Mid-Term Assignment: Implementing Q-Learning

Jiadong YU

Fall, 2025

E-mail: `jiadongyu@hkust-gz.edu.cn`
Office Hours: Tue 12:30-1:30pm
Office: W1-614

Class Hours: Tue 1:30am-4:20pm
Class Room: E1-149

Due Date: 3rd Nov 2025

Late assignments will be accepted for no penalty if a valid excuse is communicated to the instructor before the deadline. Assignments submitted up to 24 hours late will incur a **10%** penalty. Submissions between 24-48 hours late will incur a **20%** penalty. Assignments more than 48 hours late will not be accepted without prior approval.

Objective

The objective of this assignment is to implement a Q-learning algorithm based on Fig. 1 to understand how it learns optimal action values in a given environment. You will visualize the convergence of the Q-table values over episodes and analyze the effect of changing initial and target states.

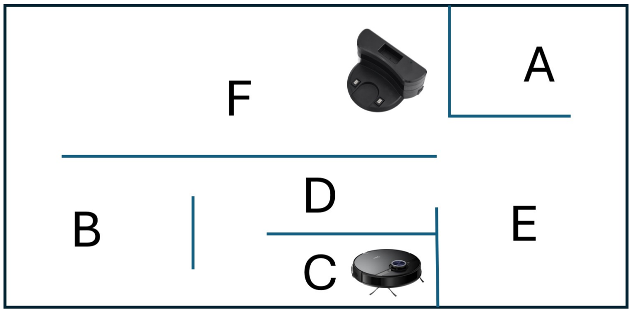

Figure 1: Grid world example.

Tasks:

1. Implement Q-Learning Algorithm:

- Write a program in Python (or your preferred programming language) to implement the Q-learning algorithm. You can use libraries such as NumPy for numerical operations.
- Ensure your implementation can handle different states and actions.

2. Environment Setup:

- Define an environment.
- Represent the states and actions appropriately.

3. Training:

- Choose a suitable number of episodes.
- Run the Q-learning algorithm and observe the convergence of the Q-table values.

4. Convergence Plot:

- Generate and report a plot showing the convergence of Q-table values.
- Discuss the convergence behaviour observed in the plot.

5. Change Initial and Target States:

- Change the initial state to A and the target state to C in your environment setup.
- Rerun the Q-learning algorithm and report the new Q-table values after convergence.
- Generate a new convergence plot for this configuration, similar to the one in Task 4.

6. Report:

Compile your findings and observations in a report. Your report should include:

- Introduction to Q-learning and its significance.
- Description of the environment and states used.
- Discussion on the initial state and target state changes.
- Q-table values before and after convergence for both configurations.
- Convergence plots for both configurations.
- Analysis of results and any observations or conclusions drawn from the experiments.

Submission Guidelines:

- Submit your code as a .zip file, including all necessary scripts and files to run your Q-learning implementation.
- Submit a PDF report that includes all the required sections listed above.
- Ensure that your code is well-commented and organized for clarity.

Grading Criteria:

- Implementation: Correctness and efficiency of the Q-learning implementation.
- Analysis: Depth of analysis in the report and quality of convergence plots.
- Presentation: Clarity, structure, and professionalism of the report.
- Creativity: Additional insights or improvements made in the implementation or analysis.