Milestone 1: Project Plan

Course: CSE 150A/250A: Probabilistic Reasoning and Learning

Date: November 14, 2025

Group Members: Tzu Ping Chen (A69041432), Kai Cheng Liu (A69042222) , Cheng-Yang Wu (A69042344), Chih Yun Lin (A69045651)

1. Problem Description

This project investigates the derivation of an optimal policy for a sequential decision-making problem under uncertainty. We will implement and analyze Reinforcement Learning (RL) agents within the Taxi-v3 environment, a classic problem from the Gymnasium library. The environment consists of a discrete 5x5 grid world where an agent must learn an optimal policy to navigate, pick up a passenger, and transport them to a designated destination.

This project's objective is to model this problem as a Markov Decision Process (MDP). Our primary focus will be a "comparison of alternative modeling approaches"[2], specifically contrasting model-based and model-free RL algorithms.

2. Dataset Source

This project will utilize a simulated environment rather than a static dataset. All interactions and experiential data will be generated by the Taxi-v3 environment from the Gymnasium Python package.

- Source: Gymnasium (a fork of OpenAI Gym).
- Environment Model: The environment is fully defined by:
    - A discrete state space (S) of 500 states.
    - A discrete action space (A) of 6 actions (North, South, East, West, Pickup, Dropoff).
    - A reward function (R) defined for state-action transitions.
- Data Processing: No preprocessing is required, as the environment provides a discrete, fully-observable state-action space suitable for direct use by our algorithms.

3. Methodology

Our methodology centers on a comparative analysis of the two primary classes of RL algorithms specified in the course overview[3]:

1. Model-Based Algorithm: Value Iteration
    - Approach: We will formally model the environment as an MDP and implement Value Iteration. This algorithm leverages the environment's known transition model P(s'|s,a) and reward function (R) to "plan" and compute the optimal state-value function, which is the *best possible* total future reward the model can get if it start in state s.
2. Model-Free Algorithm: Q-Learning
    - Approach: Conversely, we will implement Q-Learning, a model-free algorithm. This agent will operate without *a priori* knowledge of the environment's transition or reward models. It will learn the optimal action-value functio, which is the *best possible* total future reward the model can get if it start in state s, take action a *just this once*, and then play perfectly forever after.

Analysis and Comparison:

Our "Results and Discussion" will quantitatively compare these two methods. We will present convergence plots for Q-Learning (e.g., cumulative reward per episode) and contrast its derived policy with the optimal policy from Value Iteration. We will also conduct a sensitivity analysis on the Q-Learning agent's performance with respect to key hyperparameters.