WPI

CS 534 Artificial Intelligence
Assignment 4

Group 10

April 11, 2018

Group Member

Yixuan	Jiao	yjiao@wpi.edu
Yinkai	Ma	yma7@wpi.edu
Jiaming	Nie	jnie@wpi.edu
Pinyi	Xiao	pxiao@wpi.edu

Content

1 Program Description

Our program is to solve a gridworld problem by creating an agent that uses reinforcement learning. In the gridworld, there are three special states: start, goal, pit (the goal and pit offer reward to the agent). For each movement, the agent will get a reward. The goal of agent in each trial is to gain most reward that it can. For each trial, we randomly pick a start state (any grid except goal and pits) and calculate its lookup table by using Q-function. And then use ϵ-greedy to help the agent explore. The agent has five actions to choose for each step: up, down, left, right, and give up. There is a ϵ value of probability that the agent will choose a random action, and a $1 - \epsilon$ value of probability that the agent will choose the action that has a maximum Q value. If the agent chose to give up, the trial will end and calculate total reward, otherwise we use actions that the agent chose with the corresponding Q values to update the old Q value of that initial step. However, because the uncertainty of the environment, there is 70% chance that the agent takes the expected movement, 10% chance it winds up at 90 degree right, 10% chance it winds up at 90 degree left, and 10% chance that it moves 2 squires forward. Then we repeat all of above steps from the grid that it winds up with until it gives up, gets in a pit, or reaches the goal. Finally, we calculate the total reward of the trial, and then restart for another trial. Through the whole process, the learned states of the gridworld are reserved for the following iterations, so that the agent can reinforce its learning and make better decision.

2 Approach

2.1 Set up the gridworld and random start(not a goal or pit)

- We store the state status in a dictionary by using i * 10 + j to represent the key of state coordination (i+1, j+1), for example, key 35 means it is row 4, column 6. We memorize the Q-values of each normal grid state by storing the action values of each state in a five numbers list to represent the rewards of the state for moving "Up, Down, Left, Right, Give up". For pits and goal, we use int type to store the reward value.
- Start with a randomly picked point (except goal and pits).

2.2 Calculate Q-value lookup table by using SARSA and Roll dice to decide the actions

- For each step, make action decision by generating a random P (range from 0 to 1), then compare P with ϵ. If $P < \epsilon$, randomly pick an action and calculate the corresponding Q-value , otherwise, take the action that has the maximum Q-value, which is calculated by using $Q(s, a) = Q(s, a) + (R(s) + \gamma Q(s, a)Q(s, a))$, where $\gamma = 1$.
- Use if statement to judge the state that next to the Wall in order to make sure the agent bounced back once it hits a wall.
- At anytime, if the agent decide to give up or get into a pit or goal, the trial end. If the trial doesnt end after the first action from the initial state, the agent will repeat step 2.(1) to see Q(s,a).

2.3 Calculate new value and update

- Update the Q(s,a) by using the Q(s,a)

2.4 Calculate the place it will wind up

- Generate a random P2 (range from 0 to 1). If $P2 > 0.3$, then the agent will take the movement as expected in step 2; if $P2 < 0.1$, it will go 90 degree left; if $0.1 <= P2 <= 0.2$, it will go 90 degree right; $0.2 < P2 <= 0.3$, it will move forward 2 steps.

2.5 Total Reward Calculation

- Repeat from the second step until the trial end(goal, pit, or give up), and then calculate the total reward of the trial.
- Total reward = number of steps * Movement Reward + State Reward/Give up Reward.

2.6 Restart

Restart another trial by using the environment that the agent learned from former trials.

3 Write Up Questions Solution

3.1 Program Result Asymptote Plot

In figure 1, the asymptote plot of scores vs. iterations is demonstrated. The number of iterations is 10000, take the median every 50 iterations and the parameters are on the following:

$\epsilon = 0.0$, $\alpha = 0.5$

3.1.1 Iterations = 10000 Asymptote Plot

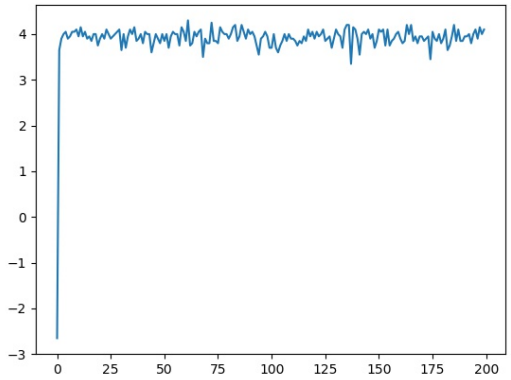

Figure 1: Scores vs. Iterations Model Training Result

3.1.2 Iterations = 1000 Asymptote Plot

The asymptote plot of scores vs. iterations is illustrated in figure 2, the parameters in this model are $\epsilon = 0.0$, $\alpha = 0.5$.

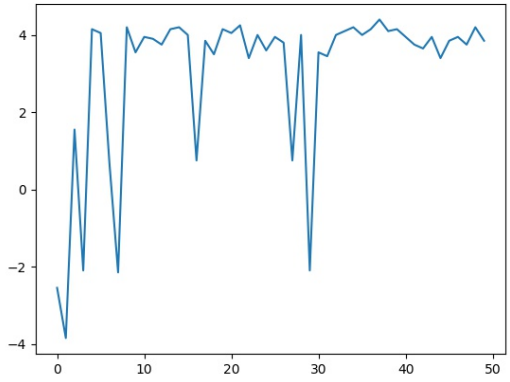

Figure 2: Scores vs. Iterations Model Training Result

3.1.3 Map Movements

The movements on the map is illustrated in figure 3, the symbols meanings are in the table 1.

Table 1: Movement Symbols

Symbols	Meaning
P	Pit State
G	Goal State
$\vee$	Move Up
$\wedge$	Move Down
$<$	Move Left
$>$	Move Right

The goal reward is 5, pit reward is -2, move reward is -0.1 and give up reward is -3.

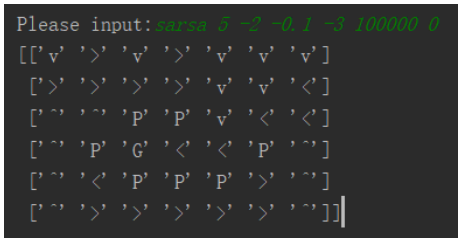

Figure 3: Movement Map

The score map for each grid is on the following figure 4.

```
[[ 4.1  4.2  4.3  4.4  4.5  4.4  4.3]
 [ 4.2  4.3  4.4  4.5  4.6  4.5  4.4]
 [ 4.1  4.2 -2.  -2.   4.7  4.6  4.5]
 [ 4.  -2.   5.   4.9  4.8 -2.   4.4]
 [ 3.9  3.6 -2.  -2.  -2.   4.2  4.3]
 [ 3.6  3.7  3.8  3.9  4.   4.1  4.2]]
```

Figure 4: Movement Map

3.2 α Optimization

In this part, different asymptotes plot will be compared to the $\alpha = 0.5$.

3.2.1 $\alpha = 0.5$

The asymptote plot of the $\alpha = 0.5$ is shown in figure 5.

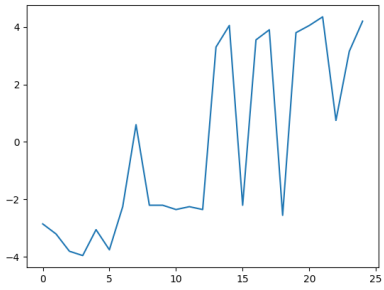

Figure 5: $\alpha = 0.05$ Iterations $= 1000$

3.2.2 $\alpha = 0.05$

The asymptote plot of the $\alpha = 0.05$ is shown in figure 6

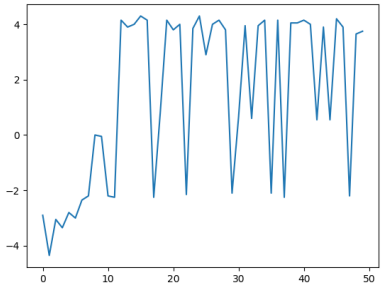

Figure 6: $\alpha = 0.05$ Iterations $= 1000$

3.2.3 $\alpha = 1$

The asymptote plot of the $\alpha = 1$ is shown in figure 7

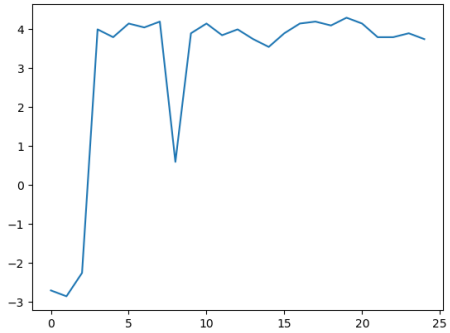

Figure 7: $\alpha = 1$ Iterations = 1000

3.3 $\epsilon = 0.2$ Asymptotes

3.3.1 Iterations = 1000

In the figure 8, the result asymptote plot under the situation $\epsilon = 0.2$ is illustrated. Compared to figure 2 with $\epsilon = 0$, the $\epsilon = 0.2$ illustrated the convergence result with slower result.

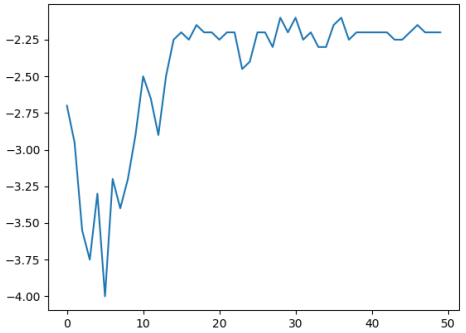

Figure 8: $\epsilon = 0.2$ Asymptote Plot Iterations = 1000

3.3.2 Iterations = 10000

In the figure 9, the result asymptote plot under the situation $\epsilon = 0.2$ is illustrated. Compared to figure 2 with $\epsilon = 0$, the $\epsilon = 0.2$ illustrated the convergence result with slower result.

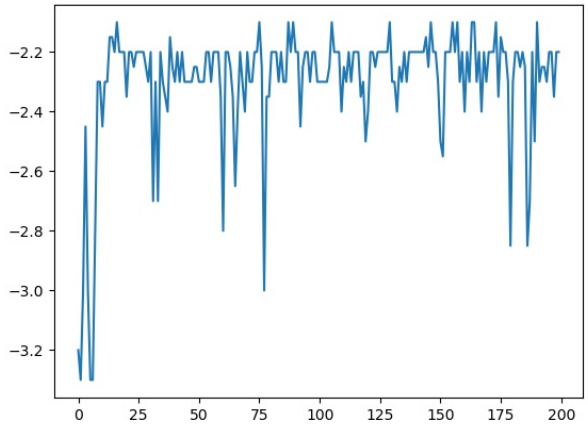

Figure 9: $\epsilon = 0.2$ Asymptote Plot Iterations = 10000

3.3.3 Recommended Actions Map

The score map of the program is shown in figure 10.

```
Please input:sarsa 5 -2 -0.1 -3 10000 0.2
[['>' '>' 'v' 'v' '>' 'v' 'v']
 ['v' '>' '>' '>' '<' 'v' '>']
 ['v' 'v' 'P' 'P' 'v' '<' '<']
 ['>' 'P' 'G' '<' '<' 'P' 'v']
 ['>' '^' 'P' 'P' 'P' '^' '^']
 ['^' '^' '^' '^' '^' '<' '^']]
[[-0.19  0.98  0.4   1.09  0.53  2.64  0.24]
 [-2.46  0.32  0.68  0.48  0.46  0.29  0.48]
 [-2.2  -2.1  -2.    -2.    0.71  1.06 -0.25]
 [-2.1  -2.    5.     4.9   2.94 -2.    -1.4 ]
 [-2.45 -2.1  -2.    -2.    -2.    -2.1  -1.13]
 [-2.47 -2.2  -2.1  -2.1  -2.1  -2.33 -1.61]]
```

Figure 10: $\epsilon = 0.2$ Score Map

3.4 ϵ Exploration

In this part, the result asymptote plots of the $\epsilon = 0.01$ with iterations 1000 and 10000 are given.

3.4.1 Iterations = 1000

In the figure 11, the result asymptote plot under the situation $\epsilon = 0.01$ is illustrated. Compared to figure 2 with $\epsilon = 0$, the $\epsilon = 0.01$ illustrated the convergence result with less iterations.

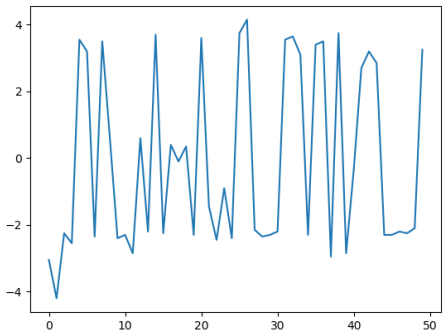

Figure 11: $\epsilon = 0.01$ Asymptote Plot Iterations = 1000

3.4.2 Iterations = 10000

In the figure 12, the result asymptote plot under the situation $\epsilon = 0.01$ is illustrated. Compared to figure 2 with $\epsilon = 0$, the $\epsilon = 0.01$ illustrated the convergence result with less iterations.

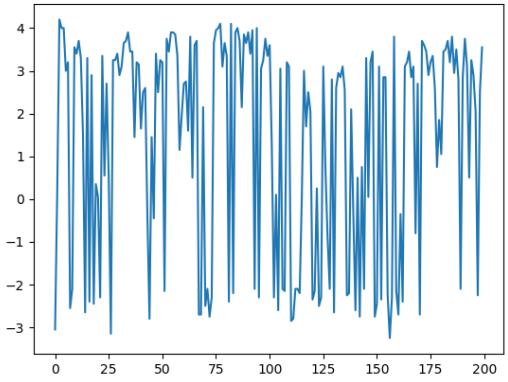

Figure 12: $\epsilon = 0.01$ Asymptote Plot Iterations = 10000

3.4.3 Recommended Actions Map

The recommended action map of the program is shown in figure 13.

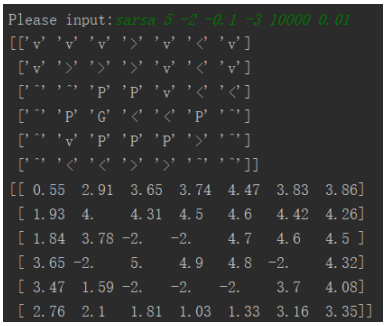

Figure 13: $\epsilon = 0.01$ Recommended Actions Map

3.5 Parameters Optimization

3.5.1 Parameters Not Pointing to Goal

On the following figures, 2 different model parameters are given:

- The result ends in Pit state or give up.
- The result ends in Goal state only.

In the figure 14, the recommended action map which leads to pit or give up only is illustrated.

The parameters are on the following: goal reward: -2, pit reward: -2, move action reward: -1, give up reward : -3, $\epsilon = 0$.

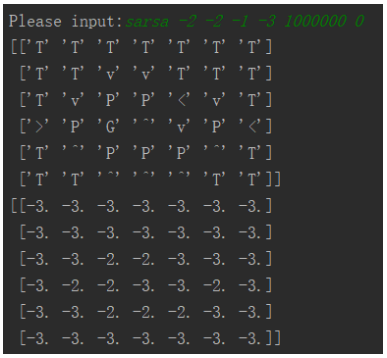

Figure 14: Parameters for Multiple Result

In the figure 15, the recommended action map which leads to goal state only.

The parameters are on the following: goal reward: 5, pit reward: -2, move action reward: -0.01, give up reward : -3, $\epsilon = 0$.

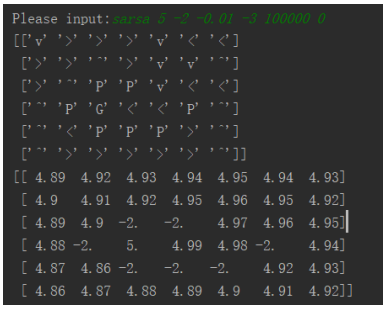

Figure 15: Parameters for Multiple Result

3.5.2 Multiple Result

In the figure 16, the recommendation action map is given and the parameters are: goal reward: 5, pit reward: -2, move action reward: -1, give up reward : -3, $\epsilon = 0$.

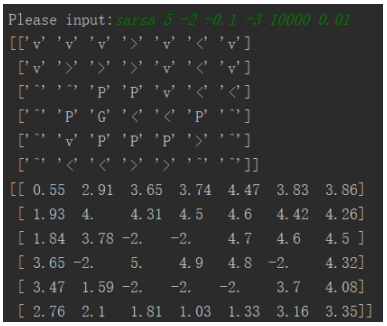

Figure 16: Parameters for Multiple Result

4 Q functions Initialization (Extra Credit)

In the program, the Q functions initialization includes the give up reward.

The regulation for the Q functions initialization, the value for different 5 states should be 0. In the initialization, the program set the give up reward as -3 directly.

The initialization is on the following:

$Q(0,0,0,0,0) \rightarrow Q(0,0,0,0,-3)$