Project Proposal: Reinforcement Learning for Grid-based Navigation

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

March 28, 2024

1 Problem Description

We aim to develop an autonomous agent capable of navigating through complex grid-based maps to reach a designated goal state, with the challenge extended to solving a hard map as outlined in our project repository.

2 Proposed Solution

Our approach employs the TD(λ) learning algorithm, enhanced with novel Q-value initialization strategies to better navigate the complexities of the environment. Specifically, we refine our heuristics for initializing Q-values as follows:

- **Distance of First Encounter:** We adopt a BFS methodology to initialize Q-values based on the shortest path to the goal that avoids obstacles. This distance is measured as the steps required to first encounter the goal without hitting barriers, significantly refining our agent's strategic planning over mere proximity.
- **Strategic Positioning:** We maintain our emphasis on strategic positioning, recognizing states with obstacles on opposite cardinal directions as crucial waypoints. Such states are identified during the BFS exploration and are assigned higher Q-values due to their strategic importance in navigating mazes and complex maps.
- **Collision Penalty:** Actions leading directly to collisions continue to receive negative reinforcement, discouraging the agent from engaging with walls and obstacles, thereby enhancing its navigational efficiency.

Q-value updates are performed leveraging the eligibility trace mechanism, as previously outlined, to reinforce the agent's learning from both past and present decisions:

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

where δ_t encapsulates the temporal-difference error, refined by our enhanced heuristics.

3 Plots and graphs

Latest simulations demonstrate the agent's improved ability to navigate through complex environments effectively. The BFS-based heuristic, combined with strategic positioning and penalization for collisions, has shown promising results in enabling the agent to solve hard maps with increased efficiency. Below is the graph where we compare our "TDLambdaLearner Model" with the "RandomAgent Model", "SARSALearner Model" and "QLearner Model", as it it can be seen that our model has performed better than all other model for "Hard - 0" map

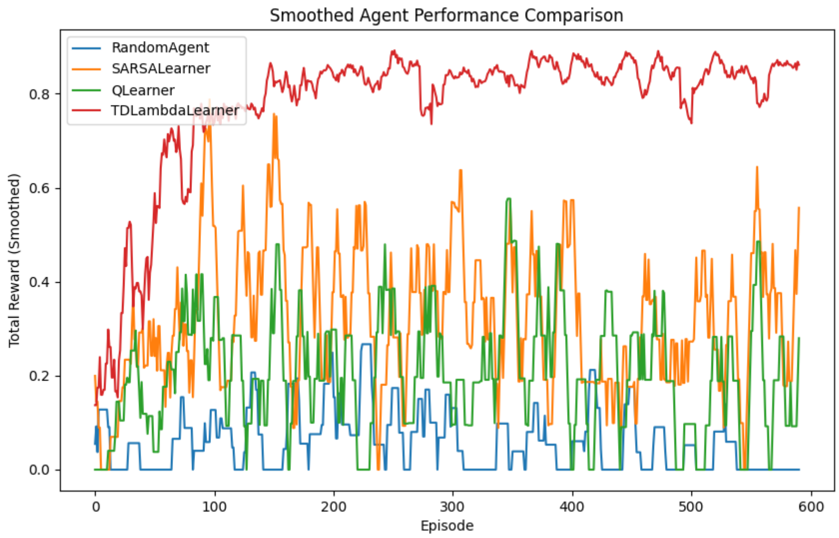

Figure 1: Comparative performance of the agent with updated heuristics on the "Hard 0" map, showcasing enhanced navigational capabilities.

4 Plan

We have outlined a structured timeline for the development, implementation, and rigorous evaluation of our proposed solution:

Table 1: Projected timeline and division of responsibilities.

Date/Week	**Activity**	**Outcome**	**Responsibility**
Week 1	Setup and Implementation	Codebase Setup	Arnav Kumar
Week 1	Algorithm Enhancement	Enhanced TD(λ) Algorithm	Prathamesh Koranne
Week 2	Initial Testing	Positive Results on Hard Maps	Arnav Kumar
Week 3	Hyperparameter Optimization	Optimization on Various Maps	Prathamesh Koranne
Week 4+	Final Evaluation	Comprehensive Performance Evaluation	Both