-greedy action selection. You may optimize the hyperparameters (α, ϵ) either manually or using automated search (e.g. grid search or random search in the space of hyperparameters). Report the hyperparameters you use. Include learning curve over 100 episodes in the .pdf file. Average results over 100 trials, and include standard deviation error bars. You may use an initial policy of your choice but clearly state the initial policy. in the document.

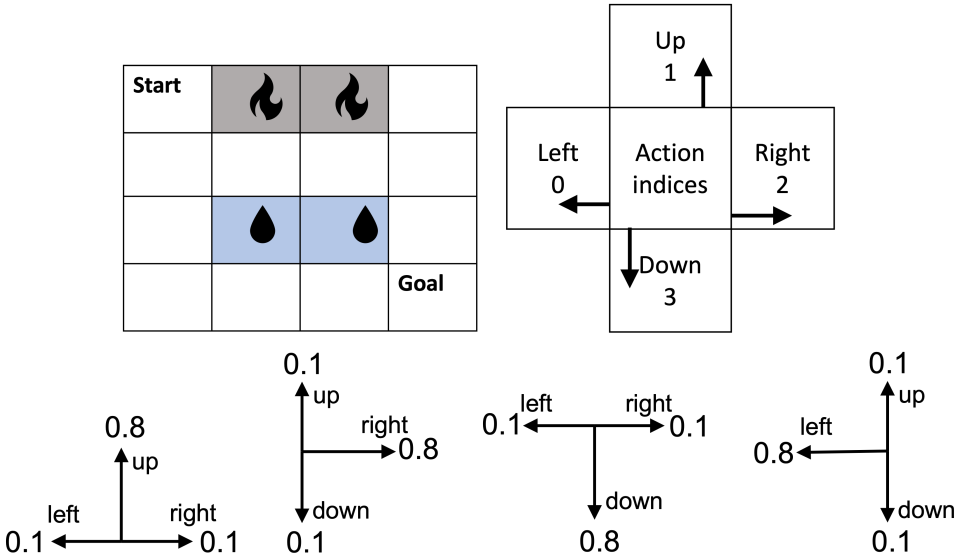

Figure 1: MDP

3. **(20 points)** Implement tabular SARSA(λ) for this problem with $\gamma = 0.95$, using ϵ-greedy action selection. Use the backward algorithm (with eligibility traces). You may optimize the hyperparameters (α, ϵ) either manually or using automated search (e.g. grid search or random search in the space of hyperparameters). Report the hyperparameters you use. Include learning curve over 100 episodes in the .pdf file. Average results over 100 trials, and include standard deviation error bars. You may use an initial policy of your choice but clearly state the initial policy. in the document.

Part B: Actor-Critic (40 points)

1. **(25 points)** Implement actor-critic algorithm for this problem with $\gamma = 0.95$ and a *function approximator of your choice*. Clearly state the initial policy in the submission file. Report the best hyperparameter values for your setting and how you identified them. Include plots of the learning curve across 100 episodes, averaged over 100 trials, along with the standard deviation in the .pdf file.

2. **(15 points)** Compare the accumulated reward at the end of 100 episodes with that of SARSA, Q-learning, and SARSA(λ). Discuss which algorithm performs better in this problem and why.