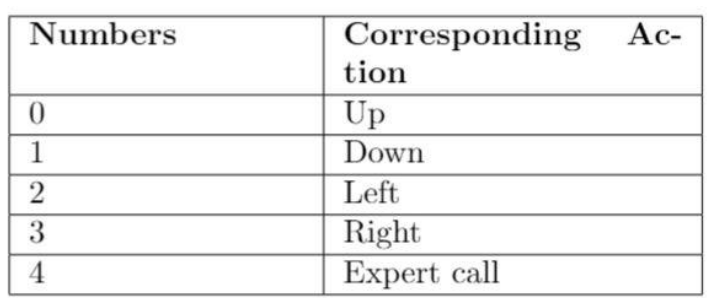
**Human in the loop Reinforcement Learning**

As in the paper, there are 4 implementations here:

1. **Vanilla:** toy grid world problem
2. **ALG1:** where agent has the option to choose expert action at a cost of -5
3. **ALG2:** we are learning the variance of the rewards using Bellman equations

We will follow this legend for all our plots (Expert call is only present in ALG1):



Grid structure: Notice the placement of trap and obstacles

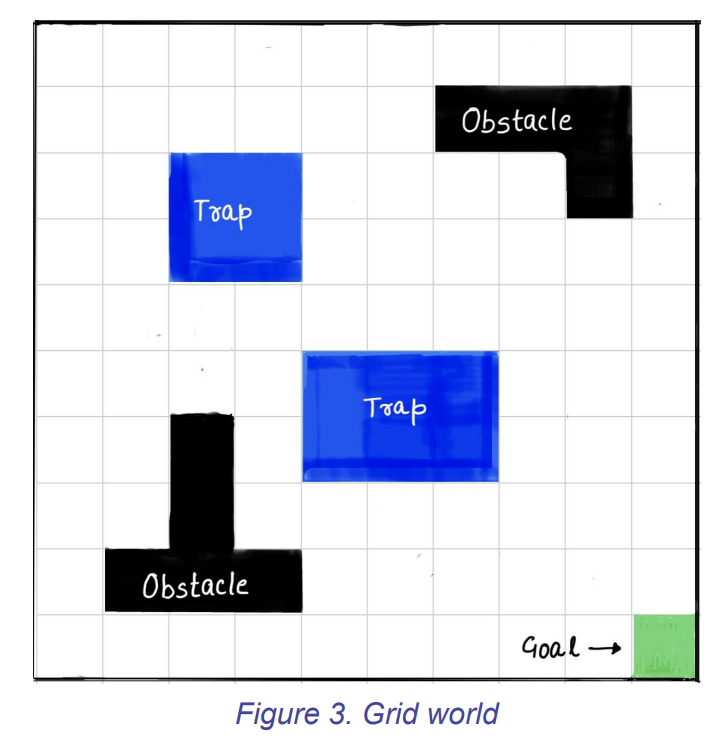


Figure - Grid World

1. **Vanilla:**
   1. **Training – 10 million episodes:**

Training the vanilla agent required a lot of episodes because 30% times the action will be chosen at random. It takes more experience for the agent to learn which would be the best action, especially near the traps. One key criterion while training the agent is where the agent starts when the episode is reset, aka starting\_states of the environment.

To boost the training process, we can set the starting\_states as the cells that are one step away from the traps, this ensures the agent has seen these states multiple times and helps it gain more experience in these crucial states. The agent learns how to escape the traps faster using this method

1. self.start\_states = [[1,2], [1,3], [2,4], [3,4], [4,2], [4,3], [2,1], [3,1], [4,4], [4,5],\
2. [4,5], [4,6], [5,7], [6,7], [6,7], [7,6], [7,5], [7,4], [6,3], [5,3]]



A screenshot of a computer

Description automatically generated with low confidence

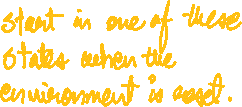


Figure - Starting states

* Training was done for 10 million episodes
* Epsilon decay was used starting from 0.2 which decayed to 0.01 in 5 million episodes

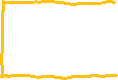
Chart, histogram

Description automatically generated

Figure - Training rewards

A picture containing diagram

Description automatically generated



In the converged action space, almost every action near the trap takes the agent away from the trap



Figure - Best Action

* 1. **Total rewards per episode using Monte Carlo – 100000 episodes:**

Note that in MC sampling the starting states for the agent can either be

1. **“One step away from traps”** which means that even if the best actions are taken every time, due to the stochasticity of the environment, there is roughly a 9-10% chance that the agent will still fall into the trap. This explains the strong peaks in our rewards curve both in the training and during sampling.

Chart, bar chart

Description automatically generated

Figure - MC Sampling rewards (starting state one step away from traps)

1. **“All possible states”** in the grid other than the traps and obstacles. This should result in a rewards curve where the agent does not fall into the trap as much. Even though the actions chosen by this agent are near optimal, we still see spikes in our sampling because of the stochasticity of the environment.

Chart

Description automatically generated

Figure - MC Rewards (starting states are all possible states)

To get a baseline of how well our agent has performed when starting states are all possible states, I used the expert actions provided in Sandip’s memo and ran MC sampling using those actions. I have plotted the returns curve for that below:

Chart

Description automatically generated

Figure - MC Sampling with expert actions (all possible start states)

As you can see our agent performed slightly better than our expert, since it fell in the trap a smaller number of times.

* 1. **Mean of return from each state:**

The following diagrams for the mean and variance of returns are created by keeping the starting states as all the possible states in the environment. Just as a note, the mean of returns, and standard deviation of returns does not change much if the starting states have been changed

Application

Description automatically generated with medium confidence

Figure - Mean of returns

* 1. **Standard deviation of return from each state:**

I chose to plot the standard deviation instead of the variance so that it is more easily understandable, and the values do not go off in the thousands:

A picture containing graphical user interface

Description automatically generated

Figure - Standard Deviation of returns MC sampling

* 1. **State Visitation Count**

Table

Description automatically generated

Figure – State Visitation (Vanilla)

1. **ALG2 –** 
   1. **Trained for 10M episodes:**

Starting states are set to *one step away from the traps* as in Figure 2

Chart, histogram

Description automatically generated

Figure - Rewards per episode for ALG2

A picture containing text, indoor, white

Description automatically generated

Figure - Best action learned by ALG2

All our actions in this grid are optimal and move the agent as far away from the traps as possible

A picture containing graphical user interface

Description automatically generated

Figure - Standard deviation of return calculated from ALG2

These standard deviations have been calculated using Bellman equations. We expect these values to be similar to the values in Figure 9. Although the values aren’t exactly same, the patterns are somewhat similar. One possible difference could be that because we have changed our starting states, ALG2 does not explore a lot of states that are away from our goal. It has more experience in states close to the traps and the states going from the traps towards the goal., however it has less experience in the top left corner of the grid.

1. **ALG1: Trained for 1M episodes**

Initially I took the reward for calling the expert to be -5, but this resulted in a lot of expert calls, so I changed the reward for calling an expert to be -10.

* 1. **Training Rewards per episode:**

Chart, bar chart

Description automatically generated

Figure - Rewards per episode (ALG1)

* 1. **Best Action in each state:**

A picture containing calendar

Description automatically generated

Figure - Best Action (ALG1)

The agent has learned a far better policy than the trained ALG1 in Sandip’s memo (page 8) . One similarity in both our trained action spaces however is that near the goal, the agent chooses to call the expert. This could be because the agent chooses an immediate reward of +100 over a delayed reward with -2 for every step, but with our current reward structure this does not make complete sense yet.

* 1. **MC Sampling: Total rewards per episode**

A picture containing text, antenna

Description automatically generated

Figure - Total Rewards MC Sampling (ALG1)

* 1. **MC Sampling: Mean of returns**

A picture containing graphical user interface

Description automatically generated

Figure - Mean of Returns MC Sampling (ALG1)

* 1. **MC Sampling: Standard Deviation of returns**

A picture containing diagram

Description automatically generated

Figure - Standard Deviation of returns MC Sampling (ALG1)

* 1. **MC Sampling: State visitation counts**

Table

Description automatically generated

Figure - State Visitation Counts