Exploring and Addressing Limitations of Straightforward RL Approaches on the Frozen Lake Problem

Carson Young [1]

Abstract

Using a variety of strategies, I will attempt to explore possible solution developments for the Frozen Lake problem as it approaches larger map sizes. Currently, generalized strategies like SARSA or Q-learning will begin failing at map sizes greater than 11×11, due to the agent's inability to reach the goal, leaving the Q-table empty for these strategies. This impossibility of reaching the goal for larger map sizes has been mitigated through two specific strategies: reworking the reward system, and allowing the agent to start from any valid tile on the map (for learning only). A step penalization measure has also been implemented in order to help mitigate issues with local maxima as the agent operates in this alternative reward system.

1. Domain

1.1. History

Frozen Lake from Open AI is a simple, grid environment. Solving the environment involves solving a path finding problem; the agent starts at one corner of the grid, and its goal is to reach the opposite corner while avoiding holes in the ice (Pena & Banuti, 2021).

1.2. Gymnasium and Custom Packaging

Gymnasium provides a platform that allows users to quickly and easily implement RL strategies on various environments. I use Gymnasium to benchmark the typical Frozen Lake problem, and have implemented my own Python package for the alternative reward system.

[1]Department of Computer Science, University of Oklahoma. Correspondence to: Carson Young <major.add.9@ou.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1.3. Environment

The observation and action spaces of the Frozen Lake problem are incredibly simple and intuitive:

The observation space is simply the tile that the agent is currently positioned at, an integer value from 0 to $(\text{nrows})^2 - 1$.

The action space can be consolidated to a four-value map as follows,

$0 \rightarrow$ move left,
$1 \rightarrow$ move down,
$2 \rightarrow$ move right,
$3 \rightarrow$ move up.

2. Hypotheses

Hypotheses are as follows:

1. Q-Learning, on the whole, due to its slightly different updating function, will outperform SARSA in this environment.
2. Altering the reward system and the agent's initial positions throughout should drastically improve said agent's ability to learn Frozen Lake for *any* sufficiently large map size.

3. Experiments

3.1. Frozen Lake Changes

As mentioned before, I have altered some of the underlying code that runs Frozen Lake, so it would be best to first explain these changes in detail.

3.1.1. Alternative Reward System

The way that Frozen Lake gives out rewards has been overhauled in such a way that, on larger map sizes, it incentivizes the agent to move towards the bottom right corner (goal space). This methodology itself has been balanced more to alleviate issues with the agent being stuck in local maxima,

Algorithm 1 Alternate Reward

Input: tile info x_t, current row r, current column c
if $x_t =$ goal **then**
  return 200
end if
if $x_t =$ hole **then**
  return 0
end if
Initialize $min = \min(r, c)$, $max = \max(r, c)$
if $min \neq 0$ **then**
  if $0 \equiv min \bmod 3$ **then**
    if $0 \equiv max \bmod 2$ **then**
      return $\frac{min}{map\ size}$
    end if
  end if
end if

though that problem is also more thoroughly addressed in Section 3.1.3.
Each valid map tile (not hole, goal or start) goes through a check at the initialization of the environment, which then associates a reward value with the tile, as per Gym's environment specifications. The tiles that can give out a reward, and what reward they give out, adhere to Algorithm 1.

Visually, this can be expressed by Figure 1,

where each concentric square path (largest to smallest) gives $\frac{3}{15} = \frac{1}{5}, \frac{2}{5}, \frac{3}{5}$, and $\frac{4}{5}$ of reward respectively.

3.1.2. Agent Respawn Point

This can quite simply be explained as changing the start point of the agent for each new episode, with the small caveat that any invalid tile (goal tile, hole tile) can *not* be chosen as the start point for the agent.

3.1.3. Step Penalty

This is a counteractive measure, as I noticed upon changing the reward system, that with certain policies, the agent would run back and forth between small reward tiles instead of further exploring and discovering either better reward tiles, or the goal tile. The reward system changing to every other tile helped mitigate this issue somewhat, however, the necessity for a more proactive measure was clear. A step penalty was implemented, altering the agent's reward to suit Algorithm 2. This allows the environment to proportionally remove the reward the longer the agent takes to receive it; rewards start at their full value (since $p_t = 1$), and at the end of the episode, their value will tend to 0.

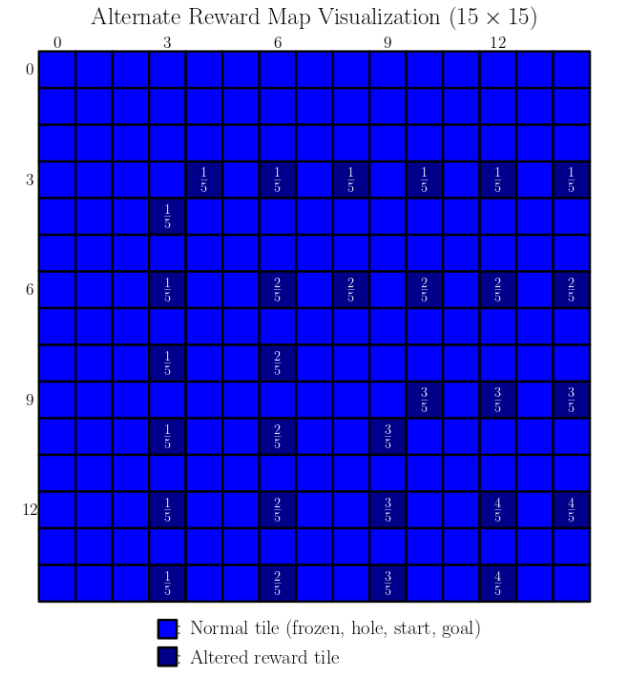

Figure 1. Map reward tile visualization

Algorithm 2 Step Penalty

Input: reward r_t, previous step penalty p_t, number of episodes n
$r_t = r_t * p_t$
$p_t = p_t - \frac{1}{n}$
return r_t

3.2. Hypothesis 1

Both the Q-learning and SARSA agent ran in identical environments (with reference to the above changes) while utilizing the following hyperparameters: $\alpha = 0.08$, $\gamma = 0.9$, and $\epsilon = 0.1$. Both agents adhered to an ϵ-greedy policy, and were chosen due to the fact that though they are similar in implementation, their outlooks and strategies (safe vs. greedy, on-policy vs. off-policy) are vastly different. Each agent was trained for 20,000 episodes, and each agent ran said episodes a total of 10 times for stochasticity, to be averaged and displayed as a confidence interval in the average rewards section of Figure 2 and Figure 3. As we can see in the plots, the average reward for the SARSA reward was actually higher than that of the Q-learned agent. This is the opposite of my theory, as I thought the off-policy, more greedy algorithm would win a game of "find the optimal action". It turns out, in fact, that the less optimistic strategy wins out, perhaps because it genuinely searches more, and therefore more consistently finds the goal post, actively

converging to the optimal strategy.

3.3. Hypothesis 2

Identical hyperparameters were used with reference to this hypothesis, however, this time, the difference being that the agent would comparatively run on the default Frozen Lake environment. The maps are randomly generated and saved beforehand so that all agents are learning on the same map, regardless of the exact Gym environment being used. As can be seen in Figures 4 to 7, the agents in the alternate environment have nearly filled their Q-tables. These results are promising, but (especially as the map sizes get larger) can see that the Q-values taper off the further from the goal we get. Though this is expected behaviour, this may imply that there is a critical maximum with respect to map size that this alternative system may be able to solve. I hesitate to include the figure maps for the default environment, as they are completely empty for map size $> 11 \times 11$, thus proving our hypothesis, that after 20,000 episodes, the alternative reward system for either agent is massively superior the larger the map size.

4. Literature Review

4.1. Safe Reinforcement Learning via Curriculum Induction

This approach is less included for comparison, and more included as exploration. This article details various methods for allowing the process of learning to be more safe – literally – for the agent, and uses Frozen Lake as a medium to explore the performance consequences of doing so (Turchetta et al., 2020). The idea of simplifying problems where safety is a critical issue while learning, like autonomous cars, into problems like Frozen Lake does make a certain amount of sense as a proof of concept, though it begs the question, how can this incredibly simplistic environment be abstracted to the ones the paper is concerned with, and their complexities?

4.2. Reinforcement Learning for Pathfinding With Restricted Observation Space in Variable Complexity Environments

The comparisons here are limited due to the largest map size within the article being 8×8, but some of the concepts in the article are both intriguing, and have some degree of comparability. The authors use variations of Q-learning and SARSA, as I have, but the training methods are built to teach general awareness about the agent's environment, so that the agent is more easily and quickly adaptable to changing environments. Their agent (on the 8×8 board, compared to my 9×9 board) converges to some successful solution within ≈ 500 and ≈ 900 timesteps (Pena & Banuti, 2021), whereas according to my step counter graphs, my solution will only converge usually after ≈ 1000 timesteps, implying their solution is more beneficial than mine, at least for the smallest board size in my representation.

5. Conclusion

Due to the changes in the reward system, we were able to greatly improve the agent's ability to solve sufficiently large map sizes. Oddly, SARSA gained more reward on average than Q-learning did, in direct opposition to my hypothesis, but with Q-learning reaching some optimally feasible solution in less time than SARSA.

6. Future Work

There is much to be improved in my agents. Finding solution developments that don't require exponentially increasing processing power and linearly increasing memory space for increasing map sizes would be a good start, but also, it is clear that my solutions are not as complete and efficient as I had initially thought. Perhaps learning on some different methods (TD(λ), Deep-Q, etc...) would also benefit in this sense.

7. Contributions

Contributions are all mine, and mine alone, except for two instances,

- The Frozen Lake environment from Gym was largely left unaltered. I only changed function calls and specific functions in their entirety before packaging it myself. The original code can be found here.
- The data visualization methods utilize much of the code found here, only with alterations on the specific visualizations.

References

Pena, B. D. and Banuti, D. T. Reinforcement learning for pathfinding with restricted observation space in variable complexity environments. *In AIAA scitech 2021 forum (p. 1755)*, 2021.

Turchetta, M., Kolobov, A., Shah, S., Krause, A., and Agarwal, A. Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems*, 2020.

8. Appendix

Figure 2. Rewards and step counts for the Q-Learning agent over 20,000 episodes, over differing map sizes.

Figure 3. Rewards and step counts for the SARSA agent over 20,000 episodes, over differing map sizes.

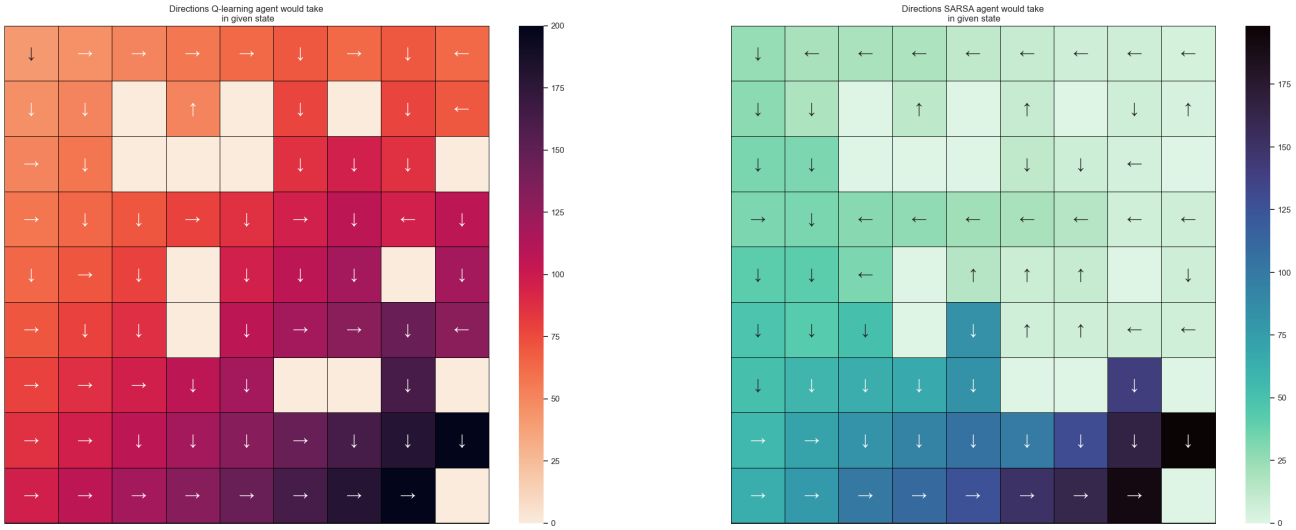

Figure 4. Maximum value in Q-table for Q-Learned and SARSA agent at end of 9×9 simulation, respectively

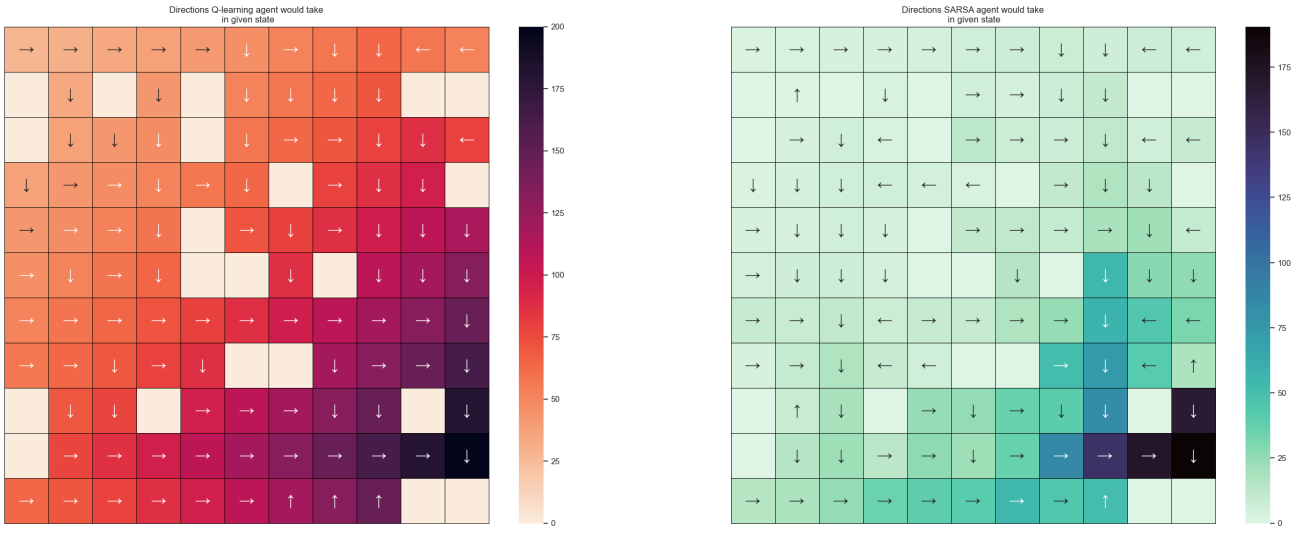

Figure 5. Maximum value in Q-table for Q-Learned and SARSA agent at end of 11×11 simulation, respectively

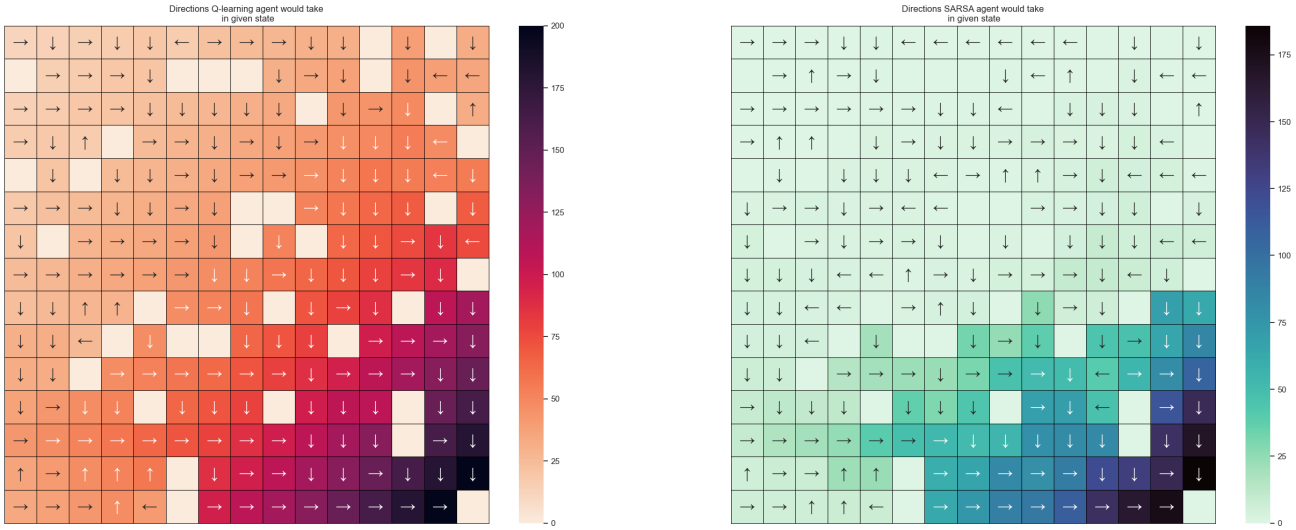

Figure 6. Maximum value in Q-table for Q-Learned and SARSA agent at end of 15×15 simulation, respectively

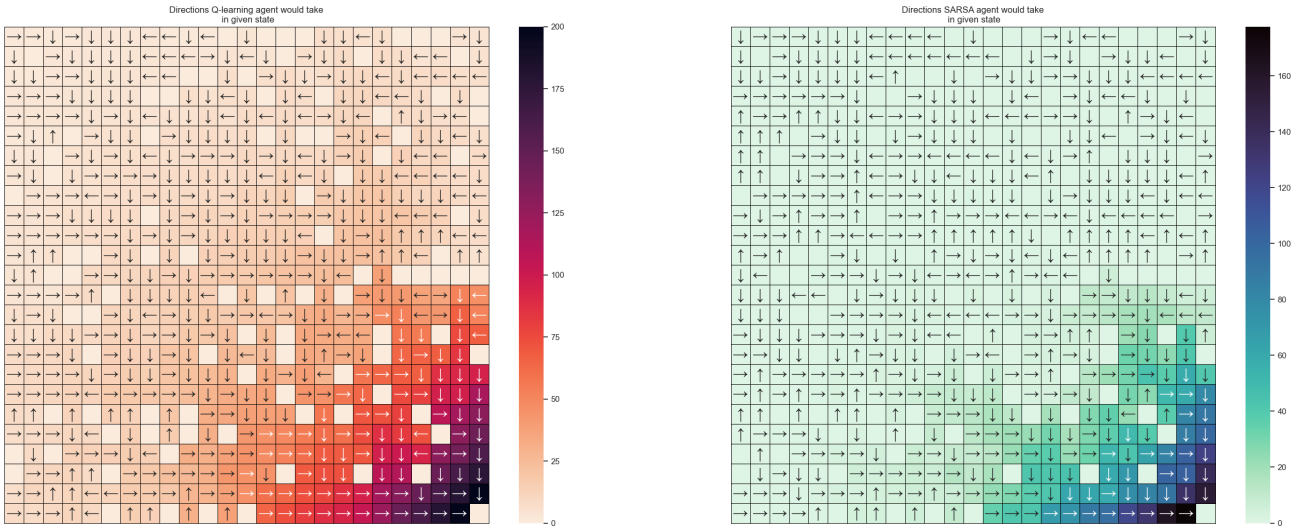

Figure 7. Maximum value in Q-table for Q-Learned and SARSA agent at end of 25×25 simulation, respectively