Project Proposal: Reinforcement Learning for Grid-based Navigation

Arnav Kumar, Prathamesh Koranne
{kumararn,koranne}@usc.edu

March 20, 2024

1 Problem Description

The task at hand is to train an autonomous agent capable of solving a grid based navigation problem to reach a goal state. We aim to solve 2 medium maps as described in the GitHub repository.

2 Proposed Solution

We propose a reinforcement learning approach utilizing Temporal-Difference (TD) learning, specifically the TD(λ) algorithm, enhanced with strategically initialized Q-values. The Q-value for a state s and action a is denoted as $Q(s, a)$ and is updated via:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right], \tag{1}$$

where:

- r is the immediate reward received after taking action a in state s,
- γ is the discount factor,
- s' is the resultant state following action a,
- a' is a possible action in state s',
- α is the learning rate.

The Q-values are initialized based on the following heuristics:

- **Distance-based Heuristic**: Q-values are initialized proportionally to the Manhattan distance from the goal state G. For a state s at coordinates (x, y), the heuristic value $H_d(s)$ is given by:

  $$H_d(s) = \frac{M - d(s,G)}{M}, \tag{2}$$

  where M is the maximum possible Manhattan distance in the grid, and $d(s, G)$ is the Manhattan distance between s and G.

- **Strategic Positioning Heuristic**: States that have obstacles on opposite cardinal directions are considered strategic. The heuristic value $H_s(s)$ for a strategic state s is initialized to:

  $$H_s(s) = H_d(s) \cdot C, \tag{3}$$

  where C is a constant that determines the relative importance of strategic positioning.

- **Collision Penalty**: For actions leading to a collision, a negative reward $R_{\text{collision}}$ is used:

  $$Q(s, a_{\text{collision}}) \leftarrow R_{\text{collision}}. \tag{4}$$

The TD(λ) algorithm introduces the concept of eligibility traces, denoted as $e(s, a)$ for a state-action pair (s, a), which are temporal records of occurrence for each state-action pair. They are updated as follows:

$$e(s,a) \leftarrow \gamma\lambda e(s,a) + \mathbf{1}(s_t = s, a_t = a), \tag{5}$$

where:

- γ is the discount factor,
- λ is the trace decay parameter, with a value between 0 and 1,
- $\mathbf{1}(s_t = s, a_t = a)$ is an indicator function that is 1 if $s_t = s$ and $a_t = a$, and 0 otherwise.

The Q-value update rule in TD(λ) is modified to include the eligibility trace, enhancing the effect of previous state-actions:

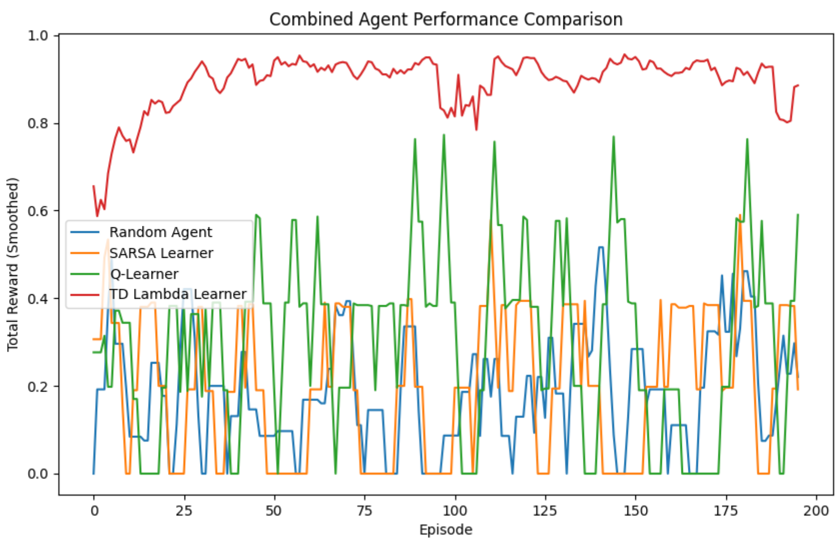
$$Q(s, a) \leftarrow Q(s, a) + \alpha \delta_t e(s, a), \tag{6}$$

where δ_t is the temporal-difference error at time t:

$$\delta_t = r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t). \tag{7}$$

3 Feasibility Study

Our preliminary results have shown that by initializing Q-values based on the proximity to the goal and penalizing collision paths, the agent learns to navigate basic maps effectively. Given below is the plot of various agents on medium-1 map.



4 Plan

Our methodology will follow a structured timeline to implement and evaluate the proposed solution rigorously. The work will be split between the team members as follows:

Table 1: Timeline, activities, expected outcomes, and responsibility of the proposed project.

Date/Week	Activity	Milestone/Outcome	Responsibility
Week 1	Setup Implementation	Setup Code Base	Arnav Kumar
Week 1	TDLambda Implementation	Algorithm Code Base	Prathamesh Koranne
Week 2	Initial Evaluation	Positive Results on Easy Maps	Arnav Kumar
Week 3	Hyperparameter Sweep	Eval on Medium Maps	Prathamesh Koranne
Week 4+	Hierarchical Distance Heuristics	Comprehensive Evaluation	Arnav Kumar & Prathamesh Koranne